Logică computațională Curs 1

Lector dr. Pop Andreea-Diana

Baze de numerație

Scrierea pozițională a unui număr

$$a_n a_{n-1} \dots a_1 a_0 (p) = a_n * p^n + a_{n-1} * p^{n-1} + \dots + a_1 * p^1 + a_0 * p^0$$

p – baza numărului, $\forall i=0,n, 0 \le a_i \le p$

Compararea a două numere

$$a_n a_{n-1} \dots a_1 a_{0(p)}$$
 și $b_m b_{m-1} \dots b_1 b_{0(p)}$

$$a_i?b_i, i=\overline{n,0}$$

Adunarea a două numere

$$a_n a_{n-1} \dots a_1 a_{0(p)} + b_m b_{m-1} \dots b_1 b_{0(p)} = c_k c_{k-1} \dots c_1 c_{0(p)}$$

$$t_0=0$$

 $i=\overline{0,k}$, $k=max(n,m)+1$

$$(a_i + b_i + t_i) : p = t_{i+1} \text{ rest } c_i$$

Scăderea a două numere

Precondiție: $A \ge B$

$$a_n a_{n-1} \dots a_1 a_{0 (p)} - b_m b_{m-1} \dots b_1 b_{0 (p)} = c_n c_{n-1} \dots c_1 c_{0 (p)}$$

$$t_0 = 0$$

$$i = \overline{0,n} , c_i = \begin{cases} a_i + t_i - b_i, \text{ dacă } a_i + t_i \ge b_i; t_{i+1} = 0 \\ p + a_i + t_i - b_i, \text{ altfel}; t_{i+1} = -1 \end{cases}$$

Înmulțirea unui număr cu o cifră

$$a_n a_{n-1} \dots a_1 a_{0(p)} * b_{(p)} = c_k c_{k-1} \dots c_1 c_{0(p)}$$

$$t_0=0$$

 $i=\overline{0,k}$, $k=n+1$

$$(a_i *b + t_i) : p = t_{i+1} \text{ rest } c_i$$

 $c_{n+1} = t_{n+1}$

Împărțirea unui număr cu o cifră

$$a_n a_{n-1} \dots a_1 a_{0(p)} : b_{(p)} = c_n c_{n-1} \dots c_1 c_{0(p)} \text{ rest } r_{(p)}$$

$$t_n = 0$$
 $i = \overline{n,0}$

$$(t_i * p + a_i) : b = c_i \text{ rest } t_{i-1}$$

$$r = t_{-1}$$

Conversia nr. întregi prin împărțiri succesive

- calculele se efectuează în baza sursă (10)
- nr. se împarte cu baza destinație => cât și rest
- câtul se împarte cu baza destinație => cât și rest
- . . .
- până când câtul =0
- se iau resturile în ordinea inversă => nr. în baza destinație

Conversia nr. subunitare prin înmulțiri succesive

- calculele se efectuează în baza sursă (10)
- nr. se înmulțește cu baza destinație => parte întreagă și o parte fracționară
- partea fracționară se înmulțește cu baza destinație => parte întreagă și o parte fracționară
- ...
- până când partea fracționară =0 v se repetă partea fracționară v s-au obținut suficiente cifre
- se iau părțile întregi în ordinea obținerii => nr. în baza destinație

Conversia prin substituție

• calculele se efectuează în baza destinație (10)

$$a_n a_{n-1} \dots a_1 a_0, a_{-1} a_{-2} \dots a_{-m (p)} = (a_n * p^n + a_{n-1} * p^{n-1} + \dots + a_1 * p^1 + a_0 * p^0 + a_{-1} * p^{-1} + a_{-2} * p^{-2} + \dots + a_{-m} * p^{-m})_{(10)}$$

Conversii rapide

între bazele 2, 4, 8, 16

- \bullet 2->4=2² / 8=2³ / 16=2⁴
 - se pornește de la virgulă spre stânga și spre dreapta
 - se grupează câte k $(2^k = baza destinație)$
 - dacă e cazul se adaugă 0 la începutul/sfârșitul nr.
 - e convertește fiecare grup la cifra corespunzătoare din baza destinație (pe baza tabelului)
- \bullet 4=2² / 8=2³ / 16=2⁴->2
 - se înlocuiește fiecare cifră cu grupul de k cifre corespunzător din tabel

Tabel conversii rapide 2->4=2²

2	4
00	0
01	1
10	2
11	3

Tabel conversii rapide 2->8=2³

2	8
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Tabel conversii rapide 2->16=24

2	16
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	В
1100	C
1101	D
1110	E
1111	F

Conversia utilizând o bază intermediară

$$a_n a_{n-1} \dots a_1 a_0, a_{-1} a_{-2} \dots a_{-m (p)} = ?_{(q)}$$

Se poate utiliza baza intermediară 10:

Prin substituție

$$a_n a_{n-1} \dots a_1 a_0, a_{-1} a_{-2} \dots a_{-m (p)} = b_x b_{x-1} \dots b_1 b_0, b_{-1} b_{-2} \dots b_{-y(10)}$$

Prin împărțiri succesive

$$b_x b_{x-1} \dots b_1 b_0 (10) = c_1 c_{1-1} \dots c_1 c_0 (q)$$

Prin înmulțiri succesive

$$0, b_{-1} b_{-2} \dots b_{-y(10)} = 0, c_{-1} c_{-2} \dots c_{-k(q)}$$

$$a_n a_{n-1} \dots a_1 a_0, a_{-1} a_{-2} \dots a_{-m (p)} = c_1 c_{1-1} \dots c_1 c_0, c_{-1} c_{-2} \dots c_{-k (q)}$$

Conversia din baza 2 utilizând o bază intermediară

$$a_n a_{n-1} \dots a_1 a_0, a_{-1} a_{-2} \dots a_{-m} = ?_{(10)}$$

Se poate utiliza baza intermediară putere a lui 2 (p):

Prin conversii rapide

$$a_n a_{n-1} \dots a_1 a_0, a_{-1} a_{-2} \dots a_{-m} = b_x b_{x-1} \dots b_1 b_0, b_{-1} b_{-2} \dots b_{-y(p)}$$

Prin substituție

$$b_x b_{x-1} \dots b_1 b_0, b_{-1} b_{-2} \dots b_{-y(p)} = c_l c_{l-1} \dots c_1 c_0, c_{-1} c_{-2} \dots c_{-k} c_{(10)}$$

$$a_n a_{n-1} \dots a_1 a_0, a_{-1} a_{-2} \dots a_{-m} (2) = c_1 c_{l-1} \dots c_1 c_0, c_{-1} c_{-2} \dots c_{-k} (10)$$

Conversia în baza 2 utilizând o bază intermediară

$$a_n a_{n-1} \dots a_1 a_0, a_{-1} a_{-2} \dots a_{-m (10)} = ?_{(2)}$$

Se poate utiliza baza intermediară putere a lui 2 - (p):

Prin împărțiri succesive

$$a_n a_{n-1} \dots a_1 a_{0(10)} = b_x b_{x-1} \dots b_1 b_{0(p)}$$

Prin înmulțiri succesive

$$0, a_{-1} a_{-2} \dots a_{-m (10)} = 0, b_{-1} b_{-2} \dots b_{-y (p)}$$

Prin conversii rapide

$$b_x b_{x-1} \dots b_1 b_0, b_{-1} b_{-2} \dots b_{-y(p)} = c_1 c_{l-1} \dots c_1 c_0, c_{-1} c_{-2} \dots c_{-k(2)}$$

$$a_n a_{n-1} \dots a_1 a_0, a_{-1} a_{-2} \dots a_{-m} (10) = c_1 c_{l-1} \dots c_1 c_0, c_{-1} c_{-2} \dots c_{-k} (2)$$



- un prieten că dacă se-ntâm plă asa, trebuie să apăs pe About Internet Security într-un meniu (da' care, nu-mi mai amintesc) și să chem pe cineva să-mi traducă.
- Mai e neste faze acilea tot cu saiturile securizate, da' dacă nu esti hecăr nici nu te gândi să-ncerci... Nu că n-ar merge, da' de ce atâta chin câtă vreme... (ie'te... ce chestie... mi'am pierdut ideea... în orice caz, era vorba despre SSL 2.0, SSL 3.0, TLS 1.0, PCT 1.0 si alte cretinităti din astea. Oricum, trăiască si-nflorească scumpa noastră patrie, partidul de guvernământ, prietena mea care mi-a fost mereu aproape în momente din astea și cine mai vreti voi. Hai baftă.
- Na că era să uit... stie careva la ce foloseste asta? ☐ Back.

Cannot find server or DNS Error Internet Explorer