Systemy liczbowe

Dla każdej liczby naturalnej $x \in$ N oraz liczby naturalnej $p >= 2$ istnieją jednoznacznie wyznaczone:
liczba $n \in$ N oraz ciąg cyfr $c_0, c_1, \ldots, c_{n-1}$ (gdzie $c_k \in \{0, 1, \ldots, p - 1\}$) taki, że
$x = c_0 + c_1 \cdot p + c_2 \cdot p^2 + \ldots + c_{n-1} \cdot p^{n-1}$
Ciąg $(c_{n-1} \ldots c_0)_p$ nazywamy reprezentacją liczby x w systemie o podstawie p.

Zamiana liczby z systemu o podstawie 10 na system dwójkowy. Algorytm Euklidesa

Przy zamianie liczby korzystamy z dzielenia całkowitego oznaczanego operatorem „/" i reszty z dzielenia całkowitego oznaczanego operatorem „%" (**modulo)** dla przykładu:
`7 / 3 = 2,` bo w `7` mieszczą się dwie trójki
`7 % 3 = 1`, bo reszta z dzielenia `7` przez `3 = 1`;

- *Samodzielnie wykonaj następujące operacje:*

```
13 / 2 =        30 / 5 =        73 / 15 =       15 / 23 =
13 % 2 =        30 % 5 =        73 % 15 =       15 % 23 =
```

Przykład Zamień liczbę 41 na system dwójkowy:
liczbę 41 dzielimy modulo 2 i wynik zapisujemy z prawej strony, następnie liczbę 41 dzielimy całkowicie przez 2 i wynik zapisujemy pod spodem aż do momentu otrzymania 0;

```
41    |     1     , bo 41 % 2 = 1
20    |     0     , bo 20 % 2 = 0
10    |     0
5     |     1
2     |     0
1     |     1
0
```

Aby mieć prawidłową postać liczby `41` w systemie dwójkowym (binarnym) należy prawą kolumnę przepisać w kolejności od dołu do góry: `101001`

- *Samodzielnie dokonaj zamiany na system dwójkowy następujących liczb:*

```
55,  79,  64, 128,  32,  33,  31,  1, 256, 333
```

Zamiana liczby dziesiętnej na dowolny system.

Analogicznie możemy dokonać zamiany z systemu dziesiętnego na system o dowolnej podstawie wykonując dzielenie nie przez 2 lecz przez liczbę, która jest podstawą systemu. W zapisie liczb w systemach o podstawie >10 przyjmujemy następujące oznaczenia dla cyfr: `10 = A, 11 = B, 12 = C` , itd.
Przedstawmy liczbę `63` w systemie o podstawie `5`:

```
63          3     , bo 63 % 5 = 3
12          2     , bo 12 % 5 = 2,   (12 to wynik z dzielenia całkowitego 63 / 5)
2           2
0
```

Zatem $63_{(10)} = 223_{(5)}$

- *Analogicznie dokonaj zamiany:* $73_{(10)} = \ldots\ldots\ldots_{(5)}$; $87_{(10)} = \ldots\ldots\ldots_{(6)}$;
  $125_{(10)} = \ldots\ldots\ldots_{(13)}$ $125_{(10)} = \ldots\ldots\ldots_{(16)}$

Obliczanie wartości dziesiętnej liczb zapisanych w innych systemach:

Aby odczytać wartość dziesiętną liczby zapisanej w systemie o danej podstawie postępujemy w następujący sposób: $1\ 0\ 1\ 0\ 0\ 1_{(2)} = 1 * 2^0 + 0 * 2^1 + 0 * 2^2 + 1 * 2^3 + 0 * 2^4 + 1 * 2^5 = 1 + 1*8 + 1 * 32 = 41$;

- *Oblicz wartości liczb zgodnie z podstawą podaną w nawiasie:*

$10111011_{(2)} =\ldots_{(10)}$ $11111111_{(2)} =\ldots_{(10)}$ $10000000_{(2)} =\ldots_{(10)}$

$11011_{(2)} =\ldots_{(10)}$ $10000_{(2)} =\ldots_{(10)}$ $312_{(4)} =\ldots_{(10)}$

$25_{(16)} =\ldots_{(10)}$ $1D_{(16)} =\ldots_{(10)}$ $1D_{(12)} =\ldots_{(10)}$

$314_{(16)} =\ldots_{(10)}$ $11111111_{(2)} =\ldots_{(4)}$ $10000000_{(2)} =\ldots_{(5)}$

$11011_{(2)} =\ldots_{(6)}$ $10000_{(2)} =\ldots_{(16)}$

Obliczanie wartości dziesiętnej liczb przy pomocy schematu Hornera

$c_{n-1} \cdot p^{n-1} + c_{n-2}\, p^{n-2} + \ldots + c_2 \cdot p^2 + c_1 \cdot p + c_0 = (\ldots((c_{n-1} p + c_{n-2})p + c_{n-3})\ldots + c_1)p + c_0$

Schemat Hornera

$b_{n-1} = c_{n-1}$

$b_k = b_{k+1} \cdot p + c_k, \quad (k = n-2, \ldots, 0)$

Przykład $x = (11001111)_2$

c_k	1	1	0	0	1	1	1	1
b_k	1	3	6	12	25	51	103	207

Stąd $(11001111)_2 = (207)_{10}$

- Oblicz wartości dziesiętne liczb korzystając ze schematu Hornera

$(10111011110011)_2 =(\ldots\ldots\ldots)_{10}$

$(11001010101010)_2 =(\ldots\ldots\ldots)_{10}$

$(1234)_{16} =(\ldots\ldots\ldots)_{10}$

$(ABCD)_{16} =(\ldots\ldots\ldots)_{10}$

Zamiana systemu dwójkowego na szesnastkowy

W łatwy sposób dokonujemy przejścia z systemu dwójkowego na 16-tkowy: grupujemy od prawej po cztery bity rozwinięcia binarnego liczby i następnie odczytując wartość dziesiętną takiej grupy bitów podstawiamy odpowiednią cyfrę z systemu szesnastkowego:

$$101111010110_{(2)} = \underbrace{1011}_{=11=B}\ \underbrace{1101}_{=13=D}\ \underbrace{0110}_{=6} = \mathrm{BD6}_{(16)}$$

Zamiana systemu szesnastkowego na dwójkowy

Odwrotnie postępujemy przy przejściu z systemu szesnastkowego na binarny:

$$\mathrm{BD6}_{(16)} = \underbrace{1011}_{=11=B}\ \underbrace{1101}_{=13=D}\ \underbrace{0110}_{=6} = 101111010110_{(2)}$$

pamiętając o konieczności zapisania każdej cyfry szesnastkowej na 4 bitach, czyli na uzupełnieniu z przodu o ile zachodzi taka potrzeba odpowiednią ilością zer (tak jak to miało miejsce w powyższym przykładzie dla cyfry 6 która zapisaliśmy jako 0110)

- *Samodzielnie dokonaj zamiany liczb:*

$10011001110_{(2)} =\ldots_{(16)}$ $1001010010110_{(2)} =\ldots_{(16)}$

$100101110_{(2)} =\ldots_{(16)}$ $100110011010_{(2)} =\ldots_{(16)}$

$10110010110_{(2)} =\ldots_{(16)}$ $1000101100010110_{(2)} =\ldots_{(16)}$

$1A1_{(16)} =\ldots\ldots\ldots_{(2)}$ $3B7_{(16)} =\ldots\ldots\ldots_{(2)}$

$596_{(16)} =\ldots\ldots\ldots_{(2)}$ $FF_{(16)} =\ldots\ldots\ldots_{(2)}$

$2D2_{(16)} =\ldots\ldots\ldots_{(2)}$ $22_{(16)} =\ldots\ldots\ldots_{(2)}$

dodawanie binarne:

```
  1101
+  101
 10010
```

1+1= 2 czyli w systemie binarnym jest to 0 i 1 na pozycji o jeden wyższej jest to analogia przy dodawaniu dziesiętnym np. 7+6=13 czyli 3 i 1 na pozycji o jeden wyższej (tzw. jeden dalej)

- *Samodzielnie dokonaj dodawania:*

1111	*10001*	*1111*	*1101*	$322_{(4)}$ $1212_{(3)}$	
+ *101*	*+1111*	*+111*	*101*	$+212_{(4)}$	$+212_{(3)}$
			+ *10010*		

- *Sprawdź wyniki odczytując wartości dziesiętne składników i wartości sumy.*

- *Dla chętnych:*

1111	*10001*	*1111*	*1101*	$322_{(4)}$ $1212_{(3)}$	
- 101	*- 1111*	*- 111*	** 101*	$- 212_{(4)}$	$- 212_{(3)}$

Reprezentacja uzupełnieniowa.

Reprezentacja uzupełnieniowa służy do zapisu liczb całkowitych (dodatnich i ujemnych). Pierwszy bit od lewej jest bitem znaku (`0` – liczba dodatnia, `1` – liczba ujemna), pozostałych `n-1` bitów, to liczba zapisana w postaci binarnej, przy czym dla liczb ujemnych przechowuje się reprezentację binarną liczby $x_{uz}=2^n-|x|$

$$\text{Reprezentacja}(x) = \begin{cases} \text{Reprezentacja}(x) \text{ dla } 0 <= x <= 2^{n-1} - 1 \\ \text{Reprezentacja}(2^n - |x|) \text{ dla } -2^{n-1} <= x < 0 \end{cases}$$

W ten sposób po prawej stronie definicji mamy reprezentacje liczb nieujemnych, które umiemy już liczyć.

Metoda negacji bitów

Liczby ujemne możemy reprezentować binarnie tylko przy ustalonej z góry ilości bitów. Najprostszą metodą zamiany jest metoda "negacji bitów"

1) wyznaczyć `n`-bitową reprezentację binarną liczby `|x|`
2) w uzyskanej reprezentacji zamienić `0` na `1` i `1` na `0`
3) do wyniku dodać `1`

Dla przykładu przedstawmy reprezentacje liczby -35 na 8 bitach.

- Najpierw tworzymy w znany już sposób reprezentacje binarną liczby `|-35|=35 = 1000011` teraz uzupełniamy zerami od lewej tą reprezentację do 8-u bitów `01000011`
- następnie dokonujemy negacji bitów czyli $0 \leftrightarrow 1;\quad 1 \leftrightarrow 0$ `10111100`
- do takiej reprezentacji dodajemy 1 w wyniku czego otrzymujemy reprezentacje uzupełnieniową liczby -35 na 8 bitach `10111101`.

- *Przedstaw w reprezentacji uzupełnieniowej na 8 bitach: -33, -3, -17, -1, 7,-55, -87*
- *Przedstaw w reprezentacji uzupełnieniowej na 16 bitach: -33, -3, -7, -1, 7,-15, -87*

Odczytywanie wartości liczb zapisanych w notacji uzupełnieniowej:

Wartość liczby o reprezentacji uzupełnieniowej $(c_{n-1}, c_{n-2}, \ldots, c_1, c_0)$ jest równa
$x = -c_{n-1} \cdot 2^{n-1} + (c_{n-2} \cdot 2^{n-2} + \ldots + c_1 \cdot 2^1 + c_0)$
(najstarszy bit n-bitowej reprezentacji traktujemy jako -2^{n-1}, a pozostałe tradycyjnie jako wartości, kolejno $2^{n-2}, \ldots, 2^0$)

Przykład.
$101011 = -1 * 2^{6-1} + 1 + 1*2 + 0*4 + 1*8 + 0*16 = -32 + 11 = -21$ (liczba 6 bitowa)
Aby podać wartość dziesiętna liczby zapisanej w notacji uzupełnieniowej podnosimy 2 do potęgi o jeden niższej niż ilość bitów (6-1), tą wartość przemnażamy przez pierwszy bit od lewej ze znakiem minus. Z pozostałych bitów (w tym przypadku 01011) odczytujemy wartość dziesiętna tak jak dla liczb dodatnich i wartość tą dodajemy do wcześniej otrzymanej potęgi dwójki. Zauważmy, że jeżeli pierwszy bit od lewej jest równy 0, to mamy reprezentacje liczby dodatniej, w pozostałych przypadkach zawsze będzie to liczba ujemna.

- *Odczytaj wartości dziesiętne liczb zapisanych w notacji uzupełnieniowej:*
  1011110 11100101 1111 111111 0101101 100000001 100010001

Algorytm Hornera dla reprezentacji uzupełnieniowej (zamiana na wartość dziesiętną):
przed najstarszą cyfrą c_{n-1} stawiamy znak minus i postępujemy tak jak poprzednio w algorytmie Hornera:

Przykład. Zapisać w postaci dziesiętnej liczbę 10111110 zapisaną w reprezentacji uzupełnieniowej korzystając z algorytmu Hornera.

i	7	6	5	4	3	2	1	0
c_i	1	0	1	1	1	1	1	0
p = 2	-1	-2	-3	-5	-9	-17	-33	-66

Reprezentacja liczb rzeczywistych – stałopozycyjna:

Ułamek to liczba postaci $0.c_{-1}c_{-2}...c_{-k}$

Zamiana ułamka dziesiętnego na binarny polega na mnożeniu liczby przez 2 i wypisywaniu kolejnych cyfr przed kropką.

Przedstawmy reprezentacje liczby 0,3

0,3	0
0,6	1
0,2	0
0,4	0
0,8	1
0,6	1
0,2	0

ułamek mnożymy przez 2, jeżeli otrzymujemy liczbę mniejsza od 1, to po prawej stronie piszemy 0, a wynik przepisujemy pod spód, jeżeli wynik jest większy od 1 to po prawej stronie piszemy jeden, a część ułamkową z wyniku przepisujemy na dół, jeżeli wynik jest równy 1 to jedynkę przepisujemy po prawej stronie i kończymy procedurę przeliczania. Aby przedstawić reprezentację liczby rzeczywistej piszemy po 0. wszystkie bity od góry do dołu 0.3 = 0.0100110 Zauważmy, że czasami dostaniemy rozwinięcie skończone np. 0.25 = 0.01; nieskończone np. 0.137 lub okresowe np. 0.3 = 0.0(1001)

- *Zamień na system binarny liczby:*
  0,7; 0,125; 0,43; 0,55;

Odczytywanie wartości dziesiętnej ułamka w reprezentacji binarnej.
Aby określić wartość dziesiętna ułamka postępujemy jak niżej:
$0.\underline{0}\ \underline{1}\ \underline{1} = \underline{0}* 2^{-1}+\underline{1}*2^{-2}+\underline{1}*2^{-3} = 0,25 + 0,125$

- *Podaj wartości dziesiętne następujących ułamków:*
  0.1011; 0.0011; 0.11; 0.010101

Zaokrąglenie ułamków
Jeżeli chcemy zaokrąglić ułamek na k-tej pozycji, to patrzymy się na k+1 cyfrę i jeśli jest ona zerem, to po prostu cały ogon odrzucamy (zaokrąglenie w dół), a jeśli jest jedynką, to dodajemy ją do uciętego na k-tym miejscu przybliżenia (zaokrąglenie w górę)
np. Zaokrąglić ułamek 1/10 do czterech miejsc po przecinku
Wiemy, że 1/10=0.0(0011)
Zapisujemy ułamek z dokładnością do 5 cyfr po przecinku 0.00011 i dodajemy 1.

```
0.00011
+     1
-------
0.00100
```

Następnie obcinamy do 4-ch miejsc zatem 1/10 = 0.0010 z dokładnością d 4-ch miejsc po przecinku