

DURÉE LIBRE

FORMULE INTÉGRALE

8,20 €
PAR MOIS

43 %
de réduction *

Partie réservée au service abonnement Ne rien inscrire

4 / MERCI DE JOINDRE IMPÉRATIVEMENT UN RIB

R

ENDEZ-VOUS

P.80 Logique & calcul
P.86 Art & science
P.88 Idées de physique
P.92 Chroniques de l'évolution
P.96 Science & gastronomie
P.98 À picorer

CHANGER DE NUMÉRATION AVEC LE SYSTÈME « ESPERLUETTE »

Comment convertir facilement l'écriture décimale d'un entier en son écriture dans une autre base de numération? Certaines procédures permettent de le faire... sans avoir à calculer!

L'AUTEUR



JEAN-PAUL DELAHAYE
professeur émérite
à l'université de Lille
et chercheur au
laboratoire Cristal
(Centre de recherche
en informatique, signal
et automatique de Lille)



Jean-Paul Delahaye a notamment publié : **Les Mathématiciens se plient au jeu**, une sélection de ses chroniques parues dans *Pour la Science* (Belin, 2017).

Le système décimal de numération de position est une merveille de simplicité et de puissance qui, pourtant, n'a été disponible dans sa forme actuelle qu'à partir du ^{vi}e siècle en Inde et n'est arrivé en Europe qu'à la fin du ^xe siècle. Il a donné naissance aux systèmes positionnels en base b (b entier positif), dont le système binaire qui est central en informatique.

Nous allons voir comment on peut, sans savoir calculer et en menant uniquement des manipulations locales sur une chaîne de caractères, écrire un nombre quelconque en base 2 ou même en base $b > 2$. Ces méthodes douces de conversion se fondent sur l'existence de systèmes de numération mixtes associant par exemple la notation unaire (ou « notation bâton », chaque unité étant représentée par un bâton) et la notation binaire, ou associant unaire et décimal. Elles conduisent aussi aux systèmes à plusieurs étages tels que le système informatique du « décimal codé binaire ». Nous procéderons progressivement en commençant par le système « esperluette » de règles pour la base 2, dont les clés de fonctionnement conduiront aux systèmes mixtes, puis aux systèmes à plusieurs étages.

De nombreuses informations présentées ici m'ont été communiquées par Roland Yéléhada, un lecteur de la rubrique qui m'a fait parvenir une longue lettre datée du 12 septembre 2020. À ma connaissance, les systèmes décrits ici sont originaux.

Les systèmes de règles de type esperluette ne nécessitent aucune compréhension de ce

qu'on fait. Ils sont plus intéressants pour concevoir des programmes informatiques que pour opérer à la main, car ils manipulent de longues séquences de caractères. Il est amusant de découvrir que, même en 2021, on peut encore progresser dans la mise au point de méthodes destinées à représenter et manipuler les nombres entiers.

Rappelons que le nombre qui s'écrit $a_n a_{n-1} \dots a_2 a_1 a_0$ en base positionnelle b représente l'entier : $a_n b^n + a_{n-1} b^{n-1} + \dots + a_2 b^2 + a_1 b^1 + a_0 b^0$. En base 2, la suite 1101 représente donc l'entier :

$$1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 8 + 4 + 0 + 1 = 13.$$

Le plus souvent, on impose aux entiers a_i d'être compris entre 0 et $b-1$, mais il est parfois utile de s'affranchir de cette limitation, et c'est d'ailleurs une technique utilisée en informatique pour manipuler de grands nombres.

Lorsque nous manipulerons plusieurs bases de numération en même temps, nous indiquerons en indice la base utilisée; par exemple, $13_{10} = 1101_2$. Venons-en à notre premier système de règles.

LE SYSTÈME ESPERLUETTE POUR LA BASE 2

Le symbole typographique & porte le nom d'« esperluette » ou « et commercial » et aurait été introduit par Tiron, le secrétaire de Cicéron. Ce sera le nom général des systèmes de règles que nous allons décrire. Le premier et le plus simple concerne la base 2. Il permet la conversion des entiers en binaire sans même avoir à connaître la notion de puissance. Il fonctionne avec trois ➤

LE SYSTÈME DÉCIMAL DE POSITION

1

Le système de numération décimal positionnel a été conçu progressivement en Inde. Passant par les pays arabes grâce au mathématicien Al-Khwarizmi (vers 780-850) dont le nom nous a donné le mot « algorithme », il n'est arrivé en Europe qu'à la fin du x^e siècle. C'est à Gerbert d'Aurillac, devenu le pape Sylvestre II en 999, qu'on doit cette introduction. Il écrivit des traités sur les opérations arithmétiques et il inventa un abaque fondé sur le système décimal positionnel, c'est-à-dire un système de calcul avec des jetons où, selon leur position, ils représentent plusieurs unités, dizaines, centaines, etc. Grâce à son ouvrage *Liber abaci* de 1202, Léonard de Pise, alias Fibonacci, contribua cette fois d'une façon définitive à la diffusion du système, qui fut progressivement adopté partout en Europe.

Gerbert d'Aurillac, le pape de l'an mil (ici à la droite de l'empereur Otton III), était astronome et mathématicien.



2

L'ORIGINE DU SYSTÈME BINAIRE

On a trouvé en Chine, dès le ix^e siècle avant notre ère, des versions partielles du système binaire pas toujours destinées à compter, par exemple les 64 hexagrammes (I-Ching) qui sont la liste complète des figures composées de six traits horizontaux continus ou discontinus empilés. Un traité de Leibniz écrit en français en 1703 (*Explication de l'arithmétique binaire qui se sert des seuls caractères 0 et 1 avec des remarques sur son utilité et sur ce qu'elle donne le sens des anciennes figures chinoises du Fuxi*, <https://bit.ly/3kwQOjQ>) fournit la première explication claire du système binaire qu'il note comme nous avec des 0 et des 1. Le système binaire est essentiel en informatique, où il permet de représenter des nombres avec des mémoires à deux états (cartes perforées, positions d'un curseur, états électriques, etc.).

TABLE 86 MEMOIRES DE L'ACADEMIE ROYALE

DES	bre	entiers au-dessous du double du	1000	10000	100000
NOMBRES,	plus haut degré.	Car ici, c'est comme si on disoit, par exemple, que 111 ou 7 est la somme de quatre, de deux	111	1111	11111
0	0	1	1	1	1
1	1	2	2	2	2
2	2	3	3	3	3
3	3	4	4	4	4
4	4	5	5	5	5
5	5	6	6	6	6
6	6	7	7	7	7
7	7	8	8	8	8
8	8	9	9	9	9
9	9	10	10	10	10
10	10	11	11	11	11
11	11	12	12	12	12
12	12	13	13	13	13
13	13	14	14	14	14
14	14	15	15	15	15
15	15	16	16	16	16
16	16	17	17	17	17
17	17	18	18	18	18
18	18	19	19	19	19
19	19	20	20	20	20
20	20	21	21	21	21
21	21	22	22	22	22
22	22	23	23	23	23
23	23	24	24	24	24
24	24	25	25	25	25
25	25	26	26	26	26
26	26	27	27	27	27
27	27	28	28	28	28
28	28	29	29	29	29
29	29	30	30	30	30
30	30	31	31	31	31
31	31	32	32	32	32
32	32	33	33	33	33
33	33	34	34	34	34
34	34	35	35	35	35
35	35	36	36	36	36
36	36	37	37	37	37
37	37	38	38	38	38
38	38	39	39	39	39
39	39	40	40	40	40
40	40	41	41	41	41
41	41	42	42	42	42
42	42	43	43	43	43
43	43	44	44	44	44
44	44	45	45	45	45
45	45	46	46	46	46
46	46	47	47	47	47
47	47	48	48	48	48
48	48	49	49	49	49
49	49	50	50	50	50
50	50	51	51	51	51
51	51	52	52	52	52
52	52	53	53	53	53
53	53	54	54	54	54
54	54	55	55	55	55
55	55	56	56	56	56
56	56	57	57	57	57
57	57	58	58	58	58
58	58	59	59	59	59
59	59	60	60	60	60
60	60	61	61	61	61
61	61	62	62	62	62
62	62	63	63	63	63
63	63	64	64	64	64

Une page du mémoire de Leibniz de 1703.

- > règles élémentaires, qui s'écrivent avec 24 symboles ou caractères en tout.

Dans la présentation des règles, x et y désignent des suites (éventuellement vides) quelconques de symboles pris parmi 0, 1 et &.

TROIS RÈGLES POUR PASSER AU BINAIRE

Règle $r1: \&x \rightarrow 0\&x$

Règle $r_2: x0 \& \& \gamma \rightarrow x \& 0\gamma$

Règle $r3: x0 \& y \rightarrow x1y$

La règle r_1 signifie que si une suite de symboles pris parmi 0, 1 et & commence par &, on peut introduire un 0 devant la suite. La règle r_2 signifie que si, dans une suite de symboles pris parmi 0, 1 et &, on trouve quelque part la séquence 0&&, alors on peut remplacer celle-ci par &0. De même, la règle r_3 permet de remplacer 0& par 1.

Opérons, grâce à ces trois règles, la conversion de l'entier 5 écrit en unaire, &&&&&, pour obtenir son expression binaire, à savoir 101 :

$$\begin{array}{ccccccc} & \underline{\&\&\&\&} & \xrightarrow{r_1} & \underline{0\&\&\&\&} & \xrightarrow{r_2} & \&0\&\&\& \\ \xrightarrow{r_2} & \underline{\&\&0\&} & \xrightarrow{r_1} & \underline{0\&\&0\&} & \xrightarrow{r_2} & \underline{\&00\&} & \xrightarrow{r_1} & \underline{0\&00\&} \\ \xrightarrow{r_3} & 100\& & \xrightarrow{r_3} & 101. \end{array}$$

On a indiqué à chaque fois la règle utilisée et on a souligné l'endroit de la suite de symboles sur lequel la règle agit. Autre conversion, cette fois de «onze fois &» vers 1011 :

[illegible]
$$\begin{aligned} &\rightarrow_{r_2} 888808 \rightarrow_{r_1} 0888808 \rightarrow_{r_2} 8088808 \\ &\rightarrow_{r_2} 880808 \rightarrow_{r_1} 0880808 \rightarrow_{r_1} 800808 \\ &\rightarrow_{r_1} 0800808 \rightarrow_{r_3} 100808 \rightarrow_{r_3} 10108 \rightarrow_{r_3} 1011. \end{aligned}$$

La conversion se fait de la notation unaire de n , c'est-à-dire n fois le symbole $\&$, vers la notation binaire de n , et cela sans manipuler de puissances de 2 et sans rien avoir à connaître d'autre que les trois règles.

Si l'on utilise *r3* trop tôt, on pourrait se retrouver bloqué, comme dans cet exemple: $\&\&\&\&\rightarrow_{r1}0\&\&\&\&\rightarrow_{r3}1\&\&\&$. Pour que la conversion fonctionne, il faut donc s'imposer de n'utiliser *r3* que lorsque les règles *r1* et *r2* ne sont plus utilisables. En revanche, *r1* et *r2* sont utilisables librement, dans l'ordre que l'on veut.

Les manipulations pour convertir un nombre assez petit se font à la main sur une feuille de papier. Mais bien évidemment, pour les grands nombres, le procédé ne peut concerner qu'une machine programmée pour appliquer les règles. Notons qu'une machine sans processeur arithmétique conviendrait: elle doit seulement savoir reconnaître certaines séquences et opérer des ajouts ou des substitutions. Avec les règles r_1 , r_2 et r_3 , aucune compétence en calcul n'est requise pour convertir un nombre en base 2!

D'où vient le miracle? Bien sûr, on peut tester le système et découvrir qu'il fonctionne toujours bien. Mais soyons sérieux et prouvons que la méthode est infaillible.

L'idée de la démonstration est de donner un sens numérique à toute suite finie de symboles

3

LA NUMÉRATION MAYA

La numération maya a été utilisée entre le III^e et le XIII^e siècle avec plusieurs variantes. Nous en présentons une version simplifiée pour illustrer l'idée des systèmes de numération à deux étages.

Il s'agit en première analyse d'une numération de position en base 20, numération dite « vicésimale » ou « vigésimale ». Nous utilisons 10 chiffres pour notre numération de position décimale, il en aurait donc fallu 20 pour le système maya. Les Mayas se sont contentés de trois symboles : un pour le zéro (une sorte de coquille), un autre désignant l'unité (un rond noir), et un troisième (une barre horizontale) qui représente 5 ronds noirs.

Cette notation des chiffres peut être vue comme une notation unaire, légèrement perfectionnée en regroupant les paquets

de 5 ronds noirs. Sans ces regroupements, le système maya serait un pur système à deux étages en base 20 et unaire, comme le texte de l'article en définit avec des règles de manipulation ne demandant aucun calcul.

L'utilisation de la base 20 est naturelle, car c'est le nombre de doigts et d'orteils. Les systèmes en base 20 n'ont pas été utilisés seulement par les Mayas et, d'ailleurs, avec notre « quatre-vingts », nous en gardons encore une trace aujourd'hui en français (voir : https://fr.wikipedia.org/wiki/Système_vicésimal).

0	1	2	3	4	
●	●●	●●●	●●●●	●●●●●	_____
1	2	3	4	5	
●	●●	●●●	●●●●	●●●●●	_____
6	7	8	9	10	
●	●●	●●●	●●●●	●●●●●	_____
11	12	13	14	15	
●	●●	●●●	●●●●	●●●●●	_____
16	17	18	19	20	● 
16	17	18	19	20	

Exemple :

●  ●●● ●●● _____ représente :

$20 \times 20^3 + 3 \times 20^2 + 4 \times 20^1 + 5 \times 20^0 = 161285$

pris parmi 0, 1 et &, ce qui d'ailleurs définit un système de numération mixte de représentation des entiers, combinant unaire et binaire.

On attribue une valeur numérique à toute suite de 0, 1 et &. Par exemple, pour &&&11&0&110, on considère que la suite représente un entier qui est la somme de deux composantes.

- Composante A: on ne retient que les 0 et les 1, dans l'ordre, soit 110110, qu'on lit comme une écriture en base 2. Ici, $32 + 16 + 4 + 2 = 54$.

- Composante B: on considère que chaque & représente 2^i , où i est le nombre de symboles 0 ou 1 placés à sa droite. Le dernier & de notre exemple représente donc 8, celui un peu plus à gauche, 16, et chacun des trois & du début, 64. Au total, les & représentent donc $8 + 16 + 64 + 64 + 64 = 216$.

Finalement, le nombre représenté par &&&11&0&110 est $54 + 216 = 270$.

Le secret du système esperluette est alors facile à percer:

- Quand on utilise les règles r_1 , r_2 et r_3 , on ne change pas le nombre représenté par la somme des composantes A et B de la séquence que l'on transforme.

- Au départ, quand il n'y a que des &, le nombre représenté est le nombre de fois que & est répété, car il n'y a que la composante B, où chaque symbole compte pour $2^0 = 1$. À l'arrivée, quand on s'est débarrassé de tous les &, le nombre représenté est celui qu'on obtient en lisant les 0 et 1 comme un nombre en écriture binaire, car il n'y a plus que la composante A.

- Ainsi, quand on réussit à passer d'une écriture avec n fois & à une écriture avec uniquement des 0 et des 1, l'entier représenté par les composantes A et B est resté le même à chaque étape de transformation, et donc la transformation opère la conversion de n en écriture binaire.

Reste à contrôler ce qui, pour l'instant, a été affirmé sans preuve: appliquer une règle r_1 , r_2 ou r_3 ne change pas le nombre représenté en additionnant les composantes A et B.

C'est bien clair pour r_1 , car les zéros en tête d'une écriture binaire ne comptent pas. La règle r_2 remplace 0&& par &0. Elle remplace donc deux & qui valaient chacun 2^i , où i est le nombre des 0 et 1 à droite de ces deux &, par un & qui vaut 2^{i+1} puisqu'il y a un chiffre 0 en plus à sa droite. Dans le décompte de la composante B, on remplace donc $2^i + 2^i$ par 2^{i+1} , ce qui ne le change pas. On ne change pas la composante A en appliquant r_2 . Le total ne change donc pas avec r_2 . Un raisonnement du même type montre que r_3 ne change pas l'entier représenté.

Pour être certain que la méthode consistant à appliquer toujours r_1 et r_2 en priorité fonctionne, il faut s'assurer qu'on ne reste pas bloqué. Cela résulte de ce que les 0 introduits par r_1 se déplacent uniquement vers la droite

4

LE DÉCIMAL CODÉ BINAIRE

Le système de numération vigésimale maya (voir l'encadré 3) peut être vu, en le simplifiant, comme un système de numération à deux étages. Il existe cependant deux domaines où l'utilisation d'un système à deux étages fut indubitable, ce sont l'informatique et l'électronique.

Dans ces domaines, pour être proche des humains qui ne maîtrisent bien que le décimal et pour être proche en même temps des machines qui préfèrent le binaire, on utilise assez fréquemment le système « décimal codé binaire » (DCB) qui est un système à deux étages : décimal et binaire.

Les nombres sont codés par des suites de chiffres décimaux qui, eux, sont écrits en binaire, en général sur quatre bits. Le nombre 835 sera donc écrit : 1000-0011-0101. Les codes des chiffres décimaux sont :

0 : 0000	1 : 0001	2 : 0010
3 : 0011	4 : 0100	5 : 0101
6 : 0110	7 : 0111	8 : 1000
9 : 1001		

La notation DCB n'est pas très économique en espace, car,



par exemple, 16 qui s'écrit 10000 en binaire doit s'écrire 00010110 en DCB. Cependant, le DCB facilite l'affichage des résultats que les humains exigent en décimal, car la conversion du DCB vers le décimal est très simple. Les calculatrices électroniques (calculatrices), avant de devenir de véritables petits ordinateurs, utilisaient en général le DCB, de même que de nombreux instruments de mesure (voir d'autres informations sur le DCB sur https://en.wikipedia.org/wiki/Binary-coded_decimal).

Les systèmes de numération présentés dans le texte principal de la rubrique généralisent l'idée d'un système à deux étages et donnent les règles permettant la conversion facile d'une numération à une autre.

et ne sont arrêtés que lorsque la suite de symboles ne comporte plus deux & consécutifs, situation où l'utilisation de r_3 ne peut pas faire réapparaître deux & consécutifs, et où, appliquée de manière répétée, elle fait disparaître tous les &.

AJOUTER DE NOUVELLES RÈGLES

Le système de trois règles est celui nécessaire pour la conversion d'un nombre de la notation unaire à la notation binaire. Pour avoir un système permettant la conversion inverse du binaire vers le unaire, il suffit d'ajouter les règles inversées.

Règle r_1' : $0&x \rightarrow &x$

Règle r_2' : $x&0y \rightarrow x0&&y$

Règle r_3' : $x1y \rightarrow x0&y$

>

- On pourrait compléter le système de règles pour que la conversion soit plus rapide et par exemple ajouter la règle $x0\&\&\&y \rightarrow x\&\&0y$, qui correspond à deux applications de r_2 .

D'autres règles sont à imaginer, mais pour être certain que le système ne produit pas d'absurdité (par exemple une conversion erronée), il faut s'assurer, pour chaque règle ajoutée, que son application laisse stable l'entier représenté par la somme des composantes A et B.

Toute suite finie composée des trois symboles 0, 1 et & représente ainsi un nombre unique et les règles indiquent comment manipuler de telles séquences sans changer le nombre représenté. On a donc un système de numération mixte associant unaire et binaire.

ESPERLUETTE POUR LA BASE 3

Le système esperluette s'adapte à la base 3, avec les règles suivantes.

Règle s1: $\&x \rightarrow 0\&x$

Règle s2: $x0\&\&\&y \rightarrow x\&0y$

Règle s3: $x0\&y \rightarrow x2y$

Règle s4: $x0xy \rightarrow x1y$

On utilise la couleur rouge pour les chiffres car ils désignent des chiffres de la base 3 qu'on

utilisera plus loin en même temps que des chiffres de la base 2.

L'entier 11 s'écrit $9+2 = 1 \times 3^2 + 0 \times 3^1 + 2 \times 3^0 = 102_3$. Voyons comment s'opère la conversion avec le système esperluette de règles pour la base 3:

$$\&\&\&\&\&\&\&\&\&\rightarrow_{s_1} \textcolor{red}{0}\&\&\&\&\&\&\&\&\&$$
$$\rightarrow_{s_2} \&0\&\&\&\&\&\&\&\rightarrow_{s_2} \&\&0\&\&\&\&\&$$
$$\rightarrow_{s_2} \&\&\&0\&\& \rightarrow_{s_1} 0\&\&\&0\&\& \rightarrow_{s_2} \&00\&\&$$
$$\rightarrow_{s1} 0\&00\&\& \rightarrow_{s3} 0\&02 \rightarrow_{s4} 102$$

Pour que la conversion fonctionne, il faut donner priorité aux règles s1 et s2 sur les autres, et donner priorité à la règle s3 sur la règle s4.

Le même type de raisonnement que pour la base 2 démontre que les règles s_1 , s_2 , s_3 et s_4 fonctionnent toujours de façon infaillible pour convertir de la notation unaire à la notation ternaire.

UN PREMIER SYSTÈME À DEUX ÉTAGES

L'utilisation des règles s1 et s2 est intéressante et va nous donner un premier exemple de système de numération à deux étages. Dans une première phase des transformations utilisant les règles esperluette pour la base 3, les & sont poussés le plus à droite possible. Pour notre exemple, &&&&&&&&&&&, on aboutit à 0&&00&&. La séquence obtenue quand s2 n'est plus utilisable se lit de la façon suivante :

- les 0 délimitent des boîtes dans lesquelles il y a plus ou moins de &, mais toujours moins de trois, car sinon on pourrait utiliser la règle s2. En délimitant les boîtes plus clairement dans notre exemple, cela donne :

0 [&] 0 [] 0 [&&], c'est-à-dire une boîte avec un &, une boîte avec zéro & et une boîte avec deux &.

Dans chacune des boîtes, il y a, en écriture unaire, le chiffre qui correspond à l'écriture en base 3 du nombre décimal 11 : 1, 0, 2.

Ce qu'on obtient est donc une écriture de 11 dans un système à deux étages: l'un utilisant l'unaire, l'autre utilisant le ternaire. Plus loin, nous présenterons des systèmes de type esperluette aboutissant à un système de numération à deux étages, décimal et binaire.

CONVERSION DE LA BASE 2 À LA BASE 3

Les deux systèmes présentés peuvent se regrouper en un seul qui permet de convertir de la base 2 à la base 3 et inversement. Pour ce faire, on prend les règles de la base 2 et leurs inverses ($r_1, r_2, r_3, r_1', r_2', r_3'$), les règles de la base 3 et leurs inverses ($s_1, s_2, s_3, s_4, s_1', s_2', s_3', s_4'$). Cela fait 14 règles. On passera de la base 2 à la base 3 en utilisant les règles de la base 2 à l'envers pour obtenir le nombre en unaire, puis les règles de la base 3 pour passer du unaire au ternaire. Bien sûr, on pourra aussi passer du ternaire au binaire avec les 14 règles.

ADDITION EN BASE MIXTE BINAIRE-UNAIRE

Voyons comment on peut, sans calcul compliqué et en restant toujours avec des nombres en notation mixte binaire-unaire, réaliser une addition. Prenons deux nombres écrits en base mixte binaire-unaire, 108881108880 et 10881110 .

On peut savoir qu'ils valent 98 et 78 en utilisant la méthode des composantes A et B (voir le texte principal). Nous effectuons cette conversion juste pour contrôler, ces valeurs ne seront pas utilisées :

$$2 + 2 + 2 + 4 + 8 + 16 + 16 + 16 + 32 = 98$$

$$\text{et } 2 + 4 + 8 + 16 + 16 + 16 + 32 = 78 ;$$

$$98 + 78 = 176.$$

Pour faire l'addition tout en restant codé dans le système mixte binaire-unaire, on se débarrasse d'abord de tous les 1 en utilisant la règle r3' qui permet de remplacer 1 par 06. Cela conduit à 060666060606660 et 0606060606060.

Cette écriture est facile à lire : les 0 délimitent des boîtes dans lesquelles les nombres sont écrits en

unaire avec des 8. On aligne les zéros en commençant par ceux situés le plus à droite.

Cela aligne aussi les boîtes :

```
0-0-0-000-0-0-0-0-000-0;
0-0-0- 00 - 0-0-0-0- 0 -0.
```

Il suffit alors d'additionner

le contenu des boîtes, c'est-à-dire de concaténer les contenus des boîtes d'une même colonne. Pour une addition de k nombres, la méthode serait la même.

Le résultat est ici :

0-66-0-66666-0-66-0-66-
0-6666-0.

Avec la méthode des composantes, on contrôle qu'il s'agit bien de 176 (lecture des 8 de droite à gauche) :

$$\begin{array}{r} 2+2+2+2 \\ +4+4 \end{array}$$
$$+ 8 + 8$$
$$+ 16 + 16 + 16 + 16 + 16$$
$$+ 32 + 32$$
$$= 176.$$

Si on le souhaite, le

résultat est alors rapidement simplifié (par utilisation

de r_1 , r_2 et r_3) et par exemple réécrit seulement avec

des 0 et des 1, c'est-à-dire en binaire :

$$10110000 = 16 + 32 + 128 = 176.$$

Le système de numération associé à cette écriture utilisant les six symboles &, 0, 1, 0, 1, 2 est un système de numération triple. Il est certes un peu compliqué, mais il est remarquable que l'on puisse circuler d'une notation à une autre assez simplement avec quelques règles élémentaires. Si on le souhaitait, en introduisant d'autres couleurs et les règles nécessaires, on aurait par exemple un système de numération quadruple décimal-ternaire-binaire-unaire, etc.

POUR TOUTES LES BASES!

La généralisation du système esperluette pour la base 5 est facile à imaginer. Elle est donnée ci-dessous en utilisant la couleur verte pour les chiffres. La généralisation à une base quelconque, b , devient alors évidente.

Règle $t1$: $\&x \rightarrow 0\&x$

Règle $t2$: $x0\&\&\&\&y \rightarrow x\&0y$

Règle $t3$: $x0\&\&\&y \rightarrow x4y$

Règle $t4$: $x0\&\&y \rightarrow x3y$

Règle $t5$: $x0\&y \rightarrow x2y$

Règle $t6$: $x0y \rightarrow x1y$

Comme précédemment, si l'on utilise le plus possible les règles $t1$ et $t2$, on arrive à une représentation du nombre étudié dans un système à deux étages: unaire et quinaire.

Par exemple, l'entier $39_{10} = 25_{10} + 10_{10} + 4_{10} = 1 \times 5^2 + 2 \times 5^1 + 4 \times 5^0 = 124_5$ s'écrira $0\&0\&0\&0\&\&$, ce qui, en faisant apparaître les boîtes, donne:

0 [&] 0 [&&] 0 [&&&].

De tels systèmes, ou du moins des systèmes assez proches, ont déjà été utilisés. Le système des Mayas, à peu de chose près (en remplaçant le symbole qui représente le 5 par cinq points ronds), est un système à un étage en base 20 et un étage unaire (voir l'encadré 3).

En informatique et en électronique, on utilise un système de ce type, le «décimal codé binaire», ou DCB, qui est un système à deux étages: un étage décimal et un étage binaire.

Pour représenter 237_{10} en DCB, on écrit 10 11 111, c'est-à-dire les trois chiffres 2, 3 et 7 successivement écrits en binaire. Dans la notation DCB, pour éviter toute confusion, il faut soit avoir réservé des espaces mémoire distincts pour chacune des écritures 10, 11 et 111, soit utiliser un séparateur ($10 \bullet 11 \bullet 111$).

On a vu que les systèmes esperluette peuvent convertir en notation à deux étages ternaire-unaire ou quinaire-unaire, mais l'étage le plus bas est unaire, ce qui est un peu encombrant. Existe-t-il un système de règles de type esperluette qui permettraient de convertir un nombre n donné en unaire en notation à deux étages décimal-binaire (comme le décimal codé binaire)?

La réponse est oui. C'est possible en introduisant des règles qui vont convertir le contenu des boîtes en unaire évoquées plus haut pour les écrire en binaire. Attention, ce système n'est pas exactement le même que celui qu'on

utiliserait pour créer un système mixte décimal-binaire et permettre les conversions entre le décimal et le binaire.

ESPERLUETTE POUR LE DÉCIMAL CODÉ BINAIRE

Voici les règles du système esperluette pour le DCB.

Règle $u1$: $\&x \rightarrow 0\&x$

Règle $u2$: $x0\&\&\&\&\&\&y \rightarrow x\&0y$

Règle $u3$: $x0\&y \rightarrow x00\&y$

Règle $u4$: $x0\&\&y \rightarrow x\&0y$

Règle $u5$: $x0\&y \rightarrow x1y$

Règle $u6$: $x00y \rightarrow x000y$

L'utilisation des six règles se fera en donnant priorité aux règles $u1$ et $u2$ qui conduisent à l'écriture en décimal-unaire du nombre considéré. Ensuite, on utilisera les règles $u3$ et $u4$ qui mènent à une écriture à trois étages, décimal-binaire-unaire. Enfin, les règles $u5$ et $u6$ réduisent la séquence pour l'écrire en décimal-binaire.

Voici un exemple pour comprendre la série des transformations. Partons d'une séquence de 2307 fois le &. Après avoir appliqué autant que possible les règles $u1$ et $u2$, on arrive à $0\&\&0\&\&00\&\&\&\&\&$, ce qui est une représentation de 2307 en notation à deux étages décimal-unaire avec le 0 comme séparateur. L'application des règles $u3$ et $u4$ produit $00\&000\&0\&000\&0\&0\&$, ce qui, en délimitant les boîtes, s'écrit:

0 [0&0] 0 [0&0&] 0 [] 0 [0&0&0&].

On voit, séparées par des 0 (noirs), une boîte 0&0, puis une autre 0&0&, puis une boîte vide et enfin une quatrième boîte 0&0&0&. Chacune de ces boîtes est elle-même une suite de boîtes contenant plus ou moins de & séparés par des 0 bleus, et on reconnaît dans la première boîte le nombre 2 en binaire-unaire. Le nombre 3 en binaire-unaire est dans la seconde. Le 0 correspond à la troisième boîte, et le nombre 7 en binaire-unaire à la dernière.

La règle $u5$ transforme alors les boîtes du second étage en binaire, ce qui aboutit à 01001100111. En introduisant des blancs et en délimitant les boîtes:

0 [10] 0 [11] 0 [] 0 [111], où l'on reconnaît déjà 2307.

La règle $u6$ sert à placer un 0 bleu dans les boîtes délimitées par des 0 noirs ne contenant rien. Le résultat obtenu, 010011000111, est maintenant facile à lire si l'on comprend bien que les 0 noirs sont des séparateurs, et que les chiffres en bleu représentent les entiers 2, 3, 0 et 7 écrits en binaire:

0 10 0 11 0 0 0 111.

Le système des règles a donc converti le nombre 2307 du unaire au décimal-binaire. Vous imaginez sans mal que l'on conçoit aussi des systèmes de numération à autant d'étages que l'on veut, par exemple décimal-quaternaire-ternaire-binaire. De quoi s'amuser encore un moment! ■

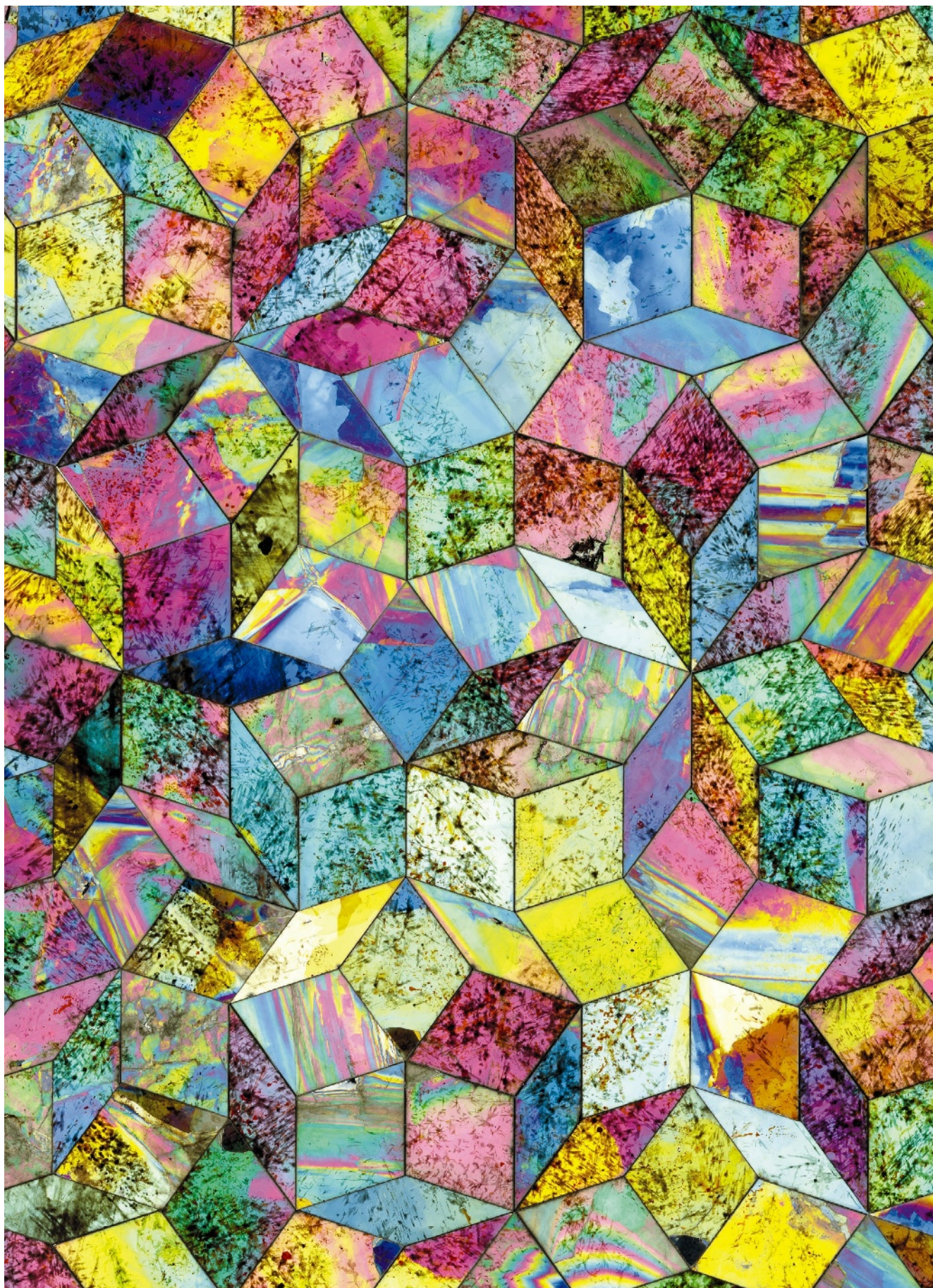
BIBLIOGRAPHIE

Wikipédia, **Système de numération indo-arabe**, consulté en septembre 2020.

Wikipédia, **Esperluette**, septembre 2020.

Wikipédia, **Binary-coded decimal**, septembre 2020.

Wikipédia, **Numération maya**, septembre 2020.



MicaPenrose, de Léa Barbazanges.