

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' : $x1y \rightarrow x0&y$

>