Kodowanie informacji

Systemy liczbowe

prof. dr hab. inż. Joanna Józefowska

Poznań, rok akademicki 2008/2009

Plan wykładu

Definicja i klasyfikacja

System liczbowy

zbiór reguł jednolitego zapisywania cyfr. Wyróżniamy systemy:

- addytywne,
- pozycyjno-wagowe.

Systemy addytywne

- definiuje się symbole odpowiadające kilku małym liczbom i ich wielokrotnościom,
- liczbę odczytuje się dodając do siebie wartości podanych znaków - stąd nazwa,
- przykładem jest system rzymski:

I = 1 X = 10 C = 100 M = 1000
V = 5 L = 50 D = 500

Przykłady

M**CM**LXX**IX** $= 1000 + (1000 - 100) + 50 + 20 + (10 - 1) = 1979$
Jak zapisać 2008?
MMVIII $= 1000 + 1000 + 5 + 1 + 1 + 1 = 2008$

Systemy pozycyjno-wagowe

- znaczenie znaków zależy od ich pozycji,
- każdej pozycji przypisana jest inna waga,
- liczba całkowita postaci $a_{n-1}a_{n-2}\ldots a_1 a_0$ w systemie o podstawie (bazie) p ma wartość: $a_{n-1}p^{n-1} + a_{n-2}p^{n-2} + \ldots + a_1 p + a_0$
- przykładem jest system dziesiętny.

System dziesiętny (decymalny, arabski)

- Tradycyjny system matematyki.
- Podstawą systemu jest liczba 10.
- Liczbę zapisujemy za pomocą cyfr 0, 1, 2, 3, 4, 5, 6, 7, 8 i 9 o przypisanych wartościach.

Przykład

$(6352)_{10} = 6x10^3 + 3x10^2 + 5x10^1 + 2x10^0$

System dwójkowy (binarny)

- Naturalny system informatyki.
- Podstawą systemu jest liczba 2.
- Liczbę zapisujemy za pomocą cyfr 0 i 1 o przypisanych wartościach.

Przykład

$(1011)_2 = 1x2^3 + 0x2^2 + 1x2^1 + 1x2^0 = 11_{10}$

- zalety
  - dwa stany napięcia odpowiadają wartościom 0 i 1
  - wartości logiczne prawda i fałsz odpowiadają wartościom 0 i 1
- wady
  - przyzwyczajenie do systemu dziesiętnego
  - zapis wymaga wielu znaków

System szesnastkowy (hexadecymalny)

- Podstawą systemu jest liczba 16.
- Liczbę zapisujemy za pomocą znaków 0, 1, 2, 3, 4, 5, 6, 7, 8 i 9 oraz A = 10, B = 11, C = 12, D = 13, E = 14 i F = 15 o przypisanych wartościach.
- adresy komórek pamięci są szesnastkowe,
- zawartość komórek (bajtów) jest określana w systemie szesnastkowym.

Przykład

$(A1E)_{16} = 10x16^2 + 1x16^1 + 14x16^0 = (2560+16+14)_{10} = 2590_{10}$

Przykłady

$$6253_8 = 6x8^3 + 2x8^2 + 5x8^1 + 3x8^0 = 3243$$

$$110011_2 = 1x2^5+1x2^4+0x2^3+0x2^2+1x2^1+1x2^0 = 32+16+2+1 = 51$$

$$ABCD_{16} = 10x16^3 + 11x16^2 + 12x16^1 + 13x16^0 = 48350$$

$$124321_5 = 1x5^5 + 2x5^4 + 4x5^3 + 3x5^2 + 2x5^1 + 1x5^0 = 4961$$

Przykłady

Jak zapisać liczbę

$$6253_8$$

w systemie o podstawie 5?

$$6253_8 = 6x8^3 + 2x8^2 + 5x8^1 + 3x8^0 = 3243_{10}$$

$$3243_{10} = 100433_5$$

$$100433_5 = 1x5^5 + 0x5^4 + 0x5^3 + 4x5^2 + 3x5^1 + 3x5^0 = 3243$$

Algorytm

- n - liczba cyfr w zapisie pozycyjnym danej liczby,
- p - podstawa systemu pozycyjnego w którym podana jest liczba,
- a_i - cyfra na i-tej pozycji liczby (od prawej strony),
- w - obliczana wartość liczby.

1. $w \longleftarrow 0$
2. $i \longleftarrow n-1$
3. $w \longleftarrow a_i + w * p$
4. jeżeli $i = 0$ to koniec - w jest wartością liczby
5. $i \longleftarrow i-1$
6. wróć do punktu 3.

Przykład

Wyznaczyć za pomocą algorytmu Hornera wartość liczby 2510432_6

$w \longleftarrow 0$
$w \longleftarrow 2 + 0 * 6 = 2$
$w \longleftarrow 5 + 2 * 6 = 17$
$w \longleftarrow 1 + 17 * 6 = 103$
$w \longleftarrow 0 + 103 * 6 = 618$
$w \longleftarrow 4 + 618 * 6 = 3712$
$w \longleftarrow 3 + 3712 * 6 = 22275$
$w \longleftarrow 2 + 22275 * 6 = 133652$

Metoda przeliczania liczb

Problem: dana jest liczba w systemie dziesiętnym. Znaleźć zapis liczby w systemie o podstawie p.
Wartość liczby: $a_{n-1}p^{n-1} + a_{n-2}p^{n-2} + \ldots + a_1 p + a_0$
Aby wyznaczyć cyfry $a_1, \ldots, a_{n-1}$ potrzebne są dwa działania:

- div - dzielenie całkowitoliczbowe,
- mod - reszta z dzielenia,
- dla liczb całkowitych x, y, istnieją liczby całkowite k, l, gdzie $l \leq y$, takie że $x = k * y + l$; wtedy $k = x$ div y, $l = x$ mod y.

Przykład:

$13 = 4 * \mathbf{3} + \mathbf{1}$
13 div 4 = 3
13 mod 4 = 1

Algorytm

- w - wartość liczby,
- p - podstawa systemu pozycyjnego,
- a_i - cyfra na i-tej pozycji liczby (od prawej strony).

1. $i \longleftarrow 0$
2. $a_i \longleftarrow$ reszta z dzielenia w/p
3. $w \longleftarrow$ część całkowita z dzielenia w/p
4. jeżeli $w = 0$ to koniec - wszystkie cyfry zostały znalezione
5. $i \longleftarrow i + 1$
6. wróć do punktu 2.

Przykład

Przedstawić liczbę o wartości 12768 w systemie o podstawie 4.

$a_0 \longleftarrow$ 12768 mod 4	$a_0 = 0$
$w \longleftarrow$ 12768 div 4	$w = 3192$
$a_1 \longleftarrow$ 3192 mod 4	$a_1 = 0$
$w \longleftarrow$ 3192 div 4	$w = 798$
$a_2 \longleftarrow$ 798 mod 4	$a_2 = 2$
$w \longleftarrow$ 798 div 4	$w = 199$
$a_3 \longleftarrow$ 199 mod 4	$a_3 = 3$
$w \longleftarrow$ 199 div 4	$w = 49$
$a_4 \longleftarrow$ 49 mod 4	$a_4 = 1$
$w \longleftarrow$ 49 div 4	$w = 12$
$a_5 \longleftarrow$ 12 mod 4	$a_5 = 0$
$w \longleftarrow$ 12 div 4	$w = 3$
$a_6 \longleftarrow$ 3 mod 4	$a_6 = 3$
$w \longleftarrow$ 3 div 4	$w = 0$

$(12768)_{10} = (3013200)_4$

Konwersja dwójkowo-ósemkowa

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

$1111000101010010001010101_2$

$$\begin{array}{ccccccccc} 001 & 111 & 000 & 101 & 010 & 010 & 001 & 010 & 101 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 1 & 7 & 0 & 5 & 2 & 2 & 1 & 2 & 5 \end{array}$$

$1111000101010010001010101_2 = 170522125_8$

Konwersja ósemkowo-dwójkowa

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

641370_8

$$\begin{array}{ccccccccc} 6 & 4 & 1 & 3 & 7 & 0 & 1 & 2 & 5 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 110 & 100 & 001 & 011 & 111 & 000 & 001 & 010 & 101 \end{array}$$

$641370125_8 = 110100001011111000001010101_2$

Konwersja dwójkowo-szesnastkowa

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

$101000011011000000110100110110000101_2$

$$\begin{array}{ccccccccc} 1010 & 0001 & 1011 & 0000 & 0011 & 0100 & 1101 & 1000 & 0101 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ A & 1 & B & 0 & 3 & 4 & D & 8 & 5 \end{array}$$

$101000011011000000110100110110000101_2 = A1B034D85_{16}$

Konwersja szesnastkowo-dwójkowa

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

$A1B034D85_{16}$

$$\begin{array}{ccccccccc} A & 1 & B & 0 & 3 & 4 & D & 8 & 5 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 1010 & 0001 & 1011 & 0000 & 0011 & 0100 & 1101 & 1000 & 0101 \end{array}$$

$A1B034D85_{16} = 101000011011000000110100110110000101_2$

Tabliczka dodawania

Do wykonywania dodawania niezbędna jest znajomość tabliczki dodawania, czyli wyników sumowania każdej cyfry z każdą inna:

$0+0=0$

$1+0=1$

$0+1=1$

$1+1=10$

Na przykład:

$$\begin{array}{r} 0101 = 5_{10} \\ +0110 = 6_{10} \\ \hline 1011 = 11_{10} \end{array} \qquad \begin{array}{r} 1100 = 12_{10} \\ +0011 = 3_{10} \\ \hline 1111 = 15_{10} \end{array} \qquad \begin{array}{r} 1010 = 10_{10} \\ +1010 = 10_{10} \\ \hline 10100 = 20_{10} \end{array} \qquad \begin{array}{r} 1111 = 15_{10} \\ +0001 = 1_{10} \\ \hline 10000 = 16_{10} \end{array}$$

Problem

- W pamięci komputera liczby binarne przechowywane są w postaci ustalonej liczby bitów (np. 8, 16, 32 bity).
- Jeśli wynik sumowania np. liczb 8 bitowych zajmuje więcej niż 8 bitów, to najstarszy bit (dziewiąty bit) zostanie utracony.
- Sytuacja taka nazywa się **nadmiarem** (ang. overflow) i występuje zawsze, gdy wynik operacji arytmetycznej jest większy niż górny zakres danego formatu liczb binarnych (np. dla 8 bitów wynik większy od $2^8 - 1$, czyli większy od 255):

Np. $11111111_2 + 00000001_2 = 1|00000000_2(255 + 1 = 0)$

Zadania

Zsumować liczby binarne:
1111001_2 oraz 10010_2
01111111_2 oraz 1_2

Tabliczka odejmowania

Do wykonywania odejmowania niezbędna jest znajomość tabliczki odejmowania:
$0 - 0 = 0$
$1 - 0 = 1$
$0 - 1 = 1$ i pożyczka do następnej pozycji
$1 - 1 = 0$
Pożyczka oznacza konieczność odjęcia 1 od wyniku odejmowania cyfr w następnej kolumnie.

$$\begin{array}{rl} 11111 & \\ 1101110 = & 110_{10} \\ - \quad 1111 = & 15_{10} \\ \hline 1011111 = & 95_{10} \end{array}$$

Problem

- Jeśli od liczby mniejszej odejmiemy większą, to wynik będzie ujemny.
- Otrzymujemy same jedynki, a pożyczka nigdy nie zanika.
- Sytuacja taka nazywa sie **niedomiarem** (ang. underflow) i występuje zawsze, gdy wynik operacji arytmetycznej jest mniejszy od dolnego zakresu formatu liczb binarnych (dla naturalnego kodu dwójkowego wynik mniejszy od zera).

Np.

$$\begin{array}{r} 11111111 \\ 00000000 \\ -00000001 \\ \hline 11111111 \end{array}$$

Zadania

Odjąć liczby binarne:
$10000000_2 - 0000001_2 =$???
$10101010_2 - 01010101_2 =$???

Pytania

- Obliczyć za pomocą algorytmu Hornera wartość liczby 2210112 zapisanej w systemie pozycyjnym przy podstawie 3.
- Przeliczyć wartość 12785 na system piątkowy.
- Przeliczyć wartość $886597_{(10)}$ na system dwójkowy.
- Przedstawić w systemie czwórkowym liczbę $325748_{(10)}$.
- Zastosować konwersję dwójkowo-ósemkową do liczby 11011001110001.
- Zastosować konwersję szesnastkowo-dwójkową do liczby D7B55D.
- Wykonać dodawanie w systemie dwójkowym.