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 $r1$, $r2$ et $r3$, 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 $r1$, $r2$ ou $r3$ ne change pas le nombre représenté en additionnant les composantes A et B.

C'est bien clair pour $r1$, car les zéros en tête d'une écriture binaire ne comptent pas. La règle $r2$ 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 $r2$. Le total ne change donc pas avec $r2$. Un raisonnement du même type montre que $r3$ ne change pas l'entier représenté.

Pour être certain que la méthode consistant à appliquer toujours $r1$ et $r2$ en priorité fonctionne, il faut s'assurer qu'on ne reste pas bloqué. Cela résulte de ce que les 0 introduits par $r1$ se déplacent uniquement vers la droite et ne sont arrêtés que lorsque la suite de symboles ne comporte plus deux & consécutifs, situation où l'utilisation de $r3$ 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 $r1'$: $0\&x \rightarrow \&x$
Règle $r2'$: $x\&0y \rightarrow x0\&\&y$
Règle $r3'$: $x1y \rightarrow x0\&y$ >

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 (calculettes), 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.

> 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 $r2$.

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 $s4$: $x0\&y \rightarrow x1y$

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 $9+2 = 1\times3^2+0\times3^1+2\times3^0 = 102_3$. Voyons comment s'opère la conversion avec le système esperluette de règles pour la base 3 :

$\&\&\&\&\&\&\&\&\&\&\& \rightarrow_{s1} 0\&\&\&\&\&\&\&\&\&\&\& \rightarrow_{s2} \&0\&\&\&\&\&\&\&\&\& \rightarrow_{s2} \&\&0\&\&\&\&\& \rightarrow_{s2} \&\&\&0\&\& \rightarrow_{s1} 0\&\&\&0\&\& \rightarrow_{s2} \&00\&\& \rightarrow_{s1} 0\&00\&\& \rightarrow_{s3} 0\&02 \rightarrow_{s4} 102$

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 $s1$, $s2$, $s3$ et $s4$ 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 ($r1$, $r2$, $r3$, $r1'$, $r2'$, $r3'$), les règles de la base 3 et leurs inverses ($s1$, $s2$, $s3$, $s4$, $s1'$, $s2'$, $s3'$, $s4'$). 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.

5 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, 10&&&110&&&0 et 10&&1110.

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
et 2 + 4 + 8 + 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 0&. Cela conduit à 0&0&&&0&0&0&&&0 et 0&0&&0&0&0&0.

Cette écriture est facile à lire : les 0 délimitent des boîtes dans lesquelles les nombres sont écrits en unaire avec des &. On aligne les zéros en commençant par ceux situés le plus à droite. Cela aligne aussi les boîtes : 0-&-0-&&&-0-&-0-&-0-&&&-0 ; 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-&&-0-&&&&&-0-&&-0-&&-0-&&&&-0.
Avec la méthode des composantes, on contrôle qu'il s'agit bien de 176 (lecture des & de droite à gauche) :
2 + 2 + 2 + 2 + 4 + 4 + 8 + 8 + 16 + 16 + 16 + 16 + 32 + 32 = 176.

Si on le souhaite, le résultat est alors rapidement simplifié (par utilisation de $r1$, $r2$ et $r3$) et par exemple réécrit seulement avec des 0 et des 1, c'est-à-dire en binaire : 10110000 = 16 + 32 + 128 = 176.