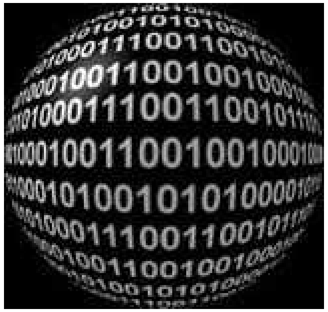
Représenter les nombres...

... autrement!

Choisir sa base…
La numération binaire



- 1 = 1 → 1
- 2 = 2 + 0 → 10
- 3 = 2 + 1 → 11
- 4 = 4 + 0 + 0 → 100
- 5 = 4 + 0 + 1 → 101
- 6 = 4 + 2 + 0 → 110
- 7 = 4 + 2 + 1 → 111
- 8 = 8 + 0 + 0 + 0 → 1000

357 = 256 + 101

357 = 256 + 64 + 37

357 = 256 + 64 + 32 + 5

357 = 256 + 64 + 32 + 4 + 1

$(357)_2 =$ 1 0 1 1 0 0 1 0 1

- Pas beaucoup de symbôles: 0 et 1
- Peu de règles permettent de faire beaucoup de calculs.

  Ex : 3 opérations à connaître pour faire des additions 0 + 0 = 0, 1 + 0 = 1 et 1 + 1 = 10
- On peut parler à un ordinateur avec…

L'addition en base 2...

Pour les additions, juste 3 opérations à connaitre:

$0 + 0 = 0$, $1 + 0 = 0 + 1 = 1$, $1 + 1 = 10$

	357 →	1	0	1	1	0	0	1	0	1
+	269 →	1	0	0	0	0	1	1	0	1

1
2
4
8
16
32
64
128
256
512
1024

L'addition en base 2...

Pour les additions, juste 3 opérations à connaitre:

$0 + 0 = 0$, $1 + 0 = 0 + 1 = 1$, $1 + 1 = 10$

$$
\begin{array}{lrl}
 & & {\scriptstyle 1} \\
 & 357 \rightarrow & 1\,0\,1\,1\,0\,0\,1\,0\,1 \\
+ & 269 \rightarrow & 1\,0\,0\,0\,0\,1\,1\,0\,1 \\ \hline
 & & 0
\end{array}
$$

1
2
4
8
16
32
64
128
256
512
1024

L'addition en base 2...

Pour les additions, juste 3 opérations à connaitre:

$0 + 0 = 0, \quad 1 + 0 = 0 + 1 = 1, \quad 1 + 1 = 10$

$$\begin{array}{rrl} & & {}^{1} \\ & 357 \rightarrow & 1\,0\,1\,1\,0\,0\,1\,0\,1 \\ + & 269 \rightarrow & 1\,0\,0\,0\,0\,1\,1\,0\,1 \\ \hline & & 1\,0 \end{array}$$

1
2
4
8
16
32
64
128
256
512
1024

L'addition en base 2...

Pour les additions, juste 3 opérations à connaitre:

$0 + 0 = 0, \quad 1 + 0 = 0 + 1 = 1, \quad 1 + 1 = 10$

$$
\begin{array}{rrl}
 & & {}^{1} \\
 & 357 \rightarrow & 1\,0\,1\,1\,0\,0\,1\,0\,1 \\
+ & 269 \rightarrow & 1\,0\,0\,0\,0\,1\,1\,0\,1 \\
\hline
 & & 0\,1\,0
\end{array}
$$

1
2
4
8
16
32
64
128
256
512
1024

L'addition en base 2...

Pour les additions, juste 3 opérations à connaitre:

$0 + 0 = 0, \quad 1 + 0 = 0 + 1 = 1, \quad 1 + 1 = 10$

$$
\begin{array}{rrl}
 & & 1 \\
 & 357 \rightarrow & 1\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 1 \\
+ & 269 \rightarrow & 1\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 1 \\
\hline
 & & 0\ 0\ 1\ 0
\end{array}
$$

1
2
4
8
16
32
64
128
256
512
1024

L'addition en base 2...

Pour les additions, juste 3 opérations à connaitre:

$0 + 0 = 0, \quad 1 + 0 = 0 + 1 = 1, \quad 1 + 1 = 10$

$$\begin{array}{rl} 357 \rightarrow & 1\;0\;1\;1\;\overset{1}{0}\;0\;1\;0\;1 \\ +\ 269 \rightarrow & 1\;0\;0\;0\;0\;1\;1\;0\;1 \\ \hline & 1\;0\;0\;1\;0 \end{array}$$

1
2
4
8
16
32
64
128
256
512
1024

L'addition en base 2...

Pour les additions, juste 3 opérations à connaitre:

$0 + 0 = 0, \quad 1 + 0 = 0 + 1 = 1, \quad 1 + 1 = 10$

$$
\begin{array}{rrr}
 & 357 \rightarrow & 1\,0\,1\,1\,0\,0\,1\,0\,1 \\
+ & 269 \rightarrow & 1\,0\,0\,0\,0\,1\,1\,0\,1 \\
\hline
 & & 1\,1\,0\,0\,1\,0
\end{array}
$$

1

2

4

8

16

32

64

128

256

512

1024

L'addition en base 2...

Pour les additions, juste 3 opérations à connaitre:

$0 + 0 = 0,\quad 1 + 0 = 0 + 1 = 1,\quad 1 + 1 = 10$

$$
\begin{array}{rrl}
 & 357 \rightarrow & 1\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 1 \\
+ & 269 \rightarrow & 1\ 0\ 0\ 0\ 0\ 1\ 1\ 0\ 1 \\
\hline
 & & 1\ 1\ 1\ 0\ 0\ 1\ 0
\end{array}
$$

1
2
4
8
16
32
64
128
256
512
1024

L'addition en base 2...

Pour les additions, juste 3 opérations à connaitre:

$0 + 0 = 0, \quad 1 + 0 = 0 + 1 = 1, \quad 1 + 1 = 10$

$$
\begin{array}{rrl}
 & 357 \rightarrow & 1\,0\,1\,1\,0\,0\,1\,0\,1 \\
+ & 269 \rightarrow & 1\,0\,0\,0\,0\,1\,1\,0\,1 \\
\hline
 & & 0\,1\,1\,1\,0\,0\,1\,0
\end{array}
$$

1
2
4
8
16
32
64
128
256
512
1024

L'addition en base 2...

Pour les additions, juste 3 opérations à connaitre:

$0 + 0 = 0, \quad 1 + 0 = 0 + 1 = 1, \quad 1 + 1 = 10$

$$
\begin{array}{rl}
357 \rightarrow & 1\,0\,1\,1\,0\,0\,1\,0\,1 \\
+\ \ 269 \rightarrow & 1\,0\,0\,0\,0\,1\,1\,0\,1 \\
\hline
& 1\,0\,0\,1\,1\,1\,0\,0\,1\,0
\end{array}
$$

1
2
4
8
16
32
64
128
256
512
1024

L'addition en base 2...

1
2
4
8
16
32
64
128
256
512
1024

Pour les additions, juste 3 opérations à connaitre:

0 + 0 = 0,   1 + 0 = 0 + 1 = 1,   1 + 1 = 10

	357 →	1 0 1 1 0 0 1 0 1
+	269 →	1 0 0 0 0 1 1 0 1
		1 0 0 1 1 1 0 0 1 0

On traduit ensuite si besoin:

512+64+32+16+2 = 626

Multiplication en Egypte et base 2

$(357)_2 = \quad 1\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 1$

$357 \times 269 = (256 + 64 + 32 + 4 + 1) \times 269$

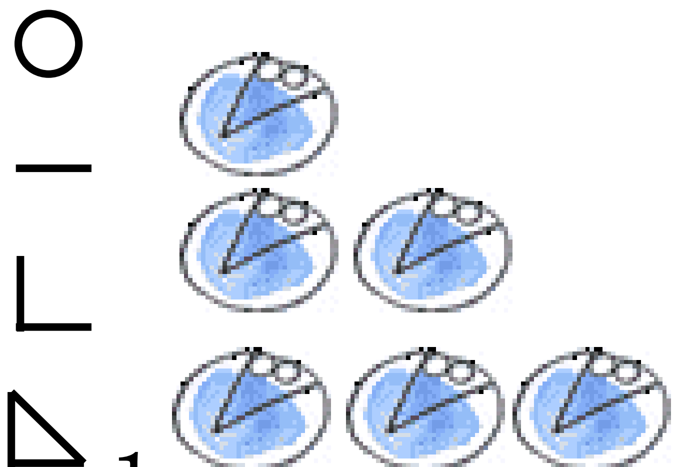
269×1	269
269×2	538
269×2^2	1076
269×2^3	2152
269×2^4	4304
269×2^5	8608
269×2^6	17216
269×2^7	34432
269×2^8	68864

$$\begin{array}{rr} & 269 \\ + & 1076 \\ + & 8608 \\ + & 17216 \\ + & 68864 \\ & 96033 \end{array}$$

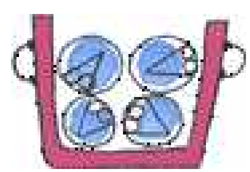
La numération Shadok

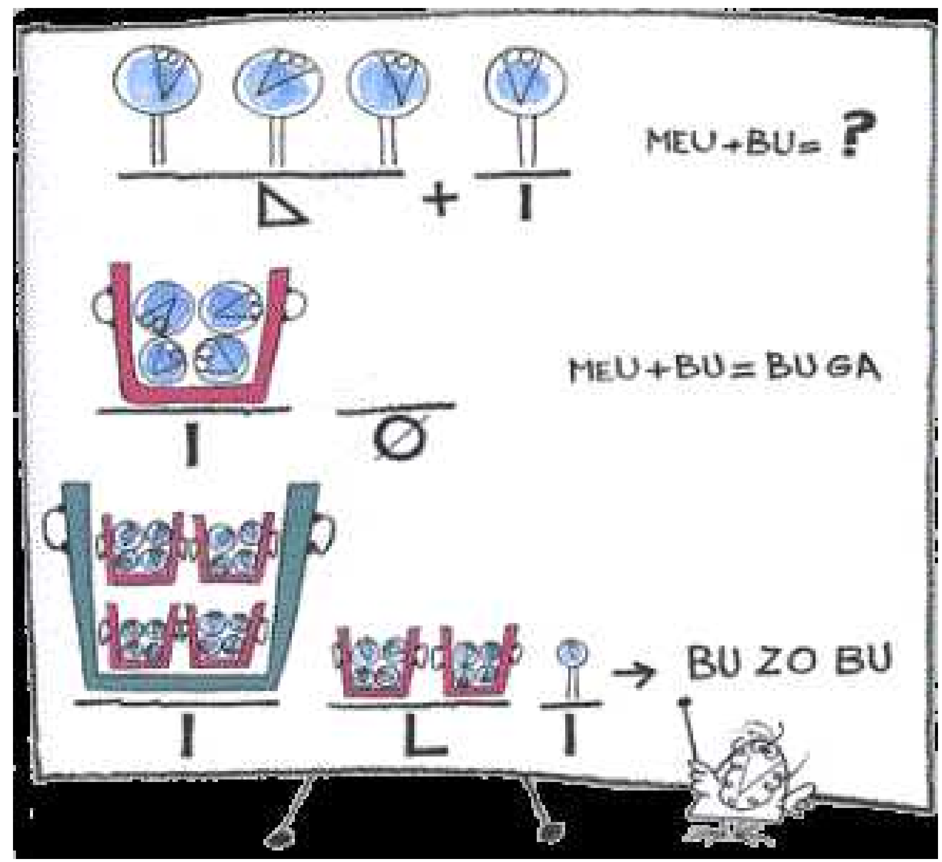
Utilisation de la base 4

- Ga
- Bu
- Zo
- Meu



- 4 = 1

- 4  =  ....



BUGAZOMEU =  $= 1 \times 4^3 + 0 \times 4^2 + 2 \times 4 + 3 = 75$

Des bases exotiques...

$(\Phi_n)_{n\in\mathbb{N}}$ = 1 , 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610,...

$\Phi_{n+2} = \Phi_{n+1} + \Phi_n$

On décompose un nombre en utilisant l'algorithme glouton

357 = 233 + 24

= 233 + 21 + 3 = $\Phi_{11} + \Phi_6 + \Phi_2$

$(357)_\Phi$ = 100001000100

Des bases exotiques...

$(\Phi_n)_{n\in\mathbb{N}} = 1\ , 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610,...$

$\Phi_{n+2} = \Phi_{n+1} + \Phi_n$

On décompose un nombre en utilisant l'algorithme glouton

$357 = 233 + 24$

$= 233 + 21 + 3 = \Phi_{11} + \Phi_6 + \Phi_2 = \Phi_{10} + \Phi_9 + \Phi_6 + \Phi_2$

$(357)_\Phi = 100001000100 = 11001000100$

La recette d'un système de numération?

TAMBOUILLE 1 :

se donner une suite d'entiers $(\Phi_n)_{n\in\mathbb{N}}$

- strictement croissante
- $\Phi_0=1$
- $(\Phi_{n+1}/\Phi_n)_{n\in\mathbb{N}}$ bornée ($\rightarrow$ garantit le nombre fini de symbôles)

Représentations fournies par l'algorithme glouton.

Cela suffit mais ne garantit pas l'unicité de l'écriture !

- Cas standard : Φ_{n+1}/Φ_n constante ou Φ_{n+1}/Φ_n périodique
- L'opération « +1 » est réalisable par un transducteur fini lorsque $(\Phi_n)_{n\in\mathbb{N}}$ vérifie une récurrence linéaire à coefficients entiers.

J. Shallit, *Numeration systems, linear recurrences, and regular sets,* Inform. Comput., 113 (1994), pp. 331-347

La recette d'un système de numération?

TAMBOUILLE 2 :

Se donner une langage L infini sur un alphabet fini ordonné $(A,<)$ totalement ordonné selon l'ordre long-lex:

$u < v$ si et seulement si :

- u est plus court que v
- si u et v sont de même longueur, de prefixe commun de taille r, $u_{r+1} < v_{r+1}$

Représentations des entiers données par l'ordre <.

Cela suffit pour garantir l'unicité de la représentation ... mais ne garantit pas que l'opération « +1 » et plus généralement les additions soit algorithmiquement faciles !

lorsque le langage L est régulier : l'opération « +1 » est réalisée par un transducteur fini.

P. Lecomte, M. Rigo, *Numeration systems on a regular langage*, Theory of Comput. Systems., 34 (2001), pp. 27-44