

Zmiennoprzecinkowa reprezentacja liczb

układ dwójkowy

$$x = \pm m \cdot 2^c$$

gdzie:

m – mantysa – liczba z zakresu [1, 2),

c – cecha – liczba całkowita.

mantysa jest reprezentowana na n bitach (n cyfr dwójkowych)

1.000...000

1.000...001

1.000...010

1.000...011

...

1.111...111

gdzie pierwsza cyfra nie może być zerem!

reprezentacja zera:

m = 0.000...000 oraz c = 0.

odpowiednik dziesiętny

$$x = \pm m \cdot 10^c$$

gdzie:

m – mantysa – liczba z zakresu [1, 10)

c – cecha – liczba całkowita.

mantysa jest reprezentowana na n pozycjach dziesiętnych (n cyfr dziesiętnych)

1.000...000

1.000...002

1.000...003

1.000...004

...

9.999...999

gdzie pierwsza cyfra nie może być zerem!

reprezentacja zera:

m = 0.000...000 oraz c = 0.

W obu układach działania arytmetyczne realizowane są podobnie. Można więc ograniczyć się do opisu w układzie dziesiętnym.

mnożenie

$$x = x_1 * x_2 = (m_1 * 10^{c_1}) * (m_2 * 10^{c_2}) = m_1 * m_2 * 10^{c_1+c_2}$$

gdzie, dla czytelności pominięto znak mantysy

Aby wynik mnożenia $m_1 * m_2$ (wynikowa mantysa) mieścił się w wymaganym zakresie $[1, 10)$ modyfikowana jest wartość wynikowej cechy.

Ponieważ wynik mnożenia $m_1 * m_2$ ma $2n$ cyfr jest on zaokrąglany do n cyfr. Wynika stąd, że błąd zaokrąglenia dotyczy ostatniej, najmniej znaczącej cyfry mantysy wyniku.

dzielenie

$$x = x_1 / x_2 = (m_1 * 10^{c_1}) / (m_2 * 10^{c_2}) = m_1 / m_2 * 10^{c_1-c_2} \quad (x_2 \neq 0)$$

Aby wynik dzielenia m_1 / m_2 (wynikowa mantysa) mieścił się w wymaganym zakresie $[1, 10)$ modyfikowana jest wartość wynikowej cechy.

Ponieważ wynik dzielenia m_1 / m_2 może mieć więcej niż n cyfr jest on zaokrąglany do n cyfr. Wynika stąd, że tak jak poprzednio, błąd zaokrąglenia dotyczy ostatniej, najmniej znaczącej cyfry mantysy wyniku.

dodawanie i odejmowanie

Dodawanie liczb w tej reprezentacji wymaga a by obie liczby miały tą samą cechę

$$x = x_1 + x_2 = (m_1 * 10^{c_1}) + (m_2 * 10^{c_2}) = (m_1 + m_2 * 10^{c_2-c_1}) * 10^{c_1} \quad (c_1 \geq c_2)$$

Jeśli $m_2 * 10^{c_2-c_1}$ jest mniejsze od najmniejszej liczby możliwej do zapisania w reprezentacji mantysy, np.

$$\begin{aligned} m_1 &= 1.000...000 \\ m_2 * 10^{c_2-c_1} &= 0.000...0001 \end{aligned}$$

wtedy $x_1 + x_2 = x_1$ (x_2 nie zostanie dodane).

Powyższe zjawisko dotyczy również odejmowania.

Wprowadzenie błęd wprowadzany przez zaokrąglenie w pojedynczej operacji jest taki sam jak w przypadku mnożenia lub dzielenia, ale prowadzi to do poważniejszych konsekwencji:

1. nieprzenienność dodawania

Przypuśćmy, że mamy p liczb x_i ($i = 1 \dots p$) takich, że $x + x_i = x$ ($i = 1 \dots p$)

oraz ich suma $x_s = x_1 + x_2 + \dots + x_p$ jest taka, że $x + x_s > x$

Wtedy:

$$x_1 + x_2 + \dots + x_p + x > x + x_1 + x_2 + \dots + x_p$$

(wynik zależy od kolejności sumowania !)

2. redukcja miejsc znaczących

Przypuśćmy, że mamy dwie liczby x_1 i x_2 , które mają te same cechy, a ich mantysy (m_1, m_2) różnią się tylko na 2 ostatnich cyfrach, np.

$$m_1 = 1.234...567 \text{ (} m \text{ cyfr znaczących)}$$

$$m_2 = 1.234...555 \text{ (} m \text{ cyfr znaczących)}$$

W wyniku odejmowania mantys otrzymujemy:

$$m' = 0.000...012 \text{ (2 cyfry znaczące)}$$

co daje mantysę wynikową (po sprowadzeniu do zakresu $[1, 10)$):

$$m = 1.2??...??? \text{ (2 cyfry znaczące)}$$

(argumenty odejmowania miały po m cyfr znaczących a wynik tylko 2 cyfry znaczące !)

Zjawisko redukcji miejsc znaczących może być źródłem niestabilności numerycznej algorytmów.

test w arytmetyce float:

n	x	$dx = 1/n!$	$y = x + dx$	$(y - x)/dx$
1	1.000000	1.000000e+000	2.000000000000000000	1.0000000000000000
2	1.000000	5.000000e-001	1.500000000000000000	1.0000000000000000
3	1.000000	1.666667e-001	1.16666662693023680	0.999999761581421
4	1.000000	4.166667e-002	1.04166662693023680	0.999999046325684
5	1.000000	8.333334e-003	1.00833332538604740	0.999998986721039
6	1.000000	1.388889e-003	1.00138890743255620	1.000013351440430
7	1.000000	1.984127e-004	1.00019836425781250	0.999755859375000
8	1.000000	2.480159e-005	1.00002479553222660	0.999755859375000
9	1.000000	2.755732e-006	1.00000274181365970	0.994949340820313
10	1.000000	2.755732e-007	1.00000023841857910	0.865173339843750
11	1.000000	2.505211e-008	1.000000000000000000	0.0000000000000000

test w arytmetyce double:

n	x	$dx = 1/n!$	$y = x + dx$	$(y - x)/dx$
1	1.000000	1.000000e+000	2.000000000000000000	1.0000000000000000
2	1.000000	5.000000e-001	1.500000000000000000	1.0000000000000000
3	1.000000	1.666667e-001	1.16666666666666670	1.0000000000000000
4	1.000000	4.166667e-002	1.04166666666666670	1.0000000000000002
5	1.000000	8.333333e-003	1.00833333333333330	0.9999999999999996
6	1.000000	1.388889e-003	1.00138888888888890	0.9999999999999996
7	1.000000	1.984127e-004	1.00019841269841270	0.9999999999999996
8	1.000000	2.480159e-005	1.00002480158730170	1.0000000000003354
9	1.000000	2.755732e-006	1.00000275573192240	1.0000000000012307
10	1.000000	2.755732e-007	1.00000027557319230	1.0000000000092882
11	1.000000	2.505211e-008	1.00000002505210840	1.0000000000092882
12	1.000000	2.087676e-009	1.00000000208767560	0.999999946913022
13	1.000000	1.605904e-010	1.00000000016059040	0.999999946913022
14	1.000000	1.147075e-011	1.00000000001147080	1.000006860294888
15	1.000000	7.647164e-013	1.00000000000076470	1.000006860294888
16	1.000000	4.779477e-014	1.00000000000004770	0.998845412141236
17	1.000000	2.811457e-015	1.00000000000000290	1.026720167828898
18	1.000000	1.561921e-016	1.00000000000000020	1.421612540070782
19	1.000000	8.220635e-018	1.00000000000000000	0.0000000000000000