# La représentation des données

# La représentation des nombres entiers positifs

**Problème :** On souhaite représenter des nombres naturels à l'aide de $n$ bits.

**Solution :** Il suffit d'encoder les nombres en base 2 :

- On attribue à chaque bit une *position* de 0 à $n-1$. Par convention, on procède de droite à gauche;
- On affecte au bit de position $k$ le *poids* $2^k$.

Le nombre représenté par la suite de bits $b_{n-1}b_{n-2}\ldots b_1b_0$ est donc égal à

$$\sum_{i=0}^{n-1} b_i 2^i.$$

Ce procédé porte le nom de *représentation binaire non signée* des nombres.

**Exemple :** la représentation binaire non signée 10110101 dénote le nombre 181 :

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Position | : | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Poids | : | $2^7$ | $2^6$ | $2^5$ | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ |
| | | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |

On a en effet

$$\begin{aligned}\sum_{i=0}^{7} b_i 2^i &= 2^7 + 2^5 + 2^4 + 2^2 + 2^0 \\ &= 128 + 32 + 16 + 4 + 1 \\ &= 181.\end{aligned}$$

**Note :** Les bits situés aux positions 0 et $n-1$ sont respectivement appelés *bit de poids faible* et *bit de poids fort*.

# Calcul de la représentation d'un nombre

La représentation d'un nombre $v$ peut se calculer grâce aux deux propriétés suivantes :

- Le bit de poids faible est égal à 0 si $v$ est pair, et à 1 si $v$ est impair;
- En retirant le bit de poids faible d'une représentation de $v$, on obtient une représentation de $\lfloor v/2 \rfloor$.

On a donc l'algorithme suivant :

1. Si $v$ est pair, écrire 0. Sinon, écrire 1;
2. Remplacer $v$ par $\lfloor v/2 \rfloor$;
3. Répéter les deux opérations précédentes tant que $v \neq 0$.

**Remarques :**

- Cet algorithme génère les bits de la représentation de $v$ en commençant par le bit de poids faible (c'est-à-dire de la droite vers la gauche);
- La suite de bits obtenue constitue la représentation la plus courte du nombre $v$. Des représentations plus longues s'obtiennent en préfixant le résultat d'un nombre quelconque de zéros.

**Exemple :** Représentation du nombre 109 :

$$\begin{array}{ccc}
v = 109 & \text{impair} & \longrightarrow & \mathbf{1} \\
v = 54 & \text{pair} & \longrightarrow & \mathbf{0} \\
v = 27 & \text{impair} & \longrightarrow & \mathbf{1} \\
v = 13 & \text{impair} & \longrightarrow & \mathbf{1} \\
v = 6 & \text{pair} & \longrightarrow & \mathbf{0} \\
v = 3 & \text{impair} & \longrightarrow & \mathbf{1} \\
v = 1 & \text{impair} & \longrightarrow & \mathbf{1} \\
v = 0. & & &
\end{array}$$

La représentation obtenue est donc 1101101. Il est permis d'ajouter un nombre arbitraire de zéros en tête de cette représentation.

## Les valeurs représentables

A l'aide de $n$ bits, il n'est pas possible de représenter plus de $2^n$ valeurs distinctes.

L'algorithme de calcul de la représentation d'un nombre $v$ s'arrête après avoir produit $n$ bits ou moins si et seulement si $v < 2^n$.

Les nombres possédant une représentation binaire non signée sur $n$ bits sont donc les éléments de l'intervalle

$$[0, \ldots, 2^n - 1].$$

# L'arithmétique binaire non signée

Les opérations d'addition et de multiplication de nombres entiers non signés peuvent s'effectuer selon les règles du calcul écrit.

Les tables d'addition binaire sont les suivantes (les reports sont dénotés par un rectangle) :

$$
\begin{array}{rr} & 0 \\ + & 0 \\ \hline & 0 \end{array}
\qquad
\begin{array}{rr} & 0 \\ + & 1 \\ \hline & 1 \end{array}
\qquad
\begin{array}{rr} & 1 \\ + & 0 \\ \hline & 1 \end{array}
\qquad
\begin{array}{rr} & 1 \\ + & 1 \\ \hline \boxed{1} & 0 \end{array}
$$

$$
\begin{array}{rr} & \boxed{1} \\ & 0 \\ + & 0 \\ \hline & 1 \end{array}
\qquad
\begin{array}{rr} & \boxed{1} \\ & 0 \\ + & 1 \\ \hline \boxed{1} & 0 \end{array}
\qquad
\begin{array}{rr} & \boxed{1} \\ & 1 \\ + & 0 \\ \hline \boxed{1} & 0 \end{array}
\qquad
\begin{array}{rr} & \boxed{1} \\ & 1 \\ + & 1 \\ \hline \boxed{1} & 1 \end{array}
$$

L'opération d'addition s'effectue bit par bit, en commençant par le bit de poids faible.

**Exemple :** Calcul de la somme $123 + 456 = 579$ :

$$
\begin{array}{ccccccccccc}
 & \boxed{1} & \boxed{1} & \boxed{1} & \boxed{1} & \boxed{1} & \boxed{1} & & & & \\
 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\
+ & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\
\hline
 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1
\end{array}
$$

# La multiplication de nombres binaires non signés

Le calcul d'un produit s'effectue selon des règles analogues à celles du calcul décimal :

1. Des produits partiels sont successivement calculés pour chaque bit du multiplicateur, et convenablement alignés;
2. Ces produits partiels sont ensuite additionnés.

La table de multiplication binaire est la suivante :

$$\begin{array}{r} 0 \\ \times\ 0 \\ \hline 0 \end{array} \qquad \begin{array}{r} 0 \\ \times\ 1 \\ \hline 0 \end{array} \qquad \begin{array}{r} 1 \\ \times\ 0 \\ \hline 0 \end{array} \qquad \begin{array}{r} 1 \\ \times\ 1 \\ \hline 1 \end{array}$$

**Exemple :** Calcul du produit $34 \times 12 = 408$ :

$$
\begin{array}{cccccccccc}
 & & & & 1 & 0 & 0 & 0 & 1 & 0 \\
\times & & & & & & 1 & 1 & 0 & 0 \\
\hline
 & & & & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & 0 & 0 & 0 & 0 & 0 & 0 & \\
 & & 1 & 0 & 0 & 0 & 1 & 0 & & \\
+ & 1 & 0 & 0 & 0 & 1 & 0 & & & \\
\hline
 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0
\end{array}
$$

# La représentation hexadécimale

La représentation binaire utilisée par les ordinateurs est peu commode pour l'homme. Dans certains cas, il est cependant indispensable de pouvoir raisonner sur la représentation interne des données. On utilise alors la représentation *hexadécimale* (c'est-à-dire en base 16), qui présente deux avantages :

- Elle est concise;
- Les conversions de l'hexadécimal vers le binaire et vice-versa sont immédiates.

Un chiffre hexadécimal peut prendre 16 valeurs : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Un tel chiffre représente donc exactement 4 bits d'information.

Pour convertir un nombre hexadécimal en binaire, il suffit de remplacer chaque chiffre par la séquence de 4 bits qui lui correspond. La conversion réciproque est similaire.

**Table de conversion :**

| Hexadécimal | Binaire | Hexadécimal | Binaire |
|---|---|---|---|
| 0 | 0000 | 8 | 1000 |
| 1 | 0001 | 9 | 1001 |
| 2 | 0010 | $A$ | 1010 |
| 3 | 0011 | $B$ | 1011 |
| 4 | 0100 | $C$ | 1100 |
| 5 | 0101 | $D$ | 1101 |
| 6 | 0110 | $E$ | 1110 |
| 7 | 0111 | $F$ | 1111 |

**Note :** Lorsque le contexte ne permet pas de déterminer la base choisie pour représenter les nombres, on ajoute le suffixe "h" ou le préfixe "0x" aux représentations hexadécimales, et le suffixe "b" ou le préfixe "0b" aux représentations binaires.

**Exemple :** On a
0xCAFE007 = 1100101011111110000000000111b = 212852743.

# La représentation des nombres entiers signés

Il existe plusieurs procédés permettant de représenter des nombres entiers positifs et négatifs :

- La représentation par *valeur signée*;
- La représentation par *complément à un*;
- La représentation par *complément à deux*.

Ces trois méthodes possèdent des points communs :

- Le *signe* d'un nombre est représenté par le bit de poids fort (ici appelé *bit de signe*). Celui-ci est égal à
  - 0 pour les nombres positifs;
  - 1 pour les nombres négatifs.
- La représentation d'un nombre positif est toujours identique à sa représentation binaire non signée de même taille.

# La représentation par valeur signée

**Principe :** A la suite du bit de signe, on place la représentation binaire non signée de la valeur absolue du nombre représenté.

**Exemple :** La représentation sur 8 bits du nombre $-42$ est égale à 10101010. En effet

- Ce nombre est négatif, donc le bit de signe est égal à 1;
- La représentation binaire non signée sur 7 bits de $42 = |-42|$ est 0101010.

Selon ce procédé, le nombre $v$ représenté par le groupe de bits $b_{n-1}b_{n-2}\dots b_1b_0$ est égal à

$$v = (1 - 2b_{n-1}) \sum_{i=0}^{n-2} b_i 2^i.$$

# Les valeurs représentables

A l'aide de $n$ bits, la représentation par valeur signée permet d'encoder

- tous les éléments de l'intervalle $[0, \ldots, 2^{n-1} - 1]$ (bit de signe égal à 0), et
- tous les éléments de l'intervalle $[-2^{n-1} + 1, \ldots, 0]$ (bit de signe égal à 1).

L'ensemble des valeurs représentables est donc l'intervalle

$$[-2^{n-1} + 1, \ldots, 2^{n-1} - 1].$$

**Remarques :**

- Le nombre 0 possède deux représentations distinctes;
- Ce procédé rend difficile le calcul des opérations arithmétiques.

# La représentation par complément à un

**Principe :** La représentation d'un nombre est similaire à sa représentation par valeur signée, mais les bits qui suivent le bit de signe sont *complémentés* (0 est remplacé par 1, et vice-versa).

**Exemple :** La représentation sur 8 bits du nombre $-42$ est égale à 11010101. En effet

- Ce nombre est négatif, donc le bit de signe est égal à 1;
- La représentation binaire non signée sur 7 bits de $42 = |-42|$ est 0101010, dont le complément est 1010101.

L'ensemble des nombres représentables à l'aide de $n$ bits est identique à celui obtenu pour la représentation par valeur signée, soit l'intervalle

$$[-2^{n-1} + 1, \dots, 2^{n-1} - 1].$$

Selon ce procédé, le nombre $v$ représenté par le groupe de bits $b_{n-1}b_{n-2}\ldots b_1b_0$ est égal à

$$(1-2^{n-1})b_{n-1}+\sum_{i=0}^{n-2} b_i 2^i.$$

En effet,

- Si $v > 0$, on a $b_{n-1}=0$ et

$$v=\sum_{i=0}^{n-2} b_i 2^i;$$

- Si $v < 0$, on a $b_{n-1}=1$. La suite de bits $\overline{b_{n-2}}\cdots\overline{b_1}\,\overline{b_0}$ forme la représentation binaire non signée du nombre

$$2^{n-1}-1-\sum_{i=0}^{n-2} b_i 2^i.$$

On a donc bien

$$v=-|v|=1-2^{n-1}+\sum_{i=0}^{n-2} b_i 2^i.$$

# L'arithmétique des nombres représentés par complément à un

Les algorithmes de calcul arithmétique sur les nombres non signés peuvent facilement être adaptés à la représentation par complément à un.

L'addition de deux nombres signés représentés à l'aide de $n$ bits s'effectue de la façon suivante :

1. On additionne les deux nombres comme s'il s'agissait de représentations non signées;
2. Si l'opération conduit à un report à la position $n$, on supprime ce report et on ajoute 1 à la somme calculée.

**Exemples :** Calcul de la somme $12 + (-34) = -22$ :

$$
\begin{array}{ccccccccc}
 & & & \boxed{1} & \boxed{1} & \boxed{1} & & & \\
 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
+ & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\
\hline
 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1
\end{array}
$$

Calcul de la somme $-12 + (-34) = -46$ :

$$
\begin{array}{ccccccccc}
\boxed{1} & \boxed{1} & \boxed{1} & \boxed{1} & \boxed{1} & \boxed{1} & \boxed{1} & \boxed{1} & \\
 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\
+ & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\
\hline
 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\
+ & & & & & & & & 1 \\
\hline
 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1
\end{array}
$$

# La représentation par complément à deux

**Principes :** La représentation d'un nombre $v$ sur $n$ bits est égale

1. au bit de signe 0 suivi de la représentation entière non signée de $v$ sur $n-1$ bits si $v \geq 0$;
2. à la représentation par complément à un de $v+1$ sur $n$ bits si $v < 0$. On dit alors que les $n$ bits ainsi obtenus forment le *complément à deux* des $n$ bits encodant le nombre positif $-v$.

**Exemples :**

- La représentation sur 8 bits du nombre $-42$ est égale à 11010110;
- La représentation sur $n$ bits du nombre $-1$ est composée de $n$ bits égaux à 1.

**Propriétés :**

- Le nombre $v$ représenté par le groupe de bits $b_{n-1}b_{n-2}\ldots b_1b_0$ est égal à

$$-2^{n-1}b_{n-1} + \sum_{i=0}^{n-2} b_i 2^i;$$

- La représentation d'un nombre à l'aide d'un nombre de bits donné est unique. En particulier, le nombre 0 possède une seule représentation;

- L'ensemble des nombres représentables à l'aide de $n$ bits forme l'intervalle

$$[-2^{n-1}, \ldots, 2^{n-1} - 1].$$

## L'arithmétique binaire signée

Les opérations arithmétiques sont faciles à effectuer sur des nombres représentés par la méthode du complément à deux :

- L'addition de deux nombres de $n$ bits s'effectue par le même algorithme que celui employé dans le cas des nombres non signés. Les reports apparaissant à la position $n$ sont simplement ignorés;
- La soustraction de deux nombres s'effectue en ajoutant au premier l'opposé du deuxième. La représentation du nombre $-v$ est égale au complément à deux de la représentation de $v$;

# Exemples

Calcul de la somme $12 + (-34) = -22$ :

$$
\begin{array}{ccccccccc}
 & & & \boxed{1} & \boxed{1} & \boxed{1} & & & \\
 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
+ & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\
\hline
 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0
\end{array}
$$

Calcul de la somme $-12 + (-34) = -46$ :

$$
\begin{array}{ccccccccc}
\boxed{1} & \boxed{1} & \boxed{1} & \boxed{1} & \boxed{1} & \boxed{1} & & & \\
 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 \\
+ & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\
\hline
 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0
\end{array}
$$

# Récapitulatif

Le tableau suivant reprend les différentes représentations des nombres entiers à l'aide de 4 bits :

| Bits | N.s. | V.s. | C.1 | C.2 |
|---|---|---|---|---|
| 0000 | 0 | 0 | 0 | 0 |
| 0001 | 1 | 1 | 1 | 1 |
| 0010 | 2 | 2 | 2 | 2 |
| 0011 | 3 | 3 | 3 | 3 |
| 0100 | 4 | 4 | 4 | 4 |
| 0101 | 5 | 5 | 5 | 5 |
| 0110 | 6 | 6 | 6 | 6 |
| 0111 | 7 | 7 | 7 | 7 |
| 1000 | 8 | 0 | $-7$ | $-8$ |
| 1001 | 9 | $-1$ | $-6$ | $-7$ |
| 1010 | 10 | $-2$ | $-5$ | $-6$ |
| 1011 | 11 | $-3$ | $-4$ | $-5$ |
| 1100 | 12 | $-4$ | $-3$ | $-4$ |
| 1101 | 13 | $-5$ | $-2$ | $-3$ |
| 1110 | 14 | $-6$ | $-1$ | $-2$ |
| 1111 | 15 | $-7$ | 0 | $-1$ |

# La représentation des caractères

Il existe plusieurs standards de codification des caractères alphanumériques.

**Le code ASCII**

Ce standard est à la base d'une grande majorité des encodages actuellement utilisés.

**Principes :**

- Un caractère est encodé à l'aide de 7 bits d'information. On peut donc attribuer à chaque symbole un code dans l'intervalle $[0, \ldots, 127]$;
- Les codes 0x00 à 0x1F représentent des *caractères de contrôle*. L'interprétation de ces caractères peut dépendre du système utilisé;

- Les codes 0x20 à 0x3F correspondent aux symboles mathématiques, à la ponctuation et aux chiffres. Le code du chiffre $n$ est égal à 0x3$n$;

  **Remarque :** La valeur d'un chiffre est donc égale aux quatre bits de poids faible de son encodage ASCII.

- Les codes 0x40 à 0x5F contiennent les lettres majuscules et quelques symboles spéciaux. Les lettres sont classées par ordre alphabétique et possèdent des codes consécutifs, ce qui facilite les opérations de comparaison entre chaînes de caractères;

- Les codes 0x60 à 0x7F contiennent les lettres minuscule, un caractère de contrôle (0x7F) et quelques symboles spéciaux.

  **Note :** Les codes d'une lettre majuscule et minuscule partagent les mêmes 5 bits de poids faible.

**Table des caractères imprimables ASCII :**

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 20 | | 30 | 0 | 40 | @ | 50 | P | 60 | ‘ | 70 | p |
| 21 | ! | 31 | 1 | 41 | A | 51 | Q | 61 | a | 71 | q |
| 22 | ” | 32 | 2 | 42 | B | 52 | R | 62 | b | 72 | r |
| 23 | # | 33 | 3 | 43 | C | 53 | S | 63 | c | 73 | s |
| 24 | $ | 34 | 4 | 44 | D | 54 | T | 64 | d | 74 | t |
| 25 | % | 35 | 5 | 45 | E | 55 | U | 65 | e | 75 | u |
| 26 | & | 36 | 6 | 46 | F | 56 | V | 66 | f | 76 | v |
| 27 | ’ | 37 | 7 | 47 | G | 57 | W | 67 | g | 77 | w |
| 28 | ( | 38 | 8 | 48 | H | 58 | X | 68 | h | 78 | x |
| 29 | ) | 39 | 9 | 49 | I | 59 | Y | 69 | i | 79 | y |
| 2A | * | 3A | : | 4A | J | 5A | Z | 6A | j | 7A | z |
| 2B | + | 3B | ; | 4B | K | 5B | [ | 6B | k | 7B | { |
| 2C | , | 3C | ¡ | 4C | L | 5C | \ | 6C | l | 7C | — |
| 2D | - | 3D | = | 4D | M | 5D | ] | 6D | m | 7D | } |
| 2E | . | 3E | ¿ | 4E | N | 5E | ˆ | 6E | n | 7E | ˜ |
| 2F | / | 3F | ? | 4F | O | 5F | _ | 6F | o | | |

## La norme ISO Latin-1

Ce standard représente un caractère à l'aide de 8 bits. Il est basé sur le code ASCII dont il reprend les 128 premiers caractères.

Les 128 codes supplémentaires (possédant un bit de poids fort égal à 1) permettent de représenter les caractères accentués les plus utilisés par les langues européennes.

Le standard ISO Latin-1 est celui utilisé par défaut pour l'interprétation des pages du *World-Wide Web*.

# Le standard Unicode

Certaines langues orientales nécessitent un jeu de plus de 256 caractères.

Le standard *Unicode* représente chaque caractère à l'aide de 16 bits d'information. Ses caractéristiques principales sont les suivantes :

- Il incorpore les jeux de caractères de plusieurs autres standards de représentation. Notamment, ses 256 premiers caractères sont identiques à ceux de la norme ISO Latin-1.
- Il contient un jeu de caractères unifié commun à plusieurs langues orientales.
- Il permet la superposition de caractères.

Actuellement, le standard Unicode définit environ 49000 caractères, et couvre la plupart des langues écrites courantes et historiques.

Il est probable que ce standard deviendra rapidement prédominant.