# Системы счисления: продолжение

Лекция 5

# Представление действительных чисел

$$S = a_{l-1}a_{l-2}a_{l-3}\ldots a_2a_1a_0.a_{-1}a_{-2}a_{-3}\ldots a_{-1}\ldots_{(b)}$$

$$N(s) = a_{l-1}b^{l-1} + \ldots + a_0b^0 + a_{-1}b^{1-} + \ldots + a_{-i}b^{-i} + \ldots = \sum_{i=\infty}^{l-1} a_i b^i. \qquad (5)$$

Если в дробной части числа конечное число знаков *k*, то нижний индекс суммы равен *—к* .

$$N_f(s) = (a_{-1} + (a_{-2} + (a_{-3} + (\ldots)/b)/b)/b)/b, \qquad (6)$$

**0.375=(3+(7+5/10)/10)/10=(3+(7+(5+0)/10)/10)/10**

# Связь дробной части числа со значением

$$N_f(s) = (a_{-1} + (a_{-2} + (a_{-3} + (...)/b)/b)/b)/b, \qquad (6)$$

$$\begin{cases} N_{-k-1} = 0; \\ N_{-i} = (a_{-i} + N_{-i+1})/b; \quad \text{где } i = k, \ldots, 1; \\ N_f(s) = N_{-1}. \end{cases} \qquad (7)$$

# Примеры

$$N(\text{«}1.101_{(2)}\text{»}) = 1\cdot 2^{0} + 1\cdot 2^{-1} + 0\cdot 2^{-2} + 1\cdot 2^{-3}$$
$$= 1 + 0.5 + 0.125$$
$$= 1.625$$

$$N_f(\text{«}1.101_{(2)}\text{»}) = (1 + (0 + (1 + 0)/2)/2)/2$$
$$= (1 + (0 + 0.5)/2)/2$$
$$= (1 + 0.25) / 2 = 0.625$$

$$N_f(\text{«}0.01_{(3)}\text{»}) = 1\cdot 3^{-2} = \frac{1}{9} = 0.(1)$$

Целая часть числа $N_f$*$b$ ($0 < N_f < 1$) равна первой цифре дробной части числа $N_f$

# Алгоритм А4: перевод дробной части из 10-с. с. в $b$-с.с

**Вход**: $N_f$ ( $0 \le N_f < 1$), $b > 1$;

$i := -1$;

цикл

$a_i := [N_f \cdot b]$; ([x] – взятие целой части числа)

$N_f := N_f \cdot b - a_i$; ($N_f$ остается в том же диапазоне )

$i := i - 1$;

пока $N_f \neq 0$;

$k := i$;

**Выход:** набор $a_i, k$ (число значащих цифр).

Алгоритм **А4** может не завершиться, если данное число не представимо конечной дробью в $b$-$c.c$

Требуется ***k*** умножений (выражение $N_f$*$b$ можно вычислять в цикле один раз и хранить в промежуточной переменной).

Пример:

$$0.375 = (3 + (7 + 5/10)/10)/10 = (3 + (7 + (5 + 0)/10)/10)/10$$

$$N_f = 0.375;\ N_{-1} = 0.375;$$

$$0.375 * 2 = 0.75;\ N_{-2} = 0.75; a_{-1} = 0;$$

$$0.75 * 2 = 1.5;\ N_{-3} = 0.5; a_{-2} = 1;$$

$$0.5 * 2 = 1.0;\ N_{-4} = 0; a_{-3} = 1;$$

$$0.375_{(10)} = 0.011_{(2)}$$

$$1/4 + 1/8 = 3/8 = 0.375$$

## Теорема Т2

Несократимая дробь $p/q$ конечно представима в системе счисления с основанием $b$ в том и только в том случае, когда все числа из разложения $q$ на простые множители входят в такое же разложение $b$ (количество повторений не учитывается).

## Пример

**121/675** конечна в 15-с.с.:

$675 = 3^3 * 5^2;\ 15 = 3*5;$

$1/675 = 5*15^{-3} = 0.005_{(15)};$

$121*5/15^{-3} = (2*15^2 + 10*15^1 + 5)/15^{-3} = 2/15^{-1} + 10/15^{-2} + 5/15^{-3}$

$121/675 = 0.2A5_{(15)};$

**1/10 бесконечна в 2-с.с. !!!!**

# Алгоритм A5:
# (перевод дробной части из *b*-с.с. в 10-с.с)

**Вход:** $b > 1$, $\kappa > 0$ (число дробных цифр), набор $a_i$

$N_f := 0; S := 1$ ($S$ — накапливает степень, $N_f$ — значение )

**цикл** по $i$ от -1 вниз до $-k$

$$N_f := N_f + a_i \cdot S;$$

$$S;:= S / b$$

**конец цикла**

**Выход:** $N_f$

- $2k$ операций *, /
- $k$ операций +

# Алгоритм А6:

## перевод дробной части из b-с.с. в 10-с.с.

( из формулы (7) по схеме Горнера)

**Вход:** $b > 1$, $k > 0$ (число цифр), набор $a_i$

$N_f := 0;$

**цикл** по $i$ от $-k$ до $-1$

$$N_f := (a_i + N_f) / b$$

**конец цикла;**

**Выход:** $N_f$

$k$ операций + и /

Число N в $b$-с.с. имеющее $k$ дробных цифр, при умножении на $b^k$ становится целым (это умножение соответствует сдвигу точки на $k$ позиций вправо)

## Алгоритм А7

- найти целое $N_1 = N * b_1^k$ (умножением или сдвигом точки);
- выполнить для $N_1$ один из алгоритмов А1 или А2, затем А3;
- разделить полученный результат на $b_1^k$ в системе $b_2$

# Пример

**Перевести $101.101_{(2)}$ в 10-с.с.**

1) умножим на $2^3$ → $101101_{(2)}$
2) переведем в 10-с.с. → 45
3) разделим: $45/8 = 5.625_{(10)}$

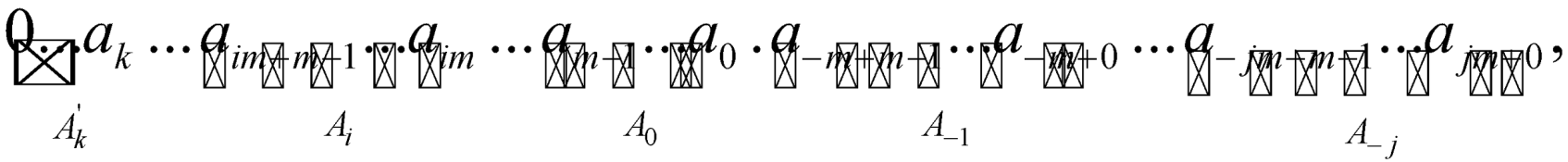
$$101.101=1*2^2+1*2^0+1*2^{-1}+1*2^{-3}=5+1/2+1/8=5.625$$

# Кратные системы счисления

Если основания двух систем счисления $b_1$ и $b_2$ связаны соотношением $b_2 = b_1^m$ для некоторого натурального $m$, то такие системы счисления называются *кратными*.

Перевод числа из одной с. с. в другую для таких систем можно выполнить проще.

Сгруппируем цифры в $b_1$-записи числа по $m$ от точки влево и вправо (добавив при нехватке цифр нужное количество незначащих нулей):



затем также сгруппируем слагаемые в формуле (5) (они содержат множитель $b_1$ в степени, равной индексу цифры), вынесем за скобки из каждой группы $i$ общий множитель

$$(b_1^{im} = (b_1^{m})^{i} = b_2^{i})$$

и обозначим для каждой группы

$$A_i = a_{im+m-1} b_1^{m-1} + a_{im+m-2} b_1^{m-2} + \ldots + a_{im+1} b_1^{1} + a_{im} b_1^{0} \qquad (8)$$

Тогда значение исходного числа может быть представлено в виде:

$$N(S') = A_{k'} * b_2^{k'} + \ldots + A_i * b_2^{i} + \ldots + A_0 * b_2^{0} + A_{-1} * b_2^{-1} + \ldots A_{-j}\ b_2^{-j},$$

что по определению совпадает со значением записи того же числа в $b_2$-с.с. с цифрами $A_i$, если заметить, что $A_i$, действительно могут принимать все значения от 0 до $b_1^{m} - 1 = b_2 - 1$.

# Таблицы соответствия последовательностей цифр кратных с.с.

| 9-с.с. | 3-с.с. |
|---|---|
| 0 | 00 |
| 1 | 01 |
| 2 | 02 |
| 3 | 10 |
| 4 | 11 |
| 5 | 12 |
| 6 | 20 |
| 7 | 21 |
| 8 | 22 |

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

| 16-с.с. | 2-с.с. |
|---|---|
| 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 |

# **Алгоритм А8:** перевод из меньшей кратной с.с. в большую

**Вход:** $b_1 > 1$, $b_2 = b_1^m$, $b_1$ - представление числа;

- разбить число на группы по $m$ цифр, начиная от точки, в обе стороны (если в крайних группах цифр меньше $m$, добавить незначащие нули: в целой части спереди, в дробной сзади);
- заменить каждую группу $b_2$-цифрой по формуле (8) или таблице.

**Выход:** $b_2$ -представление исходного числа.

## **Алгоритм А9:** перевод из большей кратной с.с. в меньшую

**Вход:** $b_1 > 1$, $b_2 = b_1^m$; $b_2$-представление числа;

- заменить каждую $b_2$-цифру цепочкой из $m$ $b_1$-цифр по формуле (8) или таблице;
- отбросить незначащие нули слева и справа.

**Выход:** $b_1$-представление исходного числа.

# Универсальные алгоритмы для арифметических операций

- Все так называемые *численные* алгоритмы для арифметических операций сложения, вычитания, умножения и деления (в том числе, вычисления «столбиком») являются *символьными,* потому что оперируют входными, выходными и промежуточными данными как строками символов.
- Символьные вычисления являются *формальными в* том смысле, что манипулируют только знаками, не обращаясь к их значениям.
- *Абстрагирование* от смысла данных различной природы и описание алгоритма в терминах чисто символьных преобразований является одним из основных методов программирования обработки данных произвольной природы

# Алгоритм A10: сложение двух чисел

**Вход:** две строки цифр, представляющие слагаемые;

- **выравнивание:** расположить слагаемые одно под другим в произвольном порядке так, чтобы разряды с одинаковым весом находились друг под другом; если какое-то число короче других слева или справа, дополнить его нулями;
- **начальные установки**:

  обнулить цифру переноса в следующий разряд;

  установить результат равным пустой строке;
- ***цикл*** по текущему разряду от младшего до старшего:

  определить сумму переноса и цифр в столбце текущего разряда чисел; младшую цифру суммы записать в текущий разряд результата, старшую — в перенос;

  *конец цикла;*
- **окончание:** если перенос не равен 0, то дописать перенос в начало результата

**Выход:** строка, представляющая результат.

Единственное место в этом алгоритме, где присутствует обращение к значениям цифровых символов, — это поразрядное сложение в цикле.

Действительно, из одного лишь вида знаков «2» и «3» нельзя извлечь информацию, что результатом их сложения будет знак «5».

Эти сведения можно задать, например, двумя таблицами сложения: в одной для каждой пары цифр записать младшую цифру результата, в другой — цифру переноса («0» или «1»);

исчерпав таким образом все немногочисленные случаи, можно заменить операцию сложения значений операцией выборки знака из таблицы.

Чтобы учесть сложение с переносом, можно завести две пары таблиц или записать в каждую клетку по две цифры.

Алгоритм А10 замечателен тем, что применим к произвольной позиционной с. с. при соответствующей замене таблиц сложения.

| + | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 1 | 0 |

| ++ | 0 | 1 |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |

| * | 0 | 1 |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |

Затраты памяти на хранение чисел и времени на выполнение операций с ними зависят от длины записи числа в цифрах рабочей системы счисления.

Для заданной *b*-с. с. следующие величины:

$k_n$ — длина записи (натурального) числа ***N***,

$N_k$ — максимальное натуральное число, записываемое ***k*** цифрами,

связаны соотношениями:

$k_n = [log_b N] + 1$, где $[x]$ — наибольшее целое, не превышающее ***x***;

$N_k = b^k - 1$.

Верхние оценки для размера результата арифметической операции над парой целых чисел $N_1$ и $N_2$ (пусть $N_1 > N_2$):

для сложения и вычитания — $k_{N1} + 1$,

для умножения — $k_{N1} + k_{N2}$,

для деления — $k_{N1} + 1$, (так как $N_2 > 1$).