Компьютерная алгебра
(курс лекций)

Игорь Алексеевич Малышев
Computer.Algebra@yandex.ru

Лекция 14

Вычисления в модулярной арифметике

Содержание лекции

- Целые числа по модулю m
- Греко-китайская теорема об остатках
- Арифметика вычетов

План лекции: тема подраздела

- **Целые числа по модулю m**
- Греко-китайская теорема об остатках
- Арифметика вычетов

Целые числа по модулю m

Вводные замечания

Замечание. При фиксированном $m > 1$ по определению (см. лекцию 6) $b \equiv_m a$ тогда и только тогда, когда $b - a = m * q$ для некоторого q (иными словами, тогда и только тогда, когда m делит $(b - a)$).

Замечание. Формулы $b \equiv_m a$ и $b \equiv a \pmod{m}$ являются синонимами (читаются: «b равно a по модулю m» или «b сравнимо с a по модулю m»). Обозначения предложены Гауссом.

Замечание. Множество классов эквивалентности $\mathbf{Z} / \equiv_m$,
которое обозначается также $\mathbf{Z}_m$,
называется множеством вычетов или целыми числами по модулю m.

Замечание. Важнейшие алгебраические свойства целых чисел верны и для целых чисел по модулю m.

Целые числа по модулю m

Свойства отношения эквивалентности $\equiv_m$

Теорема.

A. Если $a \equiv b \pmod{m}$ и d делит m, то $a \equiv b \pmod{m}$.

B. Если $a \equiv b \pmod{m}$ и $a \equiv b \pmod{n}$, то $a \equiv b \pmod{[m, n]}$.

C. Если $a \equiv c \pmod{m}$ и $b \equiv d \pmod{m}$, то :

$a + b \equiv c + d \pmod{m}$

$a - b \equiv c - d \pmod{m}$

$a * b \equiv c * d \pmod{m}$

D. (свойство сокращения) Если $a * b \equiv a * c \pmod{m}$,

то $b \equiv c \pmod{m / d}$, где $d = (a, m)$.

В частности, если $(a, m) = 1$,

то $a * b \equiv a * c \pmod{m}$ влечёт за собой $b \equiv c \pmod{m}$

Целые числа по модулю m

Конгруэнция $\equiv_m$

Рассмотрим классы эквивалентности **a** и **b** в $\mathbf{Z}_m$.

Определим сумму **a** + **b** и произведение **a** * **b** этих классов
через сумму и произведение их представителей.

Мы имеем сюръективное отображение :

$s : \mathbf{Z} \to \mathbf{Z}_m$

Положим :

a + **b** = s (a + b)

a * **b** = s (a * b)

Из предыдущей теоремы вытекает, что это определение корректно,
т.е. если $c \in$ **a** и $d \in$ **b** – другие представители,
то мы получим те же классы эквивалентности для суммы и произведения, т.к. :

s (c + d) = s (a + b)

s (c * d) = s (a * b)

Таким образом, отношение эквивалентности $\equiv_m$ сохраняет алгебраические свойства
(в частности, операции сложения и умножения), т.е. является конгруэнцией.

Целые числа по модулю m

Модулярные арифметики

Используя свойство евклидовости,
можно рассматривать арифметику целых чисел по модулю m
как арифметику остатков или **модулярную арифметику**.

Полная система остатков по модулю m состоит из m целых чисел,
по одному представителю из каждого класса эквивалентности.

Наиболее часто используются две следующие системы:

(1) система **неотрицательных** остатков по модулю m ,
состоящая из чисел $0 , 1 , 2 , \dots , m - 1$

(2) система наименьших по абсолютной величине остатков
(**симметричная** система остатков),
состоящая из чисел $0 , \pm 1 , \pm 2 , \dots , \pm (m - 1) / 2$
для нечётного числа m .

Целые числа по модулю m

Конгруэнтность целых чисел

Запишем целое число a следующим образом:

$a = m * q + r$, где $0 \le r \le m$

Остаток r (который обозначается также в виде $r_m(a)$ или $r(a)$), называется **остатком по модулю m** .

Утверждение. Два целых числа конгруэнтны тогда и только тогда, когда они имеют одинаковые остатки по модулю m (т.е. используя введённые обозначения: $b \equiv a \pmod m \Leftrightarrow r_m(a) \equiv r_m(b)$)

В соответствии с утверждением для целого числа a его класс эквивалентности :

$\mathbf{a} = a + m\mathbf{Z}$

является в точности множеством чисел, остатки которых совпадают с $r(a)$.

Остатки $0, 1, 2, \ldots, m-1$ являются представителями классов эквивалентности, поэтому иногда отождествляют класс эквивалентности с представляющим его остатком и рассматривают $\mathbf{Z}_m$ просто как множество $\{0, 1, \ldots, m-1\}$

Целые числа по модулю m

Проблема обратных элементов в Z_m

Числами в модулярной арифметике являются остатки по модулю m .

Противоположный (**аддитивный обратный**) элемент к произвольному числу $a \in \mathbf{Z}_m$ всегда существует и равен $m - a$.

Мультипликативный обратный элемент к $a \in \mathbf{Z}_m$, определяемый как решение следующего уравнения $a * x \equiv 1 \pmod{m}$ существует не всегда.

Теорема. Пусть $a \in \mathbf{Z}_m$.
Тогда a имеет мультипликативный обратный элемент по модулю m в том и только в том случае, когда $(a , m) = 1$ (т.е. a и m – взаимно простые).
Доказательство :
С помощью расширенного алгоритма Евклида можно найти целые числа x и y такие, что $(a , m) = a * x + m * y$, откуда вытекает, что $a * x \equiv (a , m) \pmod{m}$
Если $(a , m) = 1$, то предыдущее сравнение означает, что x является мультипликативным обратным к a по модулю m .
Если же $(a , m) > 1$, то не существует числа x , для которого выполняется сравнение $a * x \equiv 1 \pmod{m}$, т.к. $a * x = 1 + k * m$ влечёт $(a , m) = 1$.

Целые числа по модулю m

$\mathbf{Z}_m$ – это кольцо или поле ?

Теорема. Для всякого целого числа $m > 1$ множество $\mathbf{Z}_m = \{ 0, 1, \ldots, m - 1 \}$
с операциями сложения и умножения по модулю m
является коммутативным кольцом с единицей
и называется кольцом вычетов по модулю m.
Оно является полем тогда и только тогда, когда m – простое число.

Доказательство :

С помощью определённого ранее сюръективного отображения
$s : \mathbf{Z} \to \mathbf{Z}_m$, а также определения сложения и умножения в $\mathbf{Z}_m$,
можно легко вывести, что $\mathbf{Z}_m$ является коммутативным кольцом с единицей,
из того, что $\mathbf{Z}$ является таким кольцом.

Пусть теперь m – простое число. Тогда все ненулевые элементы в $\mathbf{Z}_m$ обратимы
и, следовательно, является $\mathbf{Z}_m$ полем.
С другой стороны, если m не является простым, то $\mathbf{Z}_m$ – не поле.
Чтобы убедиться в этом, запишем : $m = a * b$, $a < m$, $b < m$
Тогда $s(a) * s(b) = s(m) = s(0)$
Но $s(a) \neq s(0)$ и $s(b) \neq s(0)$, откуда вытекает, что $s(a)$ и $s(b)$ являются делителями нуля.

Целые числа по модулю m

Примеры колец и полей $\mathbf{Z}_m$

Пример 1.
Кольцо $\mathbf{Z}_5 = \{ 0 , 1 , 2 , 3 , 4 \}$
является полем, т.к. все его ненулевые элементы 1 , 2, 3, 4 обратимы (обратные к ним элементы – это 1 , 3, 2, 4 соответственно).

Пример 2.
Кольцо $\mathbf{Z}_8 = \{ 0 , 1 , 2 , 3 , 4 , 5, 6 , 7 \}$
полем не является, т.к. в нём есть делители нуля
(например, $2 * 4 = 0 \pmod 8$)

Замечание. Мультипликативная группа кольца $\mathbf{Z}_m$ имеет $\phi(m)$ элементов, где ϕ – это функция Эйлера (иными словами, это группа порядка $\phi(m)$).

Теорема. Характеристика конечного поля – простое число.

Целые числа по модулю m

Вычисление a^{-1} в $\mathbf{Z}_m$

В соответствии со следующими утверждениями :

(1) малая теорема Ферма (см. лекцию 12),
которая утверждает, что если m – простое число
и a – произвольное целое число, не делящееся на m ,
то $a^{m-1} \equiv 1 \pmod{m}$

(2) следствие из малой теоремы Фермы, которое утверждает,
что если m – простое число, то в кольце $\mathbf{Z}_m$ выполняется равенство
$a^{-1} = a^{m-2}$

(3) теорема Эйлера, обобщающая малую теорему Ферма
(m – не обязательно простое число),
которая утверждает, что если $(a, m) = 1$, то $a^{\phi(m)} \equiv 1 \pmod{m}$

(4) следствие из теоремы Эйлера, которое утверждает,
что в кольце $\mathbf{Z}_m$ из $(a, m) = 1$ следует, что $a^{-1} = a^{\phi(m)-1}$

для вычисления мультипликативного обратного к a по модулю m элемента
нужно возвести a в некоторую степень k , которая равна либо $m - 2$, либо $\phi(m) - 1$.

Целые числа по модулю m

Методы возведения в степень

Классический метод возведения числа a в степень k, использующий «грубую силу», требует выполнения k умножений.
Таким образом, временная сложность метода равна $O(m)$, т.к. $\phi(m) < m$

Бинарный метод, который был известен в Индии 2000 лет назад, является более эффективным. Он работает следующим образом.
Запишем k в двоичной системе счисления,
опустив нули перед первой значащей цифрой :

$$k = \Sigma k_i 2^i \text{ для всех } 0 \le i \le n-1, \text{ где } n \text{ – разрядность двоичного представления}$$

Заменим каждую цифру «1» на строку «SM_a» (SM – означает Square + Multiply)
и каждую цифру «0» на строку «S» (S – означает Square).
После этого вычеркнем слева строку «SM_a».
Таким образом, получилась последовательность букв,
которая представляет собой правило для вычисления a^k,
если интерпретировать «S» как «возвести в квадрат и взять остаток по модулю m»,
а «M_a» как «умножить на a и взять остаток по модулю m».

Замечание. Процедура бинарного метода работает слева направо
по отношению к битовому представлению числа k. Однако удобнее работать справа налево,
т.к. в этом случае умножение на 2 – это просто сдвиг вправо на один разряд.

Целые числа по модулю m

Алгоритм бинарного метода

Вход:
Ненулевые a , k , m
a – элемент $\mathbf{Z}_m$
$k = \Sigma\, k_i\, 2^i$ для всех $0 \le i \le n - 1$
$k = m - 2$ или $k = \phi(m) - 1$

Выход:
a^{-1}
(мультипликативный обратный к a элемент по модулю m)
$a^{-1} = a^k$ в кольце $\mathbf{Z}_m$

[Инициализация]
$K := k$, $B := 1$, $A := a$
[Вычисление следующего бита]
$q := [K / 2]$, $r := K - 2 * q$, $K := q$
Если $r = 0$ **То Переход** к шагу **[Возвести в квадрат и взять остаток по модулю m]**
[Умножить и взять остаток по модулю m]
$B := A * B \pmod m$
[Закончить ?]
Если $K = 0$ **То** вернуть $a^{-1} \pmod m := B$
[Возвести в квадрат и взять остаток по модулю m]
$A := A^2 \pmod m$
Переход к шагу **[Вычисление следующего бита]**

Целые числа по модулю m

Сложность алгоритма бинарного метода

Пусть двоичное представление числа k состоит из n бит.
Пусть j из них равны 1.
Тогда в алгоритме выполняется (n + j) умножений.
Т.к. j не больше, чем n , то имеем :

$$O(2 * n) = O(2 * \log_2 m) = O(\log_2 m)$$

Таким образом :

$$\mathbf{T}_E = O(L(m))$$

Замечание. Бинарный метод возведения в степень после определённой модификации может быть использован в качестве метода умножения целых чисел (называемого также «русским крестьянским» методом, подразумевая, что русские крестьяне использовали его потому, что якобы умели только умножать и делить на 2 и складывать).
Требуемая при этом модификация такова.
На вход алгоритма подаются произвольные целые числа – a и b . На выходе имеем B = a * b .
На первом шаге алгоритма вместо B : = 1 производим B := 0 , а вместо K := k – K := b .
На третьем шаге алгоритма вместо B : = A * B (mod m) производим B : = A + B .
На пятом шаге алгоритма вместо $A := A^2 \pmod{m}$ производим A : = A + A .

План лекции: тема подраздела

- Целые числа по модулю m
- **Греко-китайская теорема об остатках**
- Арифметика вычетов

Греко-китайская теорема об остатках

Группа обратимых элементов

Ранее было отмечено, что кольцо $\mathbf{Z}_m$ не всегда является полем,
т.к. в нём могут быть необратимые элементы

Обратимые элементы кольца $\mathbf{Z}_m$ образуют мультипликативную группу,
которая называется **группой обратимых элементов** кольца (или группой единиц кольца) $\mathbf{Z}_m$.
Эта группа обозначается $U_m = \{ a : (a, m) = 1 \}$ и имеет $\phi(m)$ элементов
(т.е. её порядок равен $\phi(m)$).

Пример. В кольце $\mathbf{Z}_8$
элементы 2 , 4 , 6 не имеют мультипликативных обратных, а элементы 1, 3, 5, 7 имеют.
Это обусловлено тем, что 1 , 3 , 5 , 7 взаимно просты с 8, а для 2 , 4 , 6 это не так.
Очевидно, что U_8 содержит 4 элемента { 1 , 3 , 5 , 7 } ,
причём каждая строка таблицы умножения в U_8 содержит перестановку элементов группы :

*	1	3	5	7
1	**1**	**3**	**5**	**7**
3	**3**	**1**	**7**	**5**
5	**5**	**7**	**1**	**3**
7	**7**	**5**	**3**	**1**

Греко-китайская теорема об остатках

Примитивный корень по модулю m

Пусть G – абелева группа из n элементов.
Для произвольного $a \in G$ введём обозначения :

$a^k = a * a * a * \ldots * a$ (k раз)

$a^0 = e$ – единичный элемент группы

Справедливо следующее обобщение теоремы Ферма :

Теорема. Если G – абелева группа, состоящая из n элементов, то для всякого $a \in G$ выполняется равенство $a^n = e$.

Замечание. Эта абстрактная версия теоремы Ферма справедлива и для неабелевых групп.

Пусть G – группа из n элементов, $a \in G$ и $S = \{ k \geq 1 : a^k = e \}$
Т.к. $a^n = e$, то S – не пусто. Кроме того S имеет наименьший элемент k_0 , который называется **порядком элемента** a .

Группа называется **циклической**, если в ней существует элемент a , степени которого ($1 , a , a^2 , \ldots$) пробегают все элементы группы.
Этот элемент называется образующим или, в случае группы U_m , **примитивным корнем по модулю m** .

Греко-китайская теорема об остатках

Свойство цикличности группы U_m

Замечание. Можно показать, что порядок любого элемента группы U_m делит $\phi(m)$.
Замечание. Примитивные корни, если они существуют,
являются в точности элементами максимального возможного порядка $\phi(m)$.
Замечание. Очевидно, что теорема Эйлера
– это следствие двух вышеуказанных замечаний.

Теорема. Группа U_m является циклической тогда и только тогда,
когда m равно 1 , 2 , 4 , p^a или $2p^a$,
где p – нечётное простое число и $a > 0$.
Значит, примитивные корни по модулю m
существуют в точности для таких значений m .

Следствие. Если m – нечётное простое число, то группа U_m циклическая
и уравнение $x^2 = 1$ в U_m не имеет решений, отличных от $x = \pm 1$

Замечание. Найдя примитивный корень a по модулю m в U_m ,
можно получить другой корень – мультипликативно обратный a^{-1} по модулю m .

Греко-китайская теорема об остатках

Уравнения по модулю m (часть 1)

Теорема. Уравнение $a * x \equiv b \pmod m$ имеет решение
тогда и только тогда, когда $(a, m) \mid b$.
Если решение существует, то оно единственно по модулю m / d, где $d = (a, m)$;
по модулю m уравнение имеет d решений.

Пример. Найдём решение уравнения $270 * x \equiv 36 \pmod{342}$
Применяя расширенный алгоритм Евклида, получим :

$$(-5) * 270 + 4 * 342 = 18 \qquad (*)$$

$$18 \mid 36$$

По приведённой выше теореме это уравнение имеет решение,
единственное по модулю $19 = 342 / 18$
Для нахождения этого решения умножим равенство (*) на $2 = 36 / 18$:

$$(-10) * 270 + 8 * 342 = 36$$

Отсюда следует, что (-10) – одно из решений уравнения по модулю 342.
Другими решениями по модулю 342 являются числа 9 , 28 , 47 , 66 , 85 , 104 и т.д.
Единственное решение по модулю 19 равно 9, т.к. $9 \equiv (-10) \pmod{19}$

Греко-китайская теорема об остатках

Уравнения по модулю m (часть 2)

Следствие (из теоремы – см. предыдущий слайд).
Уравнение $a * x \equiv 1 \pmod{m}$ имеет решение
тогда и только тогда, когда $(a , m) = 1$.
Решение $a^{-1} \pmod{m}$ единственно по модулю m
и является мультипликативным обратным
к a элементом по модулю m .

Пример.
Уравнение $2 * x \equiv 1 \pmod{26}$ не имеет решений, т.к. $(2 , 26) = 2$

В данном случае это можно показать и более простым способом :
мы ищем число x , такое, что :

$$2 * x - 1 = k * 26$$

Однако левая часть последнего уравнения – всегда нечётное число,
а правая часть – всегда чётное число.

Греко-китайская теорема об остатках

Обоснование теоремы об остатках

Утверждение. Если целое число m
может быть разложено в произведение степеней простых чисел :

$$m = p_1^{e_1} * p_2^{e_2} * \ldots * p_k^{e_k}$$

то кольцо $\mathbf{Z}_m$ также может быть «разложено»
в декартово произведение колец $\mathbf{Z}_{p_i^{e_i}}$

Пример. $\mathbf{Z}_6 = \mathbf{Z}_2 \times \mathbf{Z}_3$, т.к. $6 = 2 * 3$
Рассматривая пары (x_1 , x_2) , $x_1 \in \mathbf{Z}_2$, $x_2 \in \mathbf{Z}_3$,
получим 6 элементов кольца $\mathbf{Z}_6$:

$$(0 , 0) , (0 , 1) , (0 , 2) , (1 , 0) , (1 , 1) , (1 , 2)$$

Арифметические операции выполняются покомпонентно :

$$(x_1 , x_2) \bullet (y_1 , y_2) = (x_1 \bullet y_1 , x_2 \bullet y_2)$$

где операция $x_1 \bullet y_1$ выполняется в $\mathbf{Z}_2$ (арифметика по модулю 2) ,
а операция $x_2 \bullet y_2$ выполняется в $\mathbf{Z}_3$ (арифметика по модулю 3) .

Греко-китайская теорема об остатках

Теорема об остатках – формулировка

Замечание. Греко-китайская теорема об остатках
часто называется «китайской теоремой об остатках»,
которую в своей книге «Математический трактат» (III век н.э.)
сформулировал и доказал китайский математик **Сунь – Цзы**.
Однако греческий математик **Никомах** из Герасы
в своей книге «Введение в арифметику» (I век н.э.) также упоминает подобную задачу
– метод для определения натурального числа по остаткам,
полученным от деления этого числа на другие натуральные числа.

Теорема (греко-китайская теорема об остатках).
Пусть $m_1, m_2, \dots, m_k$ – попарно взаимно простые целые числа > 1 и пусть
$M = m_1 * m_2 * \dots * m_k$.
Тогда существует единственное неотрицательное решение по модулю M
следующей системы уравнений :

$$x \equiv a_1 \pmod{m_1}$$
$$\dots\dots\dots\dots\dots$$
$$x \equiv a_k \pmod{m_k}$$

Другими словами, отображение, которое каждому целому числу x, $0 \le x \le M - 1$
ставит в соответствие строку $(a_1, a_2, \dots, a_k)$, где $x \equiv a_i \pmod{m_l}$, $i = 1, 2, .., k$
является биекцией кольца $\mathbf{Z}_M$ на $\mathbf{Z}_{m_1} \times \mathbf{Z}_{m_2} \times \dots \times \mathbf{Z}_{m_k}$

Греко-китайская теорема об остатках

Теорема об остатках – доказательство

Доказательство (греко-китайской теоремы об остатках) (Часть 1) :
(Дадим конструктивное доказательство теоремы)
Нужно найти число x , $0 \le x \le M - 1$,
удовлетворяющее одновременно всем сравнениям $x \equiv a_i \pmod{m_i}$, $i = 1, 2, \ldots, k$
Будем решать уравнения по два одновременно.

Рассмотрим сначала первые два сравнения.
Первое сравнение $x \equiv a_1 \pmod{m_1}$ справедливо для всякого x вида
$x = a_1 + m_1 * q$, q – произвольное.
Для нахождения q подставим значение x во второе сравнение $x \equiv a_2 \pmod{m_2}$
После чего получим $x = a_1 + m_1 * q \equiv a_2 \pmod{m_2}$
Откуда $q \equiv (m_1)^{-1} (a_2 - a_1) \pmod{m_2}$
(Конечно, потребуется вычислить обратный к m_1 по модулю m_2)
Таким образом, $q = (m_1)^{-1} (a_2 - a_1) + r * m_2$ для некоторого r
Подставив значение q в выражение $x = a_1 + m_1 * q$,
получим, что решение x первых двух уравнений представляется в виде
$x = a_{12} + r * (m_1 * m_2)$, для некоторого r .

Греко-китайская теорема об остатках

Теорема об остатках – доказательство

Доказательство (греко-китайской теоремы об остатках) (Часть 2) :
Теперь первые два сравнения могут быть заменены на одно :

$x \equiv a_{12} \ (\mathrm{mod}\ m_1 m_2)$

которое мы рассматриваем по модулю произведения $m_1 * m_2$

Применим описанную выше процедуру к $x \equiv a_{12} \ (\mathrm{mod}\ m_1 m_2)$
и сравнению, которое первоначально было третьим.

Будем повторять этот процесс, пока не найдём число x ,
удовлетворяющее всем сравнениям.

Для доказательства единственности предположим, что существует :

$x' , 0 \le x' \le M - 1$,

такой, что $x' \equiv a_i \ (\mathrm{mod}\ m_i)$ для любого i .
Тогда $x - x' \equiv 0 \ (\mathrm{mod}\ m_i)$ для всех i , откуда следует, что :

$m_i \mid (x - x')$ для любого i .

Но тогда $M \mid (x - x')$ и, поскольку $| x - x' | < M$, то $x = x'$.

Греко-китайская теорема об остатках

Пример применения теоремы об остатках

Пример. Решим систему уравнений :

$$x \equiv 1 \ (\bmod\ 2)$$
$$x \equiv 2 \ (\bmod\ 5)$$
$$x \equiv 5 \ (\bmod\ 7)$$

В соответствии с процедурой, описанной в доказательстве теоремы об остатках, очевидно, что первое уравнение выполняется для $x = 1 + 2 * q$

Чтобы вычислить q , подставим x во второе уравнение. Получим :

$1 + 2 * q \equiv 2 \ (\bmod\ 5)$ или $2 * q \equiv (2 - 1) \ (\bmod\ 5)$

Теперь вычислим мультипликативный обратный элемент к $2 \ (\bmod\ 5)$, который равен 3 .

Таким образом, имеем :

$q \equiv 3 \ (\bmod\ 5)$ или $q = 3 + 5 * r$ для некоторого r

Следовательно, решением первых двух уравнений является :

$x = 1 + 2 * (3 + 5 * r) = 7 + 2 * 5 * r$, т.е. $x \equiv 7 \ (\bmod\ 2 * 5)$

Греко-китайская теорема об остатках

Пример применения теоремы об остатках

Пример (продолжение).

Теперь нужно решить систему из двух уравнений :

$x \equiv 7 \ (\text{mod } 2 * 5)$

$x \equiv 5 \ (\text{mod } 7)$

Имеем : $x = 7 + 2 * 5 * q \equiv 5 \ (\text{mod } 7)$ или $2 * 5 * q \equiv (5 - 7) = -2 \equiv 5 \ (\text{mod } 7)$
Мультипликативный обратный элемент к 10 по модулю 7
совпадает с обратным к 3 по модулю 7, который равен 5.

Далее получаем :

$q \equiv 5 * 5 \ (\text{mod } 7) \equiv 4 \ (\text{mod } 7)$ или $q = 4 + 7 * r$ для некоторого r

Следовательно, решением всех трех уравнений является число :

$x = 7 + 2 * 5 * (4 + 7 * r)$ или $x \equiv 47 \ (\text{mod } 2 * 5 * 7)$

Отметим, что $47 = 1 + \mathbf{3} * (2) + \mathbf{4} * (2 * 5)$,
где «красные» коэффициенты являются значениями q .

Греко-китайская теорема об остатках

Теорема об остатках – обобщения

Замечание. Если в теореме об остатках модули m_i
не являются взаимно простыми,
то решение существует тогда и только тогда, когда :

$$(m_i , m_j) \mid (a_i - a_j) \quad \text{для всех пар} \quad i , j$$

Если решение существует,
то оно единственно по модулю наименьшего общего кратного
$[m_1 , m_2 , \dots , m_k]$ чисел m_i .

Теорема. Пусть $m = p_1^{e_1} * p_2^{e_2} * \dots * p_k^{e_k}$
Тогда функция, которая каждому $x \in \mathbf{Z}_m$ ставит в соответствие строку :
$(x_1 , x_2 , \dots , x_k)$, где $x \equiv x_i \ (\mathrm{mod}\ p_i^{e_i})$, $i = 1 , 2 , \dots , k$
является кольцевым изоморфизмом
(т.е. взаимно однозначным гомоморфизмом на)
кольца $\mathbf{Z}_m$ и кольца строк $(x_1 , x_2 , \dots , x_k)$, где $x_i \in \mathbf{Z}_{p_i^{e_i}}$, для $i = 1 , 2 , \dots , k$
Более того, если обозначить через $\bullet$ любую из операций сложения (+) и умножения (*), имеем :

$$(x_1 , x_2 , \dots , x_k) \bullet (y_1 , y_2 , \dots , y_k) = (x_1 \bullet y_1 , x_2 \bullet y_2 , \dots , x_k \bullet y_k)$$

где знак $\bullet$ в правой части равенства
обозначает соответствующую операцию в $\mathbf{Z}_{p_i^{e_i}}$, для $i = 1 , 2 , \dots , k$

Греко-китайская теорема об остатках

Греко-китайское представление чисел

Замечание. Разложение, рассмотренное в предыдущей теореме, записывается в виде :

$$\mathbf{Z}_m \cong \times_{1 \le i \le k} \mathbf{Z}_{p_i^{e_i}}$$

Это разложение колец индуцирует разложение групп их обратимых элементов :

$$U_m \cong \times_{1 \le i \le k} U_{p_i^{e_i}}$$

Замечание. Одним из применений двух выше рассмотренных теорем является **греко-китайское представление** чисел.
Произвольное целое положительное число x , такое, что :

$0 < x < M$, где $M = m_1 * m_2 * , \dots , * m_k$, $(m_i , m_j) = 1$ для $i \neq j$

однозначно представимо
своими наименьшими неотрицательными остатками по модулю m_i ,
Причём операции сложения и умножения выполняются покомпонентно.

Пример. Для $m_1 = 3$ и $m_2 = 5$ имеем два числа : $6 = (0 , 1)$ и $7 = (1 , 2)$
Сумма этих чисел равна: $6 + 7 = (0 + 1 (\mathrm{mod}\ 3) , 1 + 2 (\mathrm{mod}\ 5)) = (1 , 3)$,
т.е. числу 13

Греко-китайская теорема об остатках

Греко-китайский алгоритм GCRA 2

Вход:
a , m_1, b , m_2 , такие, что
$x \equiv a \pmod{m_1}$, $x \equiv b \pmod{m_2}$
m_1 , m_2 – короткие целые числа
такие, что $(m_1, m_2) = 1$,
$m_1 > 1$, $m_2 > 1$

Выход:
x – единственное наименьшее неотрицательное решение по модулю $m_1 * m_2$ системы сравнений :

$$x \equiv a \pmod{m_1}$$
$$x \equiv b \pmod{m_2}$$

[Если a > 0 , то ничто не меняется]
$x := \mathbf{MOD}(a, m_1)$ // **MOD** (a , b) – подпрограмма вычисления
// неотрицательного остатка от деления a на b

[Вычисление m^{-1}]
$m^{-1} := \mathbf{MODINV}(m_1, m_2)$ // **MODINV** (a , b) – подпрограмма вычисления
// наименьшего неотрицательного обратного элемента
// по модулю b к элементу a

[Вычисление q]
$q := \mathbf{MOD}((m^{-1}) * (b - x), m_2)$

[Выход]
Вернуть $x := x + m_1 * q$ // Т.к. $0 \le q < m_2$, то возвращаемое значение x
// удовлетворяет неравенствам $0 \le x < m_1 * m_2$

Греко-китайская теорема об остатках

Сложность алгоритма GCRA 2

Первые два шага : шаг **[Если a > 0 , то ничто не меняется]**
и шаг **[Вычисление m^{-1}]** алгоритма **GCRA 2**
(**GCRA 2** – **G**reek – **C**hinese *R*emainder **A**lgorithm with **2** congruences)
выполняются за время равное O (1) ,
т.к. для вычисления мультипликативного обратного элемента
используется расширенный алгоритм Евклида
и операции производятся над короткими целыми числами.

На следующем (третьем) шаге **[Вычисление q]** выполняются умножение и деление,
и, наконец, на четвёртом шаге **[Выход]** – только одно умножение.

Время выполнения каждой из указанных операций
доминируется временем вычисления произведения $m_1 * m_2$.

Таким образом, временная сложность алгоритма GCRA 2 равна :

$$T_{GCRA\ 2}(a, m_1, b, m_2) = O(L(m_1 * m_2))$$

Греко-китайская теорема об остатках

Обобщение алгоритма GCRA 2

Алгоритм **GCRA 2** предназначен для решения системы из двух уравнений.

В общем случае требуется решить систему из k уравнений :

$$x \equiv a_1 \pmod{m_1}$$
$$x \equiv a_2 \pmod{m_2}$$
$$\ldots\ldots\ldots\ldots\ldots$$
$$x \equiv a_k \pmod{m_k}$$

по попарно взаимно простым модулям, т.е. $(m_1 , m_2) = 1$ для $i \neq j$.

Идея решения состоит в том, чтобы использовать **GCRA 2** для последовательного решения пар уравнений.

Реализуя эту идею, на первом этапе мы получаем решение x_0 первых двух уравнений,
где x_0 – наименьшее неотрицательное решение по модулю $m_1 * m_2$.
На следующем этапе мы получаем решение по модулю $m_1 * m_2 * m_3$
для пары уравнений : $x \equiv x_0 \pmod{m_1 * m_2}$, $x \equiv a_3 \pmod{m_3}$ и так далее.

Греко-китайская теорема об остатках

Греко-китайский алгоритм GCRA k

Вход:
пары a_i , m_i , такие, что
$x \equiv a_i \pmod{m_i}$, $i = 1 , 2 , \dots , k$
каждое m_i – короткое целое число
$m_i > 1$, $(m_i , m_j) = 1$, для $i \neq j$

Выход:
x – единственное наименьшее неотрицательное решение по модулю $m_1 * m_2 * \dots * m_k$ системы k сравнений :
$x \equiv a_i \pmod{m_i}$, $i = 1 , 2 , \dots , k$

[Инициализация]
$m := 1$, $x :=$ **MOD** (a_1 , m_1) // смысл подпрограмм **MOD** (a , b) и **MODINV** (a , b)
// аналогичен алгоритму **GCRA 2**

[Применение в цикле GCRA 2]
Цикл Для $i = 1 .. k – 1$ **Выполнять**
$m := m * m_i$, $m^{-1} :=$ **MODINV** (m , m_{i+1}) ,
$q :=$ MOD $((m^{-1}) * (a_{i+1} – x) , m_{i+1})$, $x := x + m * q$
Конец Цикла

[Выход]
Вернуть x

Греко-китайская теорема об остатках

Сложность алгоритма GCRA k

Очевидно, что время работы алгоритма **GCRA k**
доминируется временем выполнения второго шага
[Применение в цикле GCRA 2] .

Если $M = m_1 * m_2 * , \ldots, * m_k$,
то i-е выполнение тела цикла требует времени порядка :

$$O (L (m_1 * m_2 * \ldots * m_i) * L (m_{i+1}))$$

Поэтому, в целом, цикл, состоящий из (k – 1) шагов, выполняется за время :

$$\leq L (M) * \Sigma L (m_i) \quad \text{для всех } 2 \leq i \leq k$$

Далее, учитывая, что функция L () ведёт себя как логарифм,
получаем новую оценку :

$$\leq L (M) * \Pi\, m_i \quad \text{для всех } 2 \leq i \leq k , \qquad \text{т.е.} \qquad \leq (L (M))^2$$

Таким образом, временная сложность алгоритма GCRA k равна :

$$T_{GCRA\,k} (a_i , m_i , i = 1 , 2 , \ldots , k) = O ((L (m_1 * m_2 * \ldots * m_k))^2)$$

План лекции: тема подраздела

- Целые числа по модулю m
- Греко-китайская теорема об остатках
- **Арифметика вычетов**

Арифметика вычетов

Вводные замечания

Замечание 1. Арифметика вычетов (остатков) – **АВ** является средством выполнения точных (т.е. выполняющихся без ошибок округления) арифметических операций над (длинными) целыми числами.

Замечание 2. Основная идея обеспечения точности вычислений состоит в использовании операций над вычетами (для представления которых требуется существенно меньшая разрядность) вместо операций над (длинными) целыми числами.

Замечание 3. В зависимости от отношения значений исходных целых чисел и допустимых (т.е. представимых в компьютере) значений модулей, с помощью которых формируются вычеты, применяется либо одномодульная, либо многомодульная арифметики.

Арифметика вычетов

Одномодульная АВ – обоснование

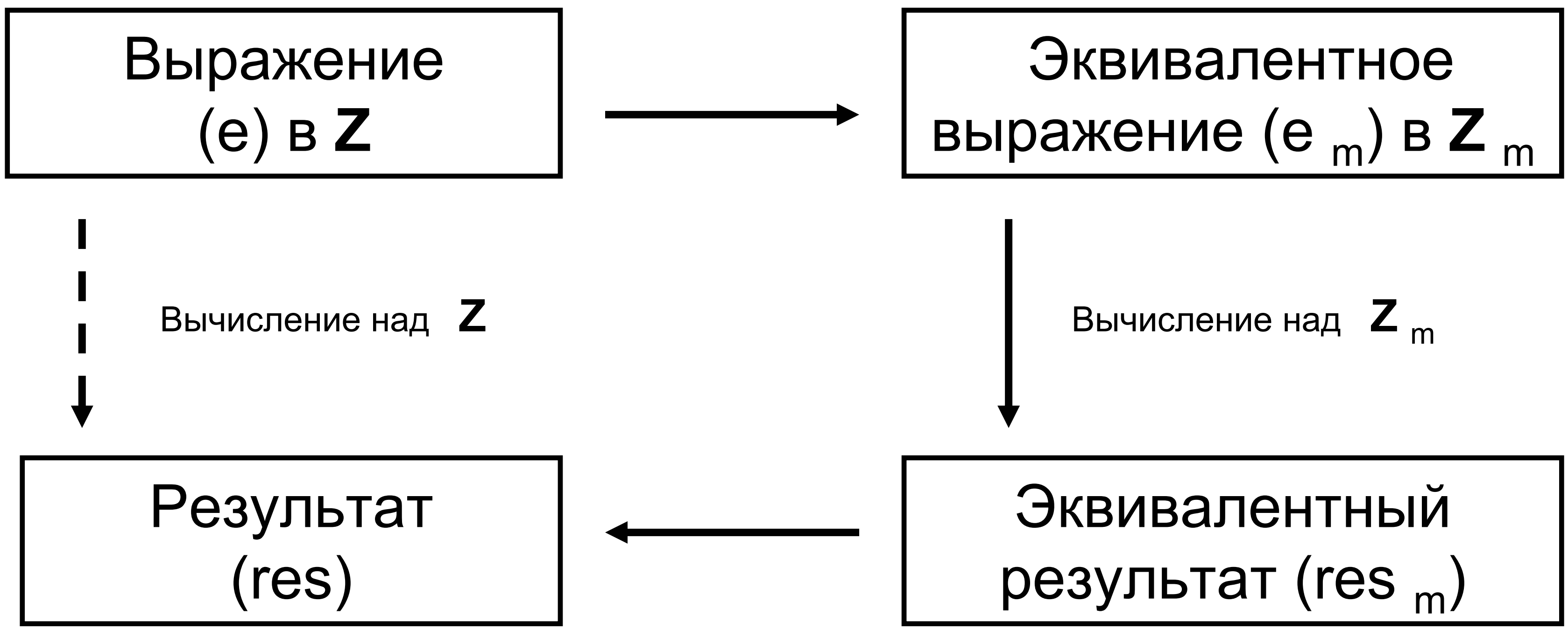
Пусть дано выражение $e(i_1, i_2, \dots, i_h)$ над $\mathbf{Z}$,
зависящее от целочисленных аргументов $i_1, i_2, \dots, i_h$,
которое нужно вычислить (оценить).

Тривиальный подход состоит в непосредственном вычислении выражения над $\mathbf{Z}$
Однако промежуточные результаты
могут не быть конечно представимыми целыми числами (например, 1/3 = 0.333 ….),
что приведёт к необходимости их аппроксимации (округления)
с возникновением ошибок.

Окольный подход структурирует решение исходной задачи на три этапа.
<u>На первом этапе</u> по выражению $e(i_1, i_2, \dots, i_h)$ над $\mathbf{Z}$
формируется эквивалентное выражение $e(i'_1, i'_2, \dots, i'_h)$ над $\mathbf{Z}_m$,
для некоторого m, где $i'_j \equiv i_j \pmod m$
(т.е. в иных обозначениях : $i'_j \equiv r_m(i_j)$), $j = 1, 2, \dots, h$
<u>На втором этапе</u> производится вычисление выражения e_m над $\mathbf{Z}_m$.
Таким образом получается результат res_m, где $res_m \equiv res \pmod m = r_m(res)$
<u>На третьем этапе</u> выполняется обратное отображение res_m в множество целых чисел $\mathbf{Z}$.

Арифметика вычетов

Одномодульная АВ – схема вычислений



Арифметика вычетов

Одномодульная АВ – однозначность

Отношение эквивалентности $res \equiv res_m \pmod{m}$
не определяет однозначно окончательный результат вычисления.

Для того, чтобы определить res однозначно,
нужно иметь априорную оценку его величины.
Эта оценка используется в качестве модуля m
и все операции выполняются в кольце $\mathbf{Z}_m$.

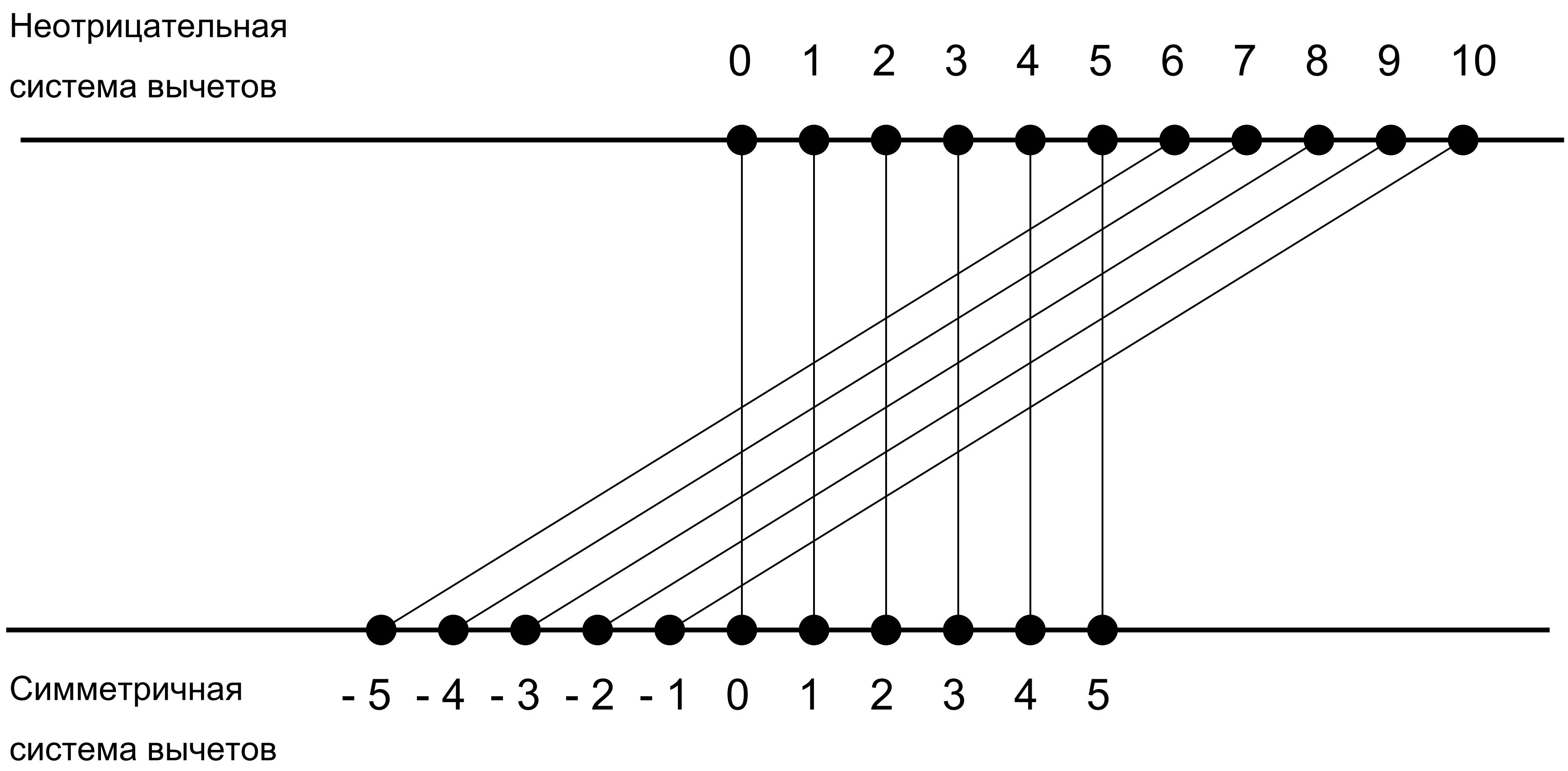
Если мы имеем оценку величины res ,
то мы ищем наименьшее неотрицательное решение уравнения $res \equiv res_m \pmod{m}$

Если мы имеем оценку не величины res , а только величины $| res |$,
то мы ищем наименьшее по абсолютной величине решение.

Замечание. Для вычисления как положительных, так и отрицательных
значений выражения можно использовать симметричную систему вычетов
$\mathbf{Z}_p = \{ -(p-1)/2 , \dots , -2 , -1 , 0 , 1 , 2 , \dots , (p-1)/2 \}$,
которая изоморфна системе неотрицательных вычетов $\mathbf{Z}_{p+} = \{ 0 , 1 , 2 , \dots , p-1 \}$.
Однако обычно все операции **АВ** производятся в $\mathbf{Z}_{p+}$ ($\mathbf{Z}_p$ обеспечивает только интерфейс с $\mathbf{Z}$)

Арифметика вычетов

Изоморфизм одномодульных вычетов



Арифметика вычетов

Одномодульная АВ – операция деления

Как уже отмечено ранее, кольцо ($\mathbf{Z}_p$, + , *) является конечным полем, если p – простое число.
(Такое конечное поле является полем Галуа и обозначается GF (p)).

В конечном поле выполнимы все арифметические операции: сложение, вычитание, умножение и деление.

Т.к. в конечном поле обратный элемент к любому ненулевому элементу всегда существует, деление по модулю p определяется следующим образом :

$$a / b \ (\mathrm{mod}\ p) = a * (b^{-1} (\mathrm{mod}\ p)) \ (\mathrm{mod}\ p)$$

где b^{-1} (mod p) – это мультипликативный обратный элемент к элементу b по модулю p .
Частное двух целых чисел a и b в GF (p) также является целым числом, даже если b не делит a в $\mathbf{Z}$.

Пусть модуль p ограничивает по величине окончательный результат res .
Возможны два случая :

(1) Если res $\notin$ GF (p) , то $res_p \neq res$ и для вычисления res требуется некоторая дополнительная информация.

(2) Если res $\in$ GF (p) , то $res_p = res$

Арифметика вычетов

Одномодульная АВ – пример вычисления

Пример. Выполним точные арифметические операции в GF (11).
Чтобы вычислить значение $x = 1 / 3 – 4 / 3$
воспользуемся априорной информацией о том,
что мы ищем результат в симметричной системе вычетов.

$x \ (\mathrm{mod}\ 11) \equiv (1/3 + (-4)/3) \ (\mathrm{mod}\ 11)$

$\equiv (1/3 + 7/3) \ (\mathrm{mod}\ 11)$ ----- в неотрицательной системе вычетов

$\equiv (1 * 3^{-1} + 7 * 3^{-1}) \ (\mathrm{mod}\ 11)$ ----- 3^{-1} является мультипликативным обратным элементом к $3 \ (\mathrm{mod}\ 11)$

$\equiv (1 * 4 + 7 * 4) \ (\mathrm{mod}\ 11)$

$\equiv 32 \ (\mathrm{mod}\ 11)$

Отображая результат обратно в симметричную систему вычетов,
получим правильный ответ $x = -1$

Замечание. В приведённом выше примере $res_p = res$,
т.к. априорно известно, что $res \in GF(p)$
и res принадлежит симметричному множеству.

Арифметика вычетов

Одномодульная АВ – контр – пример

Пример. В GF (11) вычислим значение $x = 1 / 2 - 1 / 3$
воспользовавшись той же априорной информацией, что в предыдущем примере.

$x \ (\mathrm{mod}\ 11) \equiv (1 / 2 + (-2) / 3) \ (\mathrm{mod}\ 11)$
$\equiv (1 / 2 + 9 / 3) \ (\mathrm{mod}\ 11)$ ----- в неотрицательной системе вычетов
$\equiv (1 * 2^{-1} + 9 * 3^{-1}) \ (\mathrm{mod}\ 11)$ ----- 2^{-1} и 3^{-1} являются мультипликативными обратными элементами к $2 \ (\mathrm{mod}\ 11)$ и $3 \ (\mathrm{mod}\ 11)$ соответственно
$\equiv (1 * 6 + 9 * 4) \ (\mathrm{mod}\ 11)$
$\equiv 42 \ (\mathrm{mod}\ 11)$
$\equiv 9$

Если теперь отобразить результат обратно в симметричное множество,
то полученный ответ будет неправильным $x = -2$.
Поэтому требуется дополнительная информация.

Достаточно знать, что мы ищем рациональное число $x \ (\mathrm{mod}\ 11) = (a / b) \ (\mathrm{mod}\ 11)$,
где $b = 6$ (т.е. НОК знаменателей двух дробей).
Тогда $a = (x \ (\mathrm{mod}\ 11) * (b \ (\mathrm{mod}\ 11)) = 9 * 6 \ (\mathrm{mod}\ 11) = 10 \ (\mathrm{mod}\ 11) = -1$
(отображение на симметричное множество). Следовательно, $x = (-1) / 6$. Ответ правильный.

Арифметика вычетов

Многомодульная АВ – обоснование

Замечание. Для однозначного определения результата res по его остатку res_m ($m > res$) необходимо, чтобы значение модуля m было достаточно большим. Однако значение m ограничено размером компьютерного слова, поэтому очевидным решением этой проблемы является переход от одномодульной к многомодульной **АВ**.

Вычисления с помощью многомодульной **АВ** производятся по той же трёхэтапной схеме, что и вычисления с помощью одномодульной **АВ**.

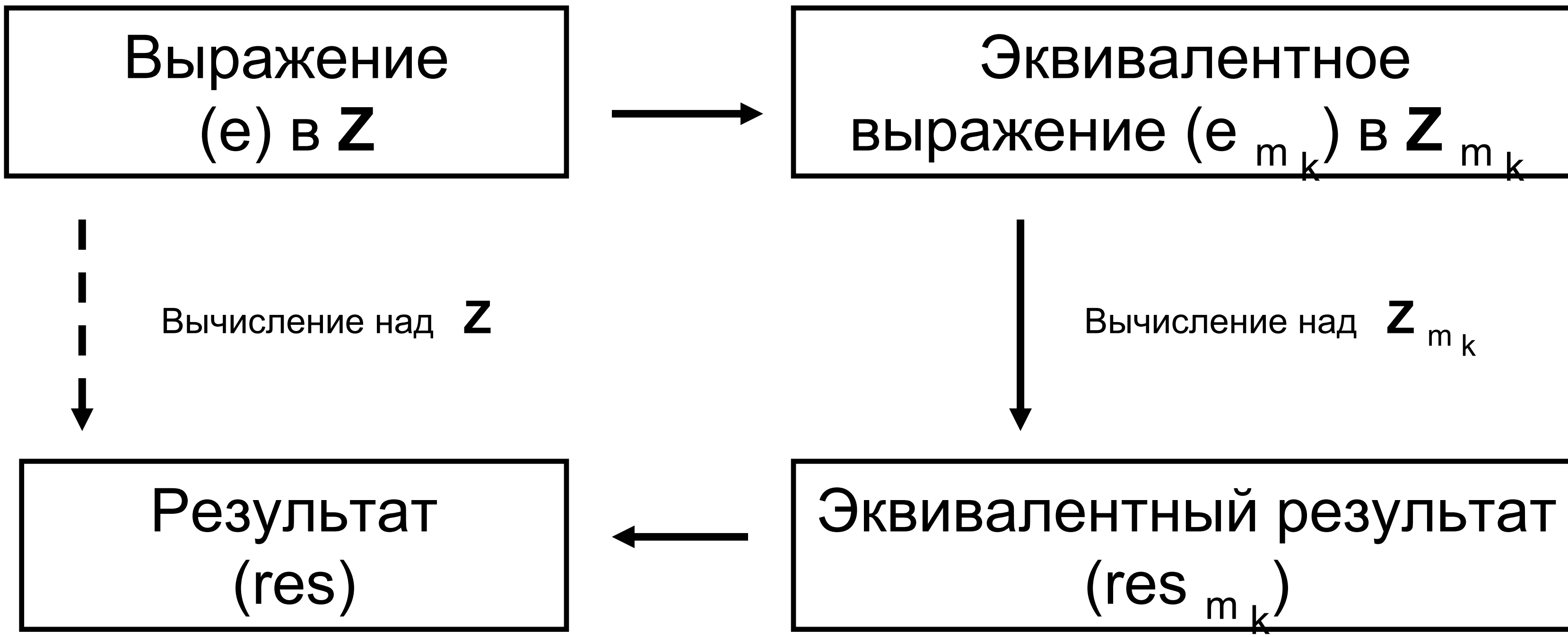
<u>На первом этапе</u> для заданного выражения $e(i_1, i_2, \dots, i_h)$, зависящего от целочисленных аргументов $i_1, i_2, \dots, i_h$, вычисляются эквивалентные ему выражения $e_{m_k}(i_{1_k}, i_{2_k}, \dots, i_{h_k})$, где $i_{j_k} = r_{m_k}(i_j)$, $j = 1, 2, \dots, h$, $k = 1, 2, \dots, n$ для коротких модулей m_k.

<u>На втором этапе</u> мы вычисляем выражения e_{m_k}, если они определены в $\mathbf{Z}_{m_k}$, и получаем эквивалентные результаты res_{m_k}, $k = 1, 2, \dots, n$

<u>На третьем этапе</u>, пользуясь греко-китайским алгоритмом (либо специальными таблицами), получаем окончательный результат res.

Арифметика вычетов

Многомодульная АВ – схема вычислений



Арифметика вычетов

Многомодульная АВ – однозначность

Замечание. Проблема однозначности восстановленного результата res в $\mathbf{Z}_m$ из результатов res_{m_k} в $\mathbf{Z}_{m_k}$ в многомодульной **АВ** имеет ту же природу, что и в одномодульной **АВ**.

Модули m_k должны выбираться таким образом,
чтобы $m_1 * m_2 * \ldots * m_n > res$

Если дана оценка на res ,
то мы ищем наименьшее неотрицательное решение
греко-китайской задачи об остатках.

Если оценивается $|res|$,
то ищется наименьшее по абсолютной величине решение.

Арифметика вычетов

Стандартный набор остатков

Общеизвестные системы счисления (например, десятичная) являются линейными, позиционными и весовыми.
Это означает, что всем позициям соответствуют веса, зависящие от одного основания.

Вместо этого многомодульная система счисления использует взаимно простые позиционные основания, которые будучи определённым образом упорядочены образуют **вектор оснований**.

Остатки, формируемые при представлении целого числа в многомодульной **АВ**
и упорядоченные в соответствии со структурой вектора оснований, называются **стандартным набором остатков** относительно данного вектора оснований.

Замечание. В многомодульной **АВ** (по аналогии с одномодульной **АВ**) мы можем для данного вектора оснований определить :

- либо наименьшую неотрицательную числовую систему;
- либо (при условии, что все модули нечётные) наименьшую по абсолютной величине числовую систему или симметричную систему остатков.

Арифметика вычетов

Изоморфизм модульных арифметик

Рассмотрим вектор оснований общего вида :

$\beta = [\, m_1 , m_2 , \ldots , m_n \,]$, $(m_i , m_j) = 1$ для $i \neq j$

Пусть $M = m_1 * m_2 * \ldots * m_n$
(Т.к. модули попарно взаимно просты, то M является их НОК)

Теорема. Два целых числа n_1 и n_2 имеют одинаковые стандартные наборы остатков относительно вектора оснований $\beta = [\, m_1 , m_2 , \ldots , m_n \,]$ тогда и только тогда, когда $n_1 \equiv n_2 \ (\mathrm{mod}\ m_1 * m_2 * \ldots * m_n)$

Из теоремы следует, что множество $\mathbf{Z}_\beta = \{ n \ (\mathrm{mod}\ \beta) : n \in \mathbf{Z} \}$ содержит M элементов, которые взаимно однозначно отображаются на элементы множества $\mathbf{Z}_M$.

Нетрудно показать, что два множества $\mathbf{Z}_\beta$ и $\mathbf{Z}_M$
с соответствующими операциями сложения и умножения
представляют собой **изоморфные** конечные коммутативные кольца,
т.е. многомодульная арифметика эквивалентна арифметике по модулю M .

Арифметика вычетов

Многомодульная АВ – реализация

Важное преимущество многомодульной числовой системы
(в дополнение к её способности обеспечивать точные вычисления)
состоит в отсутствии переносов при выполнении операций сложения и умножения.

Арифметика замкнута в каждой позиции
(т.е. арифметические действия выполняются полностью и независимо в разных позициях).
Поэтому можно выполнять сложение и умножение длинных целых чисел
так же быстро, как и обычных (коротких) целых чисел.

При реализации многомодульной **АВ** на двоичных компьютерах
удобно использовать модули следующего вида : $m = 2^{e} - 1$
т.е. каждый модуль на единицу меньше, чем степень двойки.

В некоторых случаях необходимо знать, являются ли модули взаимно простыми.
Если они имеют вид $2^{e} - 1$, то для проверки этого можно использовать следующее правило :

$$(2^{e} - 1 , 2^{f} - 1) = 2^{(e,f)} - 1$$

Из указанного правила следует, что модули взаимно просты тогда и только тогда,
когда числа e и f взаимно просты.
Замечание. Правило следует из алгоритма Евклида и следующего тождества :

$$(2^{e} - 1) (\mathrm{mod}\ (2^{f} - 1)) = 2^{e\ (\mathrm{mod}\ f)} - 1$$

Арифметика вычетов

Многомодульная АВ – сложность

Время, которое требуется для выполнения
операций сложения, вычитания и умножения двух n – значных чисел
при использовании многомодульной **АВ** равно :

$$O\,(\,L\,(\,n\,)\,)$$

(без учёта времени преобразования к модульному представлению и обратно)

Очевидно, что для сложения и вычитания
никакого роста эффективности при этом не достигается,
однако для умножения получается существенное улучшение
по сравнению с обычными методами, требующими времени $O\,(\,(\,L\,(\,n\,)\,)^2\,)$.

Замечание. Позиционная независимость выполнения арифметических операций
в многомодульной **АВ** указывает на возможность её эффективной реализации
в компьютерах, имеющих разрядно – параллельную модель вычислений.

Арифметика вычетов

Многомодульная АВ – операция деления

Для выполнения операции деления в многомодульной АВ
определим элемент $b^{-1} \pmod{\beta}$
мультипликативно обратный к элементу $b = [\, b_1 , b_2 , \ldots , b_n \,]$
по модулю вектора оснований $\beta = [\, m_1 , m_2 , \ldots , m_n \,]$
следующим образом :

$$b^{-1} \pmod{\beta} = [\, (b_1)^{-1} \pmod{m_1} , (b_2)^{-1} \pmod{m_2} , \ldots , (b_n)^{-1} \pmod{m_n} \,]$$

Далее, если $a = [\, a_1 , a_2 , \ldots , a_n \,]$, то :

$$\begin{aligned} a / b \pmod{\beta} = [\, & a_1 * (b_1)^{-1} \pmod{m_1} , \\ & a_2 * (b_2)^{-1} \pmod{m_2} , \\ & \ldots\ldots\ldots\ldots\ldots\ldots \\ & a_n * (b_n)^{-1} \pmod{m_n} \,] \end{aligned}$$

Безусловно, как и в случае одного модуля, если b не делит a ,
то результат не может быть получен без дополнительной информации
(однако он допустим в качестве промежуточного результата)

Арифметика вычетов

Многомодульная АВ – сравнение чисел

Основная трудность при работе с многомодульными числовыми системами – это выполнение **операции сравнения величин** целых чисел.

Замечание. Безусловно, используя симметричную систему остатков, можно вычесть из одного числа другое и затем определить знак разности. Но остатки в симметричной системе не несут информации о знаке числа, поэтому для определения знака потребуется преобразование к обычному (не модульному) виду, что, по сути, означает отказ от многомодульной **АВ**.

Задача определения знака числа может быть эффективно решена с помощью преобразования числа x к представлению **со смешанными основаниями**. Важно, что при этом выполняются только операции многомодульной арифметики.

Замечание. Представление со смешанными основаниями ранее было рассмотрено при выражении решения x в греко-китайском алгоритме :

$$x = q_1 + q_2 * m_1 + q_3 * m_1 * m_2 + \ldots + q_n * m_1 * m_2 * \ldots * m_{n-1} \qquad (*)$$

где каждое q_i не превосходит модуля m_i, q_n называется **старшим членом** числа x, знак числа x совпадает со знаком его старшего члена.

Арифметика вычетов

Многомодульная АВ – знак числа (часть 1)

Замечание. Для определения знака числа удобно,
чтобы последний модуль в векторе оснований был равен 2 ,
т.к. необходимо знать,
в какой половине множества возможных чисел располагается результат.

Пусть дано представление :

$x = [\, a_1 , a_2 , \ldots , a_n \,]$

относительно вектора оснований :

$\beta = [\, m_1 , m_2 , \ldots , m_n \,]$

Как вычислить знак числа x ?

Для определения знака числа x необходимо преобразовать это число к форме со смешанными основаниями и определить знак старшего члена.

Для этого необходимо вычислить цифры $q_1 , q_2 , \ldots , q_n$

Арифметика вычетов

Многомодульная АВ – знак числа (часть 2)

Очевидно, что из формулы (*) следует :

$x \equiv q_1 \pmod{m_1}$, т.е. $q_1 = a_1$ (тем самым получена 1-я цифра)

Далее вычислим разность $x - q_1$
(вычитая q_1 из каждого остатка, представляющего x).
Имеем :

$x - q_1 = q_2 * m_1 + q_3 * m_1 * m_2 + \ldots + q_n * m_1 * m_2 * \ldots * m_{n-1}$

Первая цифра (в смешанном представлении) числа $x - q_1$
равна нулю, поэтому первые цифры всех последующих чисел можно будет не рассматривать.

Таким образом, будем считать, что размерность вектора $x - q_1$ равна $n - 1$

Теперь найдём $(m_1)^{-1} (\bmod\ \beta_r)$
(многомодульный) мультипликативный обратный к элементу m_1 по модулю β_r элемент ,
где $\beta_r = [m_2 , \ldots , 2]$ (имеет размерность $n - 1$)

Далее вычислим (многомодульное) произведение $(x - q_1) * (m_1)^{-1}$
чтобы получить вторую цифру q_2 .

Будем продолжать этот процесс, пока не вычислим q_n . При $q_n = 0$ $x > 0$, при $q_n = 1$ $x < 0$.

Спасибо за внимание !

Вопросы ?