Αναπαράσταση αριθμών σε ένα αριθμητικό σύστημα

Ένας πραγματικός αριθμός γράφεται: $x = \lfloor x \rfloor + \varepsilon,\ 0 \le \varepsilon < 1$, ή

$$x = \sum_{k=-m}^{n-1} a_k b^k = \underbrace{a_{n-1}b^{n-1} + a_{n-2}b^{n-2} + \cdots + a_1 b^1 + a_0 b^0}_{\text{ακέραιο μέρος}} + \leftarrow \boxed{\lfloor x \rfloor}$$

$$+ \underbrace{a_{-1}b^{-1} + a_{-2}b^{-2} + \cdots + a_{-m+1}b^{-m+1} + a_{-m}b^{-m} + \cdots}_{\text{κλασματικό μέρος}} \leftarrow \boxed{\varepsilon,\ 0 \le \varepsilon < 1}$$

όπου

- $a_k :=$ ψηφίο του αριθμού
- $k =$ τάξη του ψηφίου α_k
- $m =$ πλήθος ψηφίων του κλασματικού μέρους
- $n =$ πλήθος ψηφίων του ακέραιου μέρους
- $b =$ ακέραιος με $b > 1$ και $0 \le a_k < b$ (ή $0 \le a_k \le b-1$), ο b λέγεται βάση του αριθμητικού συστήματος.

Ένα αριθμητικό σύστημα με βάση b απαιτεί b, το πλήθος, διαφορετικά ψηφία για την αναπαράσταση αριθμών, τα οποία παίρνουν τιμές από 0 έως $b - 1$. Η συμβολοσειρά (ακολουθία ψηφίων)

$$a_{n-1}a_{n-2}\ldots a_1 a_0 . a_{-1}a_{-2}\ldots a_{-m}$$

παριστάνει διαφορετικούς αριθμούς ανάλογα με το σύστημα αρίθμησης που χρησιμοποιούμε (επιλογή του b).

Παραδείγματα:

- $29.578 = \underset{\substack{\uparrow \\ a_1}}{2}\cdot 10^1 + \underset{\substack{\uparrow \\ a_0}}{9}\cdot 10^0 + \underset{\substack{\uparrow \\ a_{-1}}}{5}\cdot 10^{-1} + \underset{\substack{\uparrow \\ a_{-2}}}{7}\cdot 10^{-2} + \underset{\substack{\uparrow \\ a_{-3}}}{8}\cdot 10^{-3}$
- $b = 10 \rightarrow 521_{(10)} = 5\cdot 10^2 + 2\cdot 10^1 + 1\cdot 10^0$
- $b = 8 \rightarrow 521_{(8)} = 5\cdot 8^2 + 2\cdot 8^1 + 1\cdot 8^0$
- $b = 2 \rightarrow 521_{(2)} = ?$
- $1101_{(2)} = 1\cdot 2^3 + 1\cdot 2^2 + 0\cdot 2^1 + 1\cdot 2^0 = 13_{(10)}$

Οι περισσότεροι αριθμοί <u>δεν</u> μπορούν να αναπαρασταθούν ως άθροισμα πεπερασμένου πλήθους όρων. Π.χ. $\sqrt{2} \approx 1.4... = 1 + 0.4...$.

Οι ακέραιοι μπορούν να αναπαρασταθούν ακριβώς στο οποιοδήποτε αριθμητικό σύστημα.

Αν ο $x \in \mathbb{N}$ έχει m ψηφία στο σύστημα αυτό μπορεί να πάρει τιμές από 0 έως $b^m - 1$, δηλ. b^m διαφορετικές τιμές (γιατί;).

Στην ψηφιακή αναπαράσταση ενός φυσικού αριθμού το πιο αριστερό ψηφίο ονομάζεται «περισσότερο σημαντικό ψηφίο» (Most Significant Bit – MSB, στη δυαδική αναπαράσταση) γιατί πολλαπλασιάζεται με τον μεγαλύτερο συντελεστή. Το πιο δεξιό ψηφίο ονομάζεται «λιγότερο σημαντικό ψηφίο» (Less Significant Bit – LSB στη δυαδική αναπαράσταση) γιατί πολλαπλασιάζεται με τον μικρότερο συντελεστή.

Έστω $x \in \mathbb{N}$. Είναι

$$x=\sum_{k=0}^{n-1}a_k b^k = a_0+\sum_{k=1}^{n-1}a_k b^k = a_0 + b\sum_{k=1}^{n-1}a_k b^{k-1} = a_0 + bq_0$$

[Θεώρημα Διαίρεσης:

Για κάθε ακέραιο a και θετικό ακέραιο n, υπάρχουν μοναδικοί ακέραιοι q και r τέτοιοι, ώστε

$$a = q\cdot n + r \quad (1) \text{ και } \quad 0 \le r < n \quad (2).$$

Από τις (1) και (2) $\Rightarrow \frac{a}{n} = q + \frac{r}{n}$, $0 \le \frac{a}{n} < 1$

- $q = \left\lfloor \frac{a}{n} \right\rfloor$ (:= a div n)
- r := a mod n
- $n \mid a \Leftrightarrow a$ mod $n = 0$ **]**

Άρα a_0 = υπόλοιπο της διαίρεσης του x με το b.
Παρόμοια,

$$q_0 = \sum_{k=1}^{n-1}a_k b^k = a_1 + \sum_{k=2}^{n-1}a_k b^{k-1} = a_1 + b\sum_{k=2}^{n-1}a_k b^{k-2} = a_1 + bq_1$$

Συνεπώς,

a_1 = υπόλοιπο της διαίρεσης του q_0 με το b
a_2 = υπόλοιπο της διαίρεσης του q_1 με το b
…

Από τα παραπάνω είναι φανερό πλέον πώς μετατρέπουμε έναν ακέραιο από το δεκαδικό σύστημα στο σύστημα με βάση το b.

Αλγόριθμος Μετατροπής ως προς βάση *b*

```
// Είσοδος: n θετικός ακέραιος, b ακέραιος >1
  q := n
  k := 0
  while q ≠ 0
  begin
    a_k := q mod b
    q := ⌊q/b⌋
    k := k + 1
  end {αναπαράσταση του n ως προς b: (a_{k-1} ... a_1 a_0)_b}
```

Πλήθος ψηφίων

- Αν $x = \sum_{k=0}^{n-1} a_k b^k$ με $a_{n-1} \ne 0$ τότε το πλήθος των ψηφίων του x (ως προς βάση b) είναι $\text{len}(x) := n$.
- Αν $b, k \in \mathbb{Z}$ με $b > 1$ και $x \in \mathbb{R}$ τέτοιο ώστε $b^k \le x < b^{k+1}$ τότε $k = \lfloor \log_b x \rfloor$.

  Πράγματι, έχουμε

$$b^k \le x < b^{k+1} \Rightarrow k\cdot\log_b b^k \le \log_b x < \log_b b^{k+1} \Rightarrow$$
$$\Rightarrow k\cdot\log_b b \le \log_b x < (k+1)\log_b b \Rightarrow k \le \log_b x < k+1 \Rightarrow \lfloor \log_b x \rfloor = k$$

> ➢ Η αναπαράσταση ενός θετικού ακεραίου A στο αριθμητικό σύστημα με βάση $b > 1$ αποτελείται από $\text{len}(A) := \lfloor \log_b A \rfloor + 1$ ψηφία.

[***Απόδειξη:***
Έστω n το ζητούμενο πλήθος ψηφίων. Είναι

$$A = \sum_{k=0}^{n-1} a_k b^k = a_{n-1}b^{n-1} + a_{n-2}b^{n-2} + \ldots + a_1 b^1 + a_0 b^0$$

με $a_{n-1} \neq 0$, $b > 1$, και $0 \le a_k < b$ ή $0 \le a_k \le b-1$, $k = n-2,\ldots,1,0$
οπότε,

$$b^{n-1} \le a_{n-1}b^{n-1} \le n \ \ (\text{γιατί} \ \ a_{n-1} \ge 1) \Rightarrow b^{n-1} \le n \qquad (1)$$

Επίσης,

$$n = a_{n-1}b^{n-1} + \ldots + a_1 b + a_0 \le (b-1)b^{n-1} + \ldots + (b-1)b + (b-1) =$$
$$= (b-1)\left(b^{n-1} + \ldots + b + 1\right) = b^n - 1 \qquad (2)$$

Από τις (1) και (2) $\Rightarrow b^{n-1} \le A < b^n \Rightarrow b^k \le A < b^{k+1}$, όπου $k = n-1$
$\Rightarrow k = \lfloor \log_b A \rfloor \Rightarrow n-1 = \lfloor \log_b A \rfloor \Rightarrow n = \lfloor \log_b A \rfloor + 1$.]

<u>Δυαδική αναπαράσταση</u>

Όταν η βάση είναι $b = 2$ έχουμε τα δυαδικά ψηφία (**b**inary dig**it** – **bit**) $a_0 = 0$ $a_1 = 1$ και αριθμητική που ορίζεται από τις πράξεις

×	**0**	**1**
0	0	0
1	0	1

+	**0**	**1**
0	0	1
1	1	10

(είναι 1 + 1 = 10 που είναι το $2 = 1\cdot 2^1 + 0\cdot 2^0$ στο δυαδικό σύστημα)

Η ψηφίο προς ψηφίο πρόσθεση και ο πολλαπλασιασμός μπορούν να γίνουν ως προς βάση 2 με τον ίδιο τρόπο που γίνεται ως προς βάση 10.

Υπάρχουν κι άλλες ομοιότητες με τη βάση 10.

- Η ίδια η βάση αναπαρίσταται ως «10» και στις δύο περιπτώσεις· Ο πολλαπλασιασμός με τη βάση απλά μετατοπίζει την υποδιαστολή μια θέση δεξιά και η διαίρεση με τη βάση μετατοπίζει την υποδιαστολή μια θέση αριστερά.
- Αν k είναι ένας θετικός ακέραιος, η βάση υψωμένη εις την k-οστή δύναμη είναι ένας άσσος ακολουθούμενος από k μηδενικά
  - $10^3{}_{\{10\}} = 1000_{\{10\}}$
  - $10^3{}_{\{2\}} = 2^3{}_{\{10\}} = (1\cdot 2^3 + 0\cdot 2^2 + 0\cdot 2^1 + 0\cdot 2^0)_{\{10\}} = 1000_{\{2\}}$
- Αν k είναι ένας θετικός ακέραιος, η βάση υψωμένη εις την $(-k)$-οστή δύναμη είναι η υποδιαστολή ακολουθούμενη από $(k-1)$ μηδενικά και ένα άσσο
  - $10^{-4} = .0001$

Παράδειγμα

Ως ένα χαρακτηριστικό παράδειγμα αριθμητικής στο δυαδικό σύστημα ας δούμε (πάλι) τον Πολλαπλασιασμό του Ρώσου Χωρικού (ΠΡΧ) με $M = 73$ και $N = 41$.

A	**B**
73	41
~~146~~	20
~~292~~	10
584	5
~~1168~~	2
2336	1
2993	

Κάθε μια τιμή-B προέκυψε «διαιρώντας δια 2 και ελαττώνοντας το αποτέλεσμα κατά μισή μονάδα, αν απαιτείται, προκειμένου να προκύψει ακέραιος». Όταν οι τιμές-B γράφονται στη δυαδική μορφή, διαιρούμε με τη βάση και κάνουμε αποκοπή σε ακέραιο. Αυτό έχει ως αποτέλεσμα την απαλοιφή του τελευταίου bit της δυαδικής αναπαράστασης του B

$B\{10\}$	$B\{2\}$	
41	101001	// $41 = 32 + 8 + 1 = 2^5 + 2^3 + 2^0$
20	10100	// $20 = 16 + 4 \quad = 2^4 + 2^2$
10	1010	// $10 = 8 + 2 \quad = 2^3 + 2^1$
5	101	// $5 = 4 + 1 \quad = 2^2 + 2^0$
2	10	
1	1	

// <u>Τώρα</u>, αναλογίζεστε ότι η τιμή-B <u>πρέπει</u> τελικά να γίνει ίση με 1 ακριβώς?
// Πλήθος των γραμμών του πίνακα = πλήθος των bit του $N_{\{2\}}$.
// Η τιμή-B είναι άρτιος αριθμός όταν το $B_{\{2\}}$ τελειώνει σε μηδέν και είναι περιττός // αριθμός όταν το $B_{\{2\}}$ τελειώνει σε άσσο.

// Και οι τιμές-A?

Ο πολλαπλασιασμός επί 2 προσθέτει ένα μηδενικό στη δυαδική αναπαράσταση του A.

$A\{10\}$	$A\{2\}$	
73	100 100 1	// $73 = 64 + 8 + 1 = 2^6 + 2^3 + 2^0$
146	100 100 10	
292	100 100 100	
584	100 100 100 0	
1168	100 100 100 00	
2336	100 100 100 000	

Αν κάνουμε τον πολλαπλασιασμό με τον γνωστό μας τρόπο, θα έχουμε

73{10}=	1001001		
×41{10}=	×101001		
	1001001	73	// πολ/σιασμός με το ψηφίο 1
	~~1001001x~~	~~146~~	// μετατόπιση & πολ/σιασμός με το 0
	~~1001001xx~~	~~292~~	// μετατόπιση & πολ/σιασμός με το 0
	1001001xxx	584	// μετατόπιση & πολ/σιασμός με το 1
	~~1001001xxxx~~	~~1168~~	// μετατόπιση & πολ/σιασμός με το 0
	1001001xxxxx	2336	// μετατόπιση & πολ/σιασμός με το 1
	101110110001	2993	

$$
\begin{aligned}
// \ 101\ 110\ 110\ 001\{2\} &= 2^{11} + 2^{9} + 2^{8} + 2^{7} + 2^{5} + 2^{4} + 2^{0}\{10\} \\
// \qquad &= 2048 + 512 + 256 + 128 + 32 + 16 + 1\{10\} \\
// \qquad &= 2993\{10\}
\end{aligned}
$$

Ο ΠΡΧ εμφανίζεται να είναι απλά μια μεταμφιεσμένη εκδοχή του πολλαπλασιασμού ως προς βάση 2.

Παραθέτουμε την αντιστοιχία μεταξύ του δεκαδικού, δυαδικού, οκταδικού και δεκαεξαδικού συστήματος (για μερικούς από τους αρχικούς ακεραίους):

Δεκαδικό	Δυαδικό	Οκταδικό	Δεκαεξαδικό
0	0	0	0
1	1	1	1
2	10 =b	2	2
3	11	3	3
4	100 $=b^2$	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000 $=b^3$	10 =b	8
9	1001	11	9
10 =b	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000 $=b^4$	20	10 =b
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14

Ψηφιακή αποθήκευση μη ακεραίων αριθμών

Χρησιμοποιούμε ‘‘επιστημονικό συμβολισμό’’
$123.456 \rightarrow 0.123456\times10^3$

Αριθμοί κινητής υποδιαστολής

$\pm (0.d_1d_2\ldots d_n)10^s$	s = εκθέτης
$\pm (0.b_1b_2\ldots b_n)2^t$	$(0.d_1d_2\ldots d_n)$ = σημαινόμενο μέρος ή mantissa
$= \pm p\times2^t$	p = δυαδικό ισοδύναμο του σημαινόμενου μέρους
	t = δυαδικό ισοδύναμο του εκθέτη)

- ♦ καθορισμένος (fixed) αριθμός bit κατανέμεται σε κάθε αριθμό
- ➢ απλή ακρίβεια (single precision) χρησιμοποιεί 32 bit ανά αριθμό κινητής υποδιαστολής
- ➢ διπλή ακρίβεια (double precision) χρησιμοποιεί 64 bit ανά αριθμό κινητής υποδιαστολής

Σε κάθε μια απ’ αυτές τις περιπτώσεις έχουμε σταθερό αριθμό bit κατανεμημένων για την αποθήκευση του σημαινόμενου μέρους και σταθερό αριθμό bit για την αποθήκευση του εκθέτη

	b	$bb\ldots b$	$bb\ldots b$
	↓	↓	↓
	πρόσημο	σημαινόμενο	εκθέτης
απλή ακρίβεια	1	23	8
διπλή ακρίβεια	1	52	11

<u>Αλγόριθμος</u> μετατροπής από δεκαδικό στο δυαδικό σύστημα:

Έστω $r \in \mathbb{R}$, $0 < r < 1$. Μπορούμε να γράψουμε,

$r = d_1\times2^{-1} + d_2\times2^{-2} + \ldots + d_n\times2^{-n} + \ldots$ // $d_i = 0$ ή 1

$$= \sum_{k=1}^{\infty} d_k\, 2^{-k}$$

$\Rightarrow$

$$2r = d_1 + (d_2\times2^{-1} + d_2\times2^{-2} + \ldots + d_n2^{-n+1} + \ldots)$$

$\downarrow$ $\lfloor 2r \rfloor$ — το ακέραιο μέρος του $2r$

$\downarrow$ $]2r[$ — το κλασματικό μέρος του $2r$

$\Rightarrow 2r = d_1 + f_1$, με $d_1 = \lfloor 2r \rfloor$, $f_1 =]2r[$

$f_1 = d_2\times2^{-1} + \ldots + d_n2^{-n+1} + \ldots$

$\Rightarrow 2f_1 = d_2 + (d_3\times2^{-1} + \ldots + d_n\times2^{-n+2} + \ldots)$

$d_2 = \lfloor 2f_1 \rfloor$, $f_2 =]2f_1[= d_3\times2^{-1} + \ldots + d_n\times2^{-n+2} + \ldots$

$\Rightarrow$ Δημιουργούνται δύο αναδρομικές ακολουθίες

$d_n = [2f_{n-1}]$ με $d_1 = [2r]$

$f_n =]2f_{n-1}[$ με $f_1 =]2r[$

Για παράδειγμα,

$$(0.2)_{\{10\}} = (?)_{\{2\}}$$

$r = 0.2$

$2r = 0.4$	$d_1 = 0$	$f_1 = 0.4$
$2f_1 = 0.8$	$d_2 = 0$	$f_2 = 0.8$
$2f_2 = 1.6$	$d_3 = 1$	$f_3 = 0.6$
$2f_3 = 1.2$	$d_4 = 1$	$f_4 = 0.2$
$2f_4 = 0.4$	$d_5 = 0$	$f_5 = 0.4$

$0.2 \rightarrow 0011 \quad 0011 \quad 0011 \ldots$

$(0.2)_{\{10\}} = 0.\overline{0011}$

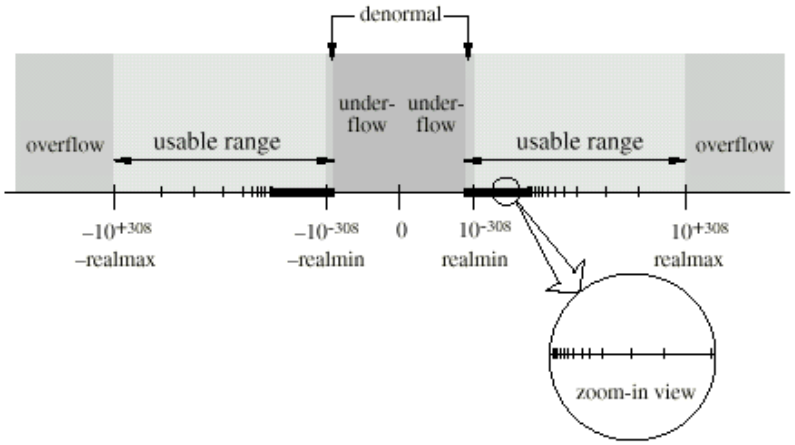
Αντίστροφα,

$$0.\overline{01} = 0\times2^{-1} + 1\times2^{-2} + 0\times2^{-3} + 1\times2^{-4} + \ldots =$$

$$= \sum_{k=1}^{\infty}\left(2^{-2}\right)^k = -1 + \sum_{k=0}^{\infty}\left(2^{-2}\right)^k = -1 + \frac{1}{1-\frac{1}{4}} = -1 + \frac{4}{3} = \frac{1}{3}$$

Τώρα μπορούμε να πούμε ότι:

- Περιορισμός του αριθμού των *bit* που κατανέμονται για αποθήκευση του εκθέτη ⇒ υπάρχει ένα άνω και ένα κάτω φράγμα για το μέγεθος του αριθμού κινητής υποδιαστολής που μπορούν να χρησιμοποιηθούν. Έτσι σε αντίθεση με την ευθεία των πραγματικών αριθμών, στην οποία έχουμε κατά συνεχή τρόπο αναπαράσταση των αριθμών, η ευθεία των αριθμών κινητής υποδιαστολής έχει την παρακάτω εικόνα



- Διπλής ακρίβειας εκθέτες χρησιμοποιούν 11 bit (1 για πρόσημο και 10 για το μέγεθος του εκθέτη)

  Το μεγαλύτερο που μπορεί να είναι : $2^{10}-1=1023$
  ⇒ η μεγαλύτερη δύναμη του 2 : $2^{1023} \cong 9\times10^{307} \cong 10^{308}$

- Περιορισμός του αριθμού των *bits* που δεσμεύονται για την αποθήκευση του σημαινόμενου μέρους ⇒ υπάρχει όριο στην <u>ακρίβεια</u> για έναν αριθμό κινητής υποδιαστολής ⇒ στρογγυλοποίηση⇒ $(0.2)_{10} = (\overline{(0011)}_2)$
- Οι περισσότεροι πραγματικοί αριθμοί δεν μπορούν να αποθηκευτούν ακριβώς (δεν υπάρχουν στην ευθεία των αριθμών κινητής υποδιαστολής)

- ακέραιοι < 2^{52} μπορούν να αναπαρασταθούν ακριβώς.

Στο *MATLAB* έχουμε,

```
» x=2^51
x =
    2.251799813685248e+015
» s=dec2bin(x)
s =
1000000000000000000000000000000000000000000000000000
» x2=bin2dec(s)
x2 =
    2.251799813685248e+015
» x2-x
ans =
     0

» x=2^52
x =
    4.503599627370496e+015
» s=dec2bin(x)
s =
10000000000000000000000000000000000000000000000000000
» x2=bin2dec(s)
??? Error using ==> bin2dec
Binary string must be 52 bits or less.
```

- Αριθμοί με 15 (δεκαδικά) ψηφία σημαινόμενο μέρος (mantissa) οι οποίοι είναι το ακριβές άθροισμα δυνάμεων του 1/2 μπορούν να αποθηκευτούν ακριβώς.