Présentation

Au commencement étaient le 0 et le 1, puis nous créâmes les nombres, les textes, les images et les sons.

Dans ce chapitre, nous voyons comment les nombres sont représentés dans les ordinateurs avec des 0 et des 1. Nous introduisons la notion de base, en partant de la notation décimale que nous utilisons ordinairement pour écrire les nombres entiers. Nous passons par la base cinq puis décrivons la base deux, aussi appelée représentation binaire. Nous généralisons ensuite aux nombres relatifs en utilisant la notation en complément à deux, puis aux nombres à virgule, représentés par leur signe, leur mantisse et leur exposant.

Le livre de l'addition et de la soustraction d'après le calcul indien de Muhammad al-Khwarizmi (783 ? -850 ?), qui présente la numération décimale à position et des algorithmes permettant d'effectuer les opérations sur les nombres exprimés dans ce système, a été le vecteur de la diffusion de ce système de numération dans le bassin méditerranéen. Le mot algorithme est dérivé du nom d'al-Khwarizmi et le mot algèbre (al- jabr) du titre d'un autre de ses livres.

Présentation

Vus de l'extérieur, les ordinateurs et les programmes que nous utilisons tous les jours permettent de mémoriser, de transmettre et de transformer des nombres, des textes, des images, des sons, etc.

Pourtant, quand on les observe à une plus petite échelle, ces ordinateurs ne manipulent que des objets beaucoup plus simples : **des 0 et des 1**.

Mémoriser, transmettre et transformer des nombres, des textes, des images ou des sons demande donc d'abord de les représenter comme des suites de 0 et de 1.

La mémoire des ordinateurs est constituée d'une multitude de petits circuits électroniques qui, chacun, ne peuvent être que dans deux états.

Présentation

Comme il fallait donner un nom à ces états, on a décidé de les appeler 0 et 1, mais on aurait pu tout aussi bien les appeler A et B, froid et chaud ou faux et vrai.

Une telle valeur, 0 ou 1, s'appelle un **booléen**, un chiffre binaire ou encore un **bit** (binary digit).

Si bien qu'un tel circuit à deux états s'appelle un circuit mémoire un bit.

L'état d'un circuit mémoire un bit se décrit donc par le symbole 0 ou par le symbole 1. L'état d'un circuit composé de plusieurs tels circuits se décrit par une suite finie de 0 et de 1, que l'on appelle un mot.

Par exemple, le mot 100 décrit l'état d'un circuit composé de trois circuits mémoires un bit, respectivement dans l'état 1, 0 et 0.

Présentation

Exercice 1 :

On imagine un ordinateur dont la mémoire est constituée de quatre circuits mémoire un bit.

a) Quel est le nombre d'états possibles de la mémoire de cet ordinateur ?

b) Même question pour un ordinateur dont la mémoire est constituée de dix circuits mémoire un bit.

c) Et pour un ordinateur dont la mémoire est constituée de 32 milliards de tels circuits.

Présentation

Exercice 2 :

On veut représenter chacune des sept couleurs de l'arc en ciel par un mot, les sept mots devant être distincts et de même longueur.

Quelle est la longueur minimale de ces mots ?

La représentation des entiers naturels

Notation à position et bases de numération

Depuis le Moyen Age, on écrit les nombres entiers naturels en notation décimale à position. Cela signifie que, pour écrire le nombre entier naturel n, on commence par imaginer n objets, que l'on groupe par paquets de dix, puis on groupe ces paquets de dix objets en paquets de dix paquets, etc. A la fin, il reste entre zéro et neuf objets isolés, entre zéro et neuf paquets isolés de dix objets, entre zéro et neuf paquets isolés de cent, etc.

Et on écrit cet entier naturel en écrivant de droite à gauche, le nombre d'objets isolés, le nombre de paquets de dix, le nombre de paquets de cent, le nombre de paquets de mille, etc.

Notation à position et bases de numération

Chacun de ces nombres étant compris entre zéro et neuf, seuls dix chiffres sont nécessaires : 0, 1, 2, 3, 4, 5, 6, 7, 8 et 9.

Par exemple, l'écriture 2359 exprime un entier naturel forme de 9 unités, 5 dizaines, 3 centaines et 2 milliers.

Le choix de faire des paquets de dix est arbitraire : on aurait pu tout aussi bien décider de faire des paquets de deux, de cinq, de douze, de vingt, de soixante, etc.

On écrirait alors les nombres entiers naturels en notation à position *en* ***base deux, cinq, douze, vingt*** **ou** ***soixante***.

La notation décimale à position s'appelle donc aussi la ***notation à position en base dix***.

Notation à position et bases de numération

En *notation binaire*, c'est-à-dire en notation à position en base deux, le nombre treize s'écrit 1101 : de droite à gauche, 1 unité, 0 deuzaine, 1 quatraine et 1 huitaine.

L'écriture d'un entier naturel en binaire est plus longue que son écriture en base dix, en moyenne 3.2 fois plus longue, mais elle ne demande d'utiliser que deux chiffres : 0 et 1.

Notation à position et bases de numération

Exercice 3 :

Un horloger excentrique a eu l'idée de fabriquer une montre sur laquelle l'heure est indiquée par 10 diodes électroluminescentes appelées 1 h, 2 h, 4 h, 8 h, 1 mn, 2 mn, 4 mn, 8 mn, 16 mn et 32 mn.

Pour connaitre l'heure, il suffit d'ajouter la valeur de toutes les diodes allumées.

a) Quelle heure est-il quand sont allumées les diodes 1 h, 2 h, 4 h, 1 mn, 2 mn, 8 mn, 16 mn et 32 mn ?

b) Quelles sont les diodes sont allumées à 5 h 55 ?

c) Est-il possible de représenter toutes les heures ?

d) Toutes les configurations sont-elles la représentation d'une heure ?

La représentation des entiers naturels

La base cinq

Pour comprendre comment transformer un entier naturel écrit en base dix dans une autre base, on commence par le cas de la base cinq, moins particulière que la base deux.

Pour écrire les entiers naturels en base cinq, on a besoin de cinq chiffres : 0, 1, 2, 3, 4.

Quand on a n objets, on les groupe par paquets de cinq, puis on groupe ces paquets en paquets de cinq paquets, etc.

Autrement dit, on fait une succession de divisions par 5, jusqu'à obtenir un quotient égal à 0.

La représentation des entiers naturels

La base cinq

Exercice 4 :
Trouver la représentation en base cinq du nombre 944.

Exercice 5 :
Trouver la représentation en base dix du nombre 401302 (en base cinq).

La base deux

Les nombres exprimés en base deux sont plus difficiles à lire, car il n'y a que deux chiffres, 0 et 1, mais le principe de la numération en base deux est en tout point similaire à celui de la numération en base cinq.

Exercice 6 :

Chercher sur le Web l'année de la mort de Charlemagne.

Trouver la représentation en base deux de ce nombre.

La représentation des entiers naturels

La base deux

Exercice 7 :

Donner les représentations en base deux des nombres 1, 3, 7, 15, 31 et 63. Expliquer le résultat.

Exercice 8 :

C'est en 11110010000 qu'a été démontré le théorème fondamental de l'informatique. Exprimer ce nombre en base dix.

La représentation des entiers relatifs

Pour représenter les entiers relatifs en notation binaire, on doit étendre la représentation aux nombres négatifs.

Une solution est de réserver un bit pour le signe de l'entier à représenter et d'utiliser les autres pour représenter sa valeur absolue.

Ainsi, avec des mots de 16 bits, en utilisant 1 bit pour le signe et 15 bits pour la valeur absolue, on pourrait représenter les entiers relatifs de – 111 1111 1111 1111 = – (215 -1) = – 32767 à 111 1111 1111 1111 = 215 – 1 = 32 767.

La représentation des entiers relatifs

Mais cette méthode a plusieurs inconvénients, l'un d'eux étant qu'il y a deux zéros, l'un positif et l'autre négatif.

On utilise alors une autre méthode, qui consiste à représenter un entier relatif par un entier naturel.

Si on utilise des mots de 16 bits, on peut représenter les entiers relatifs compris entre –32 768 et 32 767 :

- on représente un entier relatif x positif ou nul comme l'entier naturel x
- et un entier relatif x strictement négatif, comme l'entier naturel $x + 2^{16} = x + 65\,536$, qui est compris entre 32768 et 65535.

La représentation des entiers relatifs

Ainsi les entiers naturels de 0 à 32 767 servent à représenter les entiers relatifs positifs ou nuls, et les entiers naturels de 32 768 à 65535 les entiers relatifs strictement négatifs.

Cette manière de représenter les entiers relatifs s'appelle la notation en ***complément à deux***.

L'entier relatif –1 est représenté comme l'entier naturel 65535, c'est-à-dire par le mot 1111 1111 1111 1111.

Avec des mots de seize bits, on peut donc représenter les entiers relatifs compris entre $-2^{15} = -32768$ et $2^{15} - 1 = 32767$.

La représentation des entiers relatifs

Ainsi les entiers naturels de 0 à 32 767 servent à représenter les entiers relatifs positifs ou nuls, et les entiers naturels de 32 768 à 65535 les entiers relatifs strictement négatifs.

Cette manière de représenter les entiers relatifs s'appelle la notation en ***complément à deux***.

L'entier relatif –1 est représenté comme l'entier naturel 65535, c'est-à-dire par le mot 1111 1111 1111 1111.

Avec des mots de seize bits, on peut donc représenter les entiers relatifs compris entre $-2^{15} = -32768$ et $2^{15} - 1 = 32767$.

Plus généralement, avec des mots de *n* bits, on peut représenter les entiers relatifs compris entre -2^{n-1} et $2^{n-1} - 1$.

La représentation des entiers relatifs

Exercice 9 :

a) Quels entiers relatifs peut-on représenter avec des mots de 8 bits ?

b) Combien sont-ils ?

c) Mêmes questions avec des mots de 16 bits, 32 bits et 64 bits.

La représentation des nombres à virgule

Comme la notation décimale, la notation binaire permet aussi de représenter des nombres à virgule. En notation décimale, les chiffres à gauche de la virgule représentent des unités, des dizaines, des centaines, etc. et ceux à droite de la virgule, des dixièmes, des centièmes, des millièmes, etc.

De même, en binaire, les chiffres à droite de la virgule représentent des demis, des quarts, des huitièmes, des seizièmes, etc.

On peut ainsi représenter, par exemple, le nombre un et un quart : 1.01.

Mais cette manière de faire ne permet pas de représenter des nombres très grands ou très petits comme le nombre d'Avogadro ou la constante de Planck.

On utilise donc une autre représentation similaire à la « notation scientifique » des calculatrices, sauf qu'elle est en base deux et non en base dix.

Un nombre est représenté sous la forme **s m** 2^{n} où s est le **signe** du nombre, **n** son exposant et **m** sa mantisse.

Le signe est + ou –, l'exposant est un entier relatif et la mantisse est un nombre à virgule, compris entre 1 inclus et 2 exclu.

La représentation des nombres à virgule

Par exemple, quand on utilise 64 bits pour représenter un nombre à virgule, on utilise 1 bit pour le signe, 11 bits pour l'exposant et 52 bits pour la mantisse.

Le signe + est représenté par 0 et le signe – par 1.

L'exposant *n* est un entier relatif compris entre –1024 et 1023, on le représente comme l'entier naturel *n* + 1023, qui est compris entre 1 et 2046.

Les deux entiers naturels 0 et 2047 sont réservés pour des situations exceptionnelles (+∞, -∞, NaN).

La mantisse *m* est un nombre binaire à virgule compris entre 1 inclus et 2 exclu, comprenant 52 chiffres après la virgule.

Comme cette mantisse est comprise entre 1 et 2, elle a toujours un seul chiffre avant la virgule et ce chiffre est toujours un 1, il est donc inutile de le représenter et on utilise les 52 bits pour représenter les 52 chiffres après la virgule.

La représentation des nombres à virgule

Exercice 10 :

Trouver le nombre à virgule représenté par le mot :

1100010001101001001111000011100000000000000000000000000000000000

Exercice 11 :

A chaque multiplication de deux nombres à virgule, on arrondit le calcul en ne gardant que 52 chiffres après la virgule, ce qui introduit une erreur relative de l'ordre de 2^{-52}.

Quelle est la valeur de cette erreur en base dix ?

Si on fait plusieurs multiplications ces erreurs s'accumulent.

Quelle est l'erreur relative d'un calcul qui est formé d'un million de multiplications, qui dure quelques millisecondes sur un ordinateur usuel ?