

# Codage des nombres

## 1 Codage des entiers naturels

Soit  $b \in \mathbb{N} \setminus \{0, 1\}$ . On note  $\Sigma = [0..b - 1]$ . Les éléments de  $\Sigma$  seront appelés les chiffres et les mots sur  $\Sigma$  seront appelés des nombres.

### 1.1 Codage en base b

#### 1.1.1 Définition

$val_b = \left( \begin{array}{ccc} \Sigma^* & \rightarrow & \mathbb{N} \\ a_{l-1}a_{l-2}...a_0 & \mapsto & \sum_{i=0}^{l-1} a_i b^i \end{array} \right)$  où  $\Sigma^* = \bigcup_{i \in \mathbb{N}} \Sigma^i$ , c'est-à-dire l'ensemble des suites de longueur  $i$  dans  $\Sigma$  pour  $i \in \mathbb{N}$ .

**Remarque :** le mot vide sera noté  $\varepsilon$  ( $\Sigma^0 = \{\varepsilon\}$ ) et on a  $val_b(\varepsilon) = 0$ .

**Exemple :**  $val_{10}(123) = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 = 100 + 20 + 3 = 123$

$val_2(100) = 1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0$

#### 1.1.2 Taille du codage

**Lemme :** Soit  $l \in \mathbb{N}$ .  $\forall a \in \Sigma^l$ ,  $val_b(a) \in [0..b^l[$ .

▷ Soit  $(a_{l-1}, a_{l-2}, \dots, a_0) \in \Sigma^l$ .

$val_b(a_{l-1}a_{l-2}...a_0) = \sum_{i=0}^{l-1} a_i b^i$ . Or  $\forall i \in [0..l-1]$ ,  $a_i \leq (b-1)$ .

Donc  $val_b(a_{l-1}a_{l-2}...a_0) \leq \sum_{i=0}^{l-1} (b-1)b^i = \sum_{i=0}^{l-1} b^{i+1} - \sum_{i=0}^{l-1} b^i = \sum_{i=1}^l b^i - \sum_{i=0}^{l-1} b^i = b^l - b^0 = b^l - 1$

**Propriété :** Soit  $n \in \mathbb{N}$ .

Il faut  $\lceil \log_b(n+1) \rceil$  chiffres au minimum pour écrire  $n$  en base  $b$ .

▷ On note  $l = \lceil \log_b(n+1) \rceil$ .

Par l'absurde, supposons que  $a_{l-1}...a_0 \in \Sigma^{l'}$  représente  $n$  en base  $b$  avec  $l' < l$  chiffres.

On a  $l' < \log_b(n+1)$  par définition de la partie entière supérieure comme plus petit majorant entier.

$n = val_b(a_{l'-1}a_{l'-2}...a_0) \leq b^{l'} - 1 < n+1$ . Or  $b^{l'} < b^{\log_b(n+1)} = n+1$ . Donc  $n < n+1-1 \iff n < n$ .  
ABSURDE.

#### 1.1.3 Existence

**Remarque :** Pour tout  $n \in \mathbb{N}$ , il existe  $a_{l-1}a_{l-2}...a_0 \in \Sigma^l$  tel que  $val_b(a_{l-1}...a_0) = n$ . Plus précisément, tout entier  $n \in \mathbb{N}$  admet une écriture en base  $b$  à  $\lceil \log_b(n+1) \rceil$  chiffres.

▷ Montrons par récurrence sur  $l \in \mathbb{N}$  la propriété  $\mathcal{P}_l$  : " $\forall n \in [0..b^l - 1]$ ,  $n$  admet une écriture en base  $b$  à  $l$  chiffres".

$\forall n \in [0..b^l - 1]$ , c'est-à-dire  $\forall n \in [0..0]$ ,  $n$  admet une écriture en base  $b$  à 0 chiffres. En effet, le seul nombre à 0 chiffre est le mot vide  $\varepsilon$ . Par convention,  $val_b(\varepsilon) = 0 \in [0..0]$ . Donc  $\mathcal{P}_0$  est vraie.

Pour un  $l$  fixé, supposons  $\mathcal{P}_l$ . Soit  $n \in [0..b^{l+1} - 1]$ . Par définition de la division euclidienne, il existe  $(q, r) \in \mathbb{N}^2$  tel que  $n = b^l q + r$  et  $r < b^l$ , i.e.  $r \in [0..b^l - 1]$ . Par  $\mathcal{P}_l$ , on en déduit que  $r$  admet une écriture en base  $b$  à  $l$  chiffres qu'on note  $(a_i)_{i \in [0..l]}$ . On a alors  $r = \sum_{i=0}^{l-1} a_i b^i$ , et donc  $n = q b^l + \sum_{i=0}^{l-1} a_i b^i$ .

Puisque  $n < b^{l+1}$ , on a nécessairement  $q < b$  (sinon on aurait  $n \geq q b^l > b \times b^l = b^{l+1}$ ). Ainsi, en posant  $a_l = q$ , on a  $(a_i)_{i \in [0..l+1]} \in \Sigma^{l+1}$  et  $n = a_l b^l + \sum_{i=0}^{l-1} a_i b^i$ .

Donc  $n$  admet bien une écriture en base  $b$  à  $l + 1$  chiffres. Donc  $\mathcal{P}_{l+1}$  est vraie.

### 1.1.4 Quasi-unicité

**Propriété :** Soit  $n \in \mathbb{N}$ . Si  $a_{l-1}a_{l-2}...a_0 \in \Sigma^l$  est une écriture de  $n$  en base  $b$  (c'est-à-dire si  $val_b(a_{l-1}...a_0) = n$ ) alors  $\forall k \in [0..l - 1]$ ,  $a_k$  est le reste modulo  $b$  du quotient de  $n$  par  $b^k$ .

**Exemple :**  $b = 10, n = 123, a_2 = 1, a_1 = 2 = (n//10)\%10, a_0 = 3 = n\%10$

▷ Soit  $n \in \mathbb{N}$ , soit  $a_{l-1}...a_0$  une écriture de  $n$  en base  $b$ . Soit  $k \in [0..l - 1]$ . On a :

$$n = val_b(a_{l-1}...a_0) = \sum_{i=0}^{l-1} a_i b^i = \sum_{i=0}^{k-1} a_i b^i + \sum_{i=k}^{l-1} a_i b^i = \sum_{i=0}^{k-1} a_i b^i + \sum_{i=k}^{l-1} a_i (b^k \times b^{i-k}) = \underbrace{\sum_{i=0}^{k-1} a_i b^i}_{=r_k} + b^k \underbrace{\sum_{i=k}^{l-1} a_i b^{i-k}}_{=q_k}$$

On note  $r_k = \sum_{i=0}^{k-1} a_i b^i$ . On a  $r_k \in \mathbb{N}$  et puisque  $\forall i \in [0..l]$ ,  $a_i \in [0..b]$ , on a aussi :

$$r_k \leq \sum_{i=0}^{k-1} (b-1)b^i = \sum_{i=0}^{k-1} b^{i+1} - \sum_{i=0}^{k-1} b^i = b^k - 1 < b^k$$

On note  $q_k = \sum_{i=k}^{l-1} a_i b^{i-k}$ . Puisque  $i - k \geq 0 \forall i \in [k..l]$ , alors  $b^{i-k} \in \mathbb{N}$ , ainsi  $q_k$  est une somme d'entiers positifs donc  $q_k \in \mathbb{N}$ . On en déduit de la première égalité que  $q_k$  est le quotient et  $r_k$  le reste dans la division euclidienne de  $n$  par  $b^k$ . On cherche donc à montrer que  $a_k$  est le reste modulo  $b$  de  $q_k$ . On a :

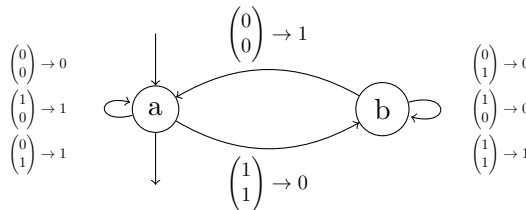
$$q_k = \sum_{i=k}^{l-1} a_i b^{i-k} = a_k \underbrace{b^{k-k}}_{=1} + \sum_{i=k+1}^{l-1} a_i (b^{i-k-1} \times b) = a_k + b \left( \sum_{i=k+1}^{l-1} a_i b^{i-k-1} \right)$$

D'une part on sait que  $a_k < b$  car  $a_k \in \Sigma$ . D'autre part, comme  $i - k - 1 \geq 0 \forall i \in [k+1..l - 1]$ ,  $\sum_{i=k+1}^{l-1} a_i b^{i-k-1} \in \mathbb{N}$ . On en déduit donc de l'égalité précédente que  $a_k$  est bien le reste de  $q_k$  modulo  $b$ .

### 1.1.5 Conclusion

Pour  $l \in \mathbb{N}$ , on note  $ec_b^l$  la fonction qui à un entier de  $[0..b^l]$  associe son écriture en base  $b$  à  $l$  chiffres.

## 1.2 Addition en base 2



### 1.3 Application

Pour  $u \in \mathbb{N}^{\mathbb{N}}$ , on note

$$\mathcal{P}_u \left\| \begin{array}{l} \text{Entrée: } n \in \mathbb{N} \\ \text{Sortie: } u_n \end{array} \right.$$

$\underbrace{A}_{\text{un algo}} \iff \text{suite finie de caractère} \iff \text{suite finie d'entiers entre 0 et 255}$

$\iff \text{un entier écrit en base 256.}$

On note  $\varphi$  la fonction qui à un algorithme  $A$  associe un entier écrit en base 256 en remplaçant les caractères de l'algorithme pour un entier entre 1 et 255.

$$\varphi = \left( \begin{array}{ccc} \text{l'ensemble des textes des algorithmes} & \rightarrow & \mathbb{N} \\ n \in \mathbb{N} & \mapsto & p \end{array} \right)$$

$\varphi$  est injective mais pas surjective.  $\varphi|_{\text{Im}(\varphi)}$  est bijective.

On note pour  $A$  le texte d'un algo qui prend en entrée un entier et qui rend en sortie un entier,  $eval(A, n)$ , la valeur absolue en lançant cet algorithme sur l'entrée  $n$ .

**Remarque :** Si  $A$  résout  $\mathcal{P}_n$  pour  $u \in \mathbb{N}^{\mathbb{N}}$ , alors  $\forall n \in \mathbb{N}$ ,  $eval(A, n) = u_n$ .

On définit  $u \in \mathbb{N}^{\mathbb{N}}$ ,  $u_n = \begin{cases} eval(\varphi^{-1}(n), n) + 1 & \text{si } n \in \text{Im}(\varphi) \\ 0 & \text{sinon} \end{cases}$

On suppose que  $A$  est un algorithme qui résout le problème  $\mathcal{P}_u$ . Alors :

$$\begin{aligned} eval(A_u, \varphi(A_u)) &= u_{\varphi(A_u)} \text{ car } A \text{ résout } \mathcal{P}_u \\ &= eval(\varphi^{-1}(\varphi(A_u)), \varphi(A_u) + 1) \\ &= eval(A_u, \varphi(A_u)) + 1 \end{aligned}$$

ABSURDE. Donc il n'existe pas d'algorithme résolvant  $\mathcal{P}_u$ .

Notation de l'addition en base 2 :

Pour  $l \in \mathbb{N}$  et  $\Sigma = \{0, 1\}$ ,