Дискретная математика. Основные алгоритмы

П. В. Трифонов

8 февраля 2006 г.

Оглавление

Глава 1

Элементы теории множеств

1.1 Множества и операции над ними

1.1.1 Основные понятия

Понятие *множества*, как и другие исходные понятия математической теории, не определяется. Множество A называется *подмножеством* множества B, если всякий элемент A содержится в B. Множества равны, если $A \subset B$ и $B \subset A$. Если $A \subset B$ и $A \neq B$, то A называют собственным, строгим или истинным подмножеством B. Множества могут быть конечные и бесконечные. Число элементов множества (мощность) обозначается как $|A|$. Множество, не содержащее ни одного элемента, называется пустым $\emptyset$. Пустое множество является подмножеством любого множества. Множество может быть задано:

1. Списком элементов, например $\{a, b, c, d\}$. Задание типа $N = 1, 2, 3, \ldots$ — это не список, а условное обозначение, допустимое только тогда, когда не может возникнуть разночтений.

2. Порождающей процедурой, задающей способ получения элементов одного множества из элементов другого множества. Например, $A = \{2^k | k \in \mathbb{N}\}$.

3. Характеристическим предикатом, т.е. описанием свойств, которыми должны обладать элементы множества, например $A = \{x | \sin(x) = 0\}$.

Задание множества характеристическим предикатом иногда приводит к проблемам. Например, рассмотрим множество всех множеств, не содержащих себя в качестве элемента:

$$Y = \{X | X \notin X\}$$

Если множество Y существует, должен быть ответ на вопрос: $Y \in Y$?. Пусть $Y \in Y \Rightarrow Y \notin Y$. С другой стороны, пусть $Y \notin Y \Rightarrow Y \in Y$. Возникло неустранимое логическое противоречие (парадокс Рассела). Возможные способы ее решения:

1. Ограничить используемые характеристические предикаты видом

$$P(x) = (x \in A) \wedge Q(x),$$

где A — известное, заведомо существующее множество (универсум). Это записывают как $B = \{x \in A | Q(x)\}$.

2. Теория типов. Объекты имеют тип 0, их множества — тип 1, множества множеств — тип 2 и т.д.

3. Характеристический предикат задан в виде вычислимой функции (алгоритма).

Для всякого множества A можно построить множество всех его подмножеств (булеан) $2^A = \{X | X \subset A\}$, причем $|2^A| = 2^{|A|}$

1.1.2 Операции над множествами

Пусть задан некоторый универсум U, множества $A, B, C \subset U$.

1. Объединение

$$A \cup B = \{x | x \in A \vee x \in B\}$$

2. Пересечение

$$A \cap B = \{x | x \in A \wedge x \in B\}$$

3. Разность

$$A \setminus B = \{x | x \in A \wedge x \notin B\}$$

4. Симметрическая разность

$$A \bigwedge B = (A \setminus B) \cup (B \setminus A)$$

5. Дополнение

$$\bar{A} = \{x | x \notin A\}$$

Свойства операций:

1. Идемпотентность

$$A \cup A = A, A \cap A = A$$

2. Коммутативность

$$A \cup B = B \cup A, A \cap B = B \cap A$$

3. Ассоциативность

$$(A \cup B) \cup C = A \cup (B \cup C), (A \cap B) \cap C = A \cap (B \cap C)$$

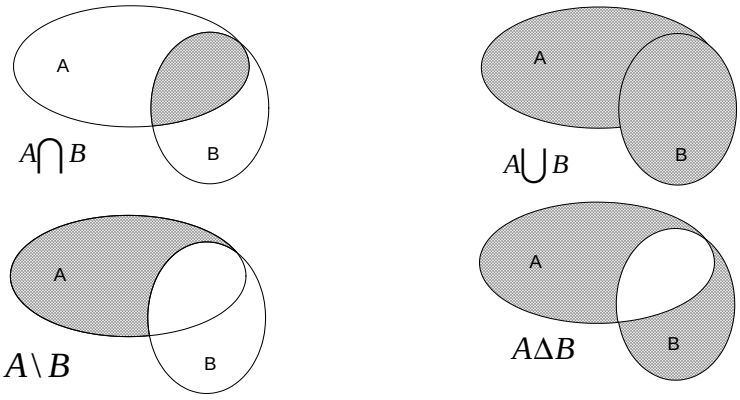


Рис. 1.1: Диаграммы Эйлера, иллюстрирующие операции над множествами

4. Дистрибутивность

$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C), A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$

5. Поглощение

$$(A \cap B) \cup A = A, (A \cup B) \cap A = A$$

6. Свойства нуля

$$A \cup \emptyset = A, A \cap \emptyset = \emptyset$$

7. Свойства единицы

$$A \cup U = U, A \cap U = A$$

8. Инволютивность

$$\bar{\bar{A}} = A$$

9. Законы де Моргана

$$\overline{A \cap B} = \bar{A} \cup \bar{B}, \overline{A \cup B} = \bar{A} \cap \bar{B}$$

10. Свойство дополнения

$$A \cup \bar{A} = U, A \cap \bar{A} = \emptyset$$

11. Выражение для разности

$$A \setminus B = A \cap \bar{B}$$

12. Выражение для симметрической разности

$$A \bigtriangleup B = (A \cup B) \setminus (A \cap B)$$

13. Ассоциативность симметрической разности

$$(A \bigtriangleup B) \bigtriangleup C = A \bigtriangleup (B \bigtriangleup C)$$

14. Коммутативность симметрической разности

$$A \bigtriangleup B = B \bigtriangleup A$$

15.

$$A \cap (B \bigtriangleup C) = (A \cap B) \bigtriangleup (A \cap C)$$

Для доказательства подобных тождеств могут быть использованы два метода:

- Метод двух включений ($X = Y \Leftrightarrow (X \subset Y) \wedge (Y \subset X)$).
- Метод эквивалентных преобразований с использованием уже доказанных тождеств.

1.1.3 Представление множеств в ЭВМ

Если универсум достаточно велик (или бесконечен), а используемые подмножества имеют малую мощность по сравнению с ним, целесообразно использовать представление множества в виде упорядоченного списка.

Если универсум конечен и сравнительно мал, то множество может быть представлено битовой маской, в которой 1 обозначает вхождение соответствующего элемента в множество. В этом случае основные операции могут быть эффективно выполнены с помощью простейших операторов.

```
//пусть универсум - множество букв латинского алфавита - 26 элементов
unsigned A,B;
A=0x00000007;//{a,b,c}
B=0x000000fe;//{b,c,d,e,f,g,h}
unsigned C=A&B;//пересечение
unsigned D=A|B;//объединение
unsigned E=~A;//дополнение. 6 последних бит игнорируем
```

Перебор всех подмножеств универсума может быть организован путем перебора соответствующих целых чисел.

1.1.4 Коды Грея

В некоторых случаях целесообразно перебирать подмножества в таком порядке, что каждое очередное подмножество получается их предыдущего добавлением или удалением ровно одного элемента. Этому соответствует упорядочение бинарных векторов в соответствии с кодом Грея. Пусть $(B_0, B_1, B_2, \ldots, B_{2^k-1})$ — последовательность

целых чисел (бинарных векторов) длины 2^k, такая что B_i и B_{i+1} различаются ровно в одной позиции (*код Грея*). Тогда код Грея длины 2^{k+1} может быть получена как $(0B_0, 0B_1, 0B_2, \ldots, 0B_{2^k-1}, 1B_{2^k-1}, \ldots, 1B_2, 1B_1 1B_0)$ (двоично отраженный код Грея) [13]. Для порождения кода Грея может быть использован следующий алгоритм:

```
GRAY(k)
 1  boolB[k];
 2  for i ← 0 to k − 1
 3      do B[i] = 0;
 4  PRINTB;
 5  for i ← 1 to 2^k − 1
 6      do p = Q(i);
 7          B[p] = 1 − B[p];
 8          PRINTB;
 9
```

```
Q(i)
 1  q := 0;
 2  while (i mod 2 = 0)
 3      do i/ = 2;
 4          q + +;
 5  return q;
```

Лемма 1.1.

$$\forall m : 0 \leq m \leq 2^k - 1 : Q(2^k + m) = Q(2^k - m)$$

Доказательство. □

Теорема 1.1. *Процедура $GRAY(k)$ корректно строит двоичный отраженный код Грея.*

Доказательство. Ясно, что для $k = 1$ этот алгоритм строит правильный код Грея. Предположим, что он строит его для $k > 1$ и рассмотрим его работу для $k' = k + 1$. Согласно индукционному предположению, первые 2^k построенных значений являются кодом Грея. На шаге 2^k (нумерация с 0!) k-й бит поменяет свое значение, после чего функция $Q(i)$ возвратит те же значения, что и на предыдущих итерациях, но в обратном порядке, что соответствует определению двоичного кода Грея. □

Такая же последовательность может быть получена как $(B_0 0, B_0 1, B_1 1, B_1 0, B_2 0, \ldots)$. Пусть $F_k = f_1, f_2, \ldots, f_{2^k-1}$ — последовательность разрядов, в которых различаются смежные элементы кода Грея. Видно, что

$$F_{k+1} = 0, f_1 + 1, 0, f_2 + 1, 0, f_3 + 1, \ldots, f_{2^k-1} + 1, 0.$$

Заметим, что последовательность целых чисел в двоичном представлении может быть рекурсивно определена как $(T_0 0, T_0 1, T_1 0, T_1 1, \ldots, T_{2^k-1} 0, T_{2^k-1} 1)$. Пусть

$C_k = (c_1, c_2, \ldots, c_{2^k-1})$ — последовательность, такая что c_i — номер разряда, в котором перенос прекращается, когда к T_{i-1} добавляется 1. Из рекурсивного определения двоичных чисел видно, что

$$C_{k+1} = 0, c_1 + 1, 0, c_2 + 1, 0, c_3 + 1 \ldots, c_{2^k-1} + 1, 0$$

Видно, что $\forall k : C_k = F_k$, т.е. различающиеся позиции в коде Грея соответствуют тому разряду, в котором прекращается перенос в двоичном коде. Пусть $i = \sum_{j=0}^{k-1} i_j 2^j, i_j \in \{0, 1\}$, т.е. $i \leftrightarrow (i_{k-1}, \ldots, i_0)_2$. Аналогично, $B_i \leftrightarrow (b_{k-1}^{(i)}, \ldots, b_0^{(i)})$. Можно заметить, что $b_j^{(i)} = i_j + i_{j+1} \bmod 2$, откуда получается способ непосредственного порождения элементов кода Грея по их индексу, т.е. $B_i = i \hat{} (i >> 1)$.

Упражнения

1. Доказать, что $|A \cup B| = |A| + |B| - |A \cap B|$.
2. Доказать, что $A \cup B = (A \cap B) \cup (A \cap \overline{B}) \cup (\overline{A} \cap B$
3. Доказать, что $Q(2^k + m) = Q(2^k - m), 0 \leq m \leq 2^k - 1$, где Q — функция из п. 1.1.4
4. Доказать свойства операций над множествами.

1.2 Отношения на множествах

1.2.1 Декартово произведение

Определение 1.1. Пусть A, B — произвольные множества. *Неупорядоченной парой* на множествах A, B называется любое множество $\{a, b\} : (a \in A \wedge b \in B) \vee (a \in B \wedge b \in A)$.

Если множества A и B совпадают, то говорят о неупорядоченной паре на множества A. НУП $\{a, b\}$ равна НУП $\{c, d\}$, если $(a = c) \wedge (b = d) \vee (a = d) \wedge (b = c)$. Если $a = b$, НУП вырождается в одноэлементное множество.

Определение 1.2. *Упорядоченной парой* на множествах A, B называется набор из двух элементов $(a, b) : a \in A, b \in B$, причем порядок их записи имеет значение.

Пример: координаты точки на плоскости. Упорядоченная пара (a, b) равна УП (a', b'), если $(a = a') \wedge (b = b')$. Аналогично можно ввести понятие упорядоченного n-набора (кортежа, n-ки) $(a_1, \ldots, a_n) : a_i \in A_i$. n называется длиной (размерностью) кортежа, а a_i — его i-й проекцией.

Определение 1.3. Множество всех кортежей длины n на множествах $A_1, \ldots, A_n$ называют декартовым (прямым) произведением множеств $A_1, \ldots, A_n$ и обозначают $A_1 \times A_2 \times \ldots \times A_n$, т.е.

$$A_1 \times A_2 \times \ldots \times A_n = \{(a_1, \ldots, a_n) | a_i \in A_i\}.$$

Если все множества равны между собой, то их декартово произведение называют n-й декартовой степенью множества A и обозначают A^n. $A^1 = A$. Удобно также предположить, что $A \times (B \times C) = (A \times B) \times C$. Свойства декартова произведения:

1. $A \times (B \cup C) = (A \times B) \cup (A \times C)$.
2. $A \times (B \cap C) = (A \times B) \cap (A \times C)$.
3. $\emptyset \times A = A \times \emptyset = \emptyset$.

1.2.2 Соответствия и отношения

Отображение $f : A \to B$ из множества A в множество B считается заданным, если всякому $x \in A$ сопоставлен единственный элемент $y \in B$, называемый f-образом x и обозначаемый $f(x)$. Следовательно, каждое отображение определяется множеством $\{(x, y)|x \in A, y = f(x)\} \subset A \times B$, называемым графиком отображения. Отображение называется тождественным, если $\forall x \in A : f(x) = x$. Множество *всех* элементов $x \in A : f(x) = y_0, y_0 \in B$ называют f-прообразом элемента y_0. Множество всех $y \in B : \exists x \in A : f(x) = y$ называют *областью значений* отображения f.

Отображение $f : A \to B$ называется *инъективным*, если каждый элемент из области его значений имеет единственный прообраз, т.е. $f(x_1) = f(x_2) \Rightarrow x_1 = x_2$. Отображение $f : A \to B$ называется *сюръективным*, если его область значений совпадает со всем множеством B. Сюръективное отображение из A в B называют также *отображением* множества A *на* множество B. Отображение $f : A \to B$ называется *биективным*, если оно одновременно инъективно и сюръективно, т.е. каждому элементу A соответствует элемент B и наоборот, т.е. A и B находятся во взаимно однозначном соответствии. Биекция A на себя называют автоморфизмом (или подстановкой) A.

Понятие отображения можно обобщить. Если образ определен не для каждого элемента A, а только для элементов из некоторого его подмножества C, имеет место *частичное отображение*, а C называется *областью определения* данного частичного отображения.

Можно также допустить, что данному элементу $x \in A$ соответствует несколько элементов B. В этом случае говорят, что задано *соответствие* из A в B Областью определения соответствия $\rho \subset A \times B$ называют множество всех первых компонент упорядоченных пар

$$D(\rho) = \{x \in A|(\exists y \in B) : (x, y) \in \rho\}.$$

Областью значений соответствия называют множество вторых компонент

$$R(\rho) = \{y \in B|(\exists x \in A) : (x, y) \in \rho\}.$$

Соответствие называется всюду определенным, если $D(\rho) = A$. *Сечением* соответствия $\rho \in A \times B$ для фиксированного элемента x называется

$$\rho(x) = \{y \in B|(x, y) \in \rho\}.$$

Сечением соответствия по множеству C называется

$$\rho(C) = \{y \in B | (x, y) \in \rho, x \in C\}.$$

Соответствие $\rho \subset A \times B$ называют функциональным по второй или первой компоненте, если

$$(\forall (x, y) \in \rho, (x', y') \in \rho)(x = x') \Rightarrow (y = y')$$

или

$$(\forall (x, y) \in \rho, (x', y') \in \rho)(y = y') \Rightarrow (x = x')$$

соответственно. Определение расширяется на случай n-арных соответствий.

Соответствие $\rho \subset A \times A = A^2$ называется *бинарным отношением* на множестве A. Аналогично можно определить n-арное отношение как некоторое подмножество A^n. Если $x, y \in A : (x, y) \in \rho$, то они связаны бинарным отношением ρ, что записывают как $x \rho y$. Можно также определить бинарное отношение на множествах A, B как $\rho \subset A \times B$.

Бинарное отношение на A, состоящее из всех пар $(x, x) \in A^2$, называют *диагональю множества* A и обозначают как I или id_A.

Способы задания бинарных отношений:

1. С помощью некоторого предиката $f(x, y)$: $\rho = \{(x, y) \in A^2 | f(x, y)\}$.
2. Перечислением всех пар $\rho = \{(a, b), (c, d), (e, f)\}$.
3. Матрицей $C^\rho = ||c_{ij}^\rho|| : c_{ij}^\rho = \begin{cases} 1, & (a_i, b_j) \in \rho \\ 0, & (a_i, b_j) \notin \rho \end{cases}$. Удобно для вычислений.
4. Графом, узлами которого являются элементы множества, а дуга от x к y присутствует тогда и только тогда, когда $(x, y) \in \rho$.

1.2.3 Операции над бинарными отношениями

Здесь и далее рассматриваются бинарные отношения на множестве A, если не указано иное.

Сохраняются все операции, определенные для множеств. При этом универсумом является все декартово произведение $A \times B$. Кроме того, вводится операция композиции бинарных отношений $\rho \subset A \times B$ и $\sigma \subset B \times C$

$$\rho \circ \sigma = \{(x, z) \in A \times C | \exists y \in B : (x, y) \in \rho \wedge (y, z) \in \sigma\}.$$

n-й *степенью отношения* называется его n-кратная композиция с собой, т.е. $\rho^0 = I, \rho^1 = \rho, \rho^2 = \rho \circ \rho, \rho^n = \rho^{n-1} \circ \rho$. *Обратным отношением* называется

$$\rho^{-1} = \{(x, y) \in A^2 | (y, x) \in \rho\}.$$

Для матричной формы задания:

$$\begin{aligned} c_{ij}^{\rho\cup\sigma} &= c_{ij}^{\rho} \vee c_{ij}^{\sigma} \\ c_{ij}^{\rho\cap\sigma} &= c_{ij}^{\rho} \wedge c_{ij}^{\sigma} \\ c_{ij}^{\bar{\rho}} &= \neg c_{ij}^{\rho} \\ c_{ij}^{\rho\circ\sigma} &= \bigvee_{k=1}^{|B|} \left(c_{ik}^{\rho} \wedge c_{kj}^{\sigma}\right) \\ C^{\rho^{-1}} &= (C^{\rho})^{T} \end{aligned}$$

Свойства композиции бинарных отношений:

$$\rho \circ (\sigma \circ \tau) = (\rho \circ \sigma) \circ \tau \tag{1.1}$$

$$\rho \circ \emptyset = \emptyset \circ \rho = \emptyset \tag{1.2}$$

$$\rho \circ (\sigma \cup \tau) = (\rho \circ \sigma) \cup (\rho \circ \tau) \tag{1.3}$$

$$\rho \circ (\sigma \cap \tau) \subset (\rho \circ \sigma) \cap (\rho \circ \tau) \tag{1.4}$$

$$\rho \circ id_A = id_A \circ \rho = \rho \tag{1.5}$$

Теорема 1.2. *Если некоторая пара (a, b) принадлежит какой-либо степени отношения ρ на множестве A мощности n, то эта пара принадлежит и степени ρ не выше $n - 1$:*

$$\rho \subset A^2 \wedge |A| = n \Rightarrow ((\forall a, b \in A)(\exists k \in \mathbb{N} : a\rho^k b) \Rightarrow (\exists k < n : a\rho^k b))$$

Доказательство. Пусть $a_0 = a, a_k = b$. $a\rho^k b \Rightarrow (\exists a_1, \ldots, a_{k-1} \in A : a_i \rho a_{i+1}, i = 0..k-1)$. Если $k \geq n$, в этой цепочке найдутся повторяющиеся элементы $a_i = a_j$. Следовательно, $a = a_0 \rho a_1 \rho \ldots \rho a_i \rho a_{j+1} \rho \ldots a_k = b$, чтд. □

Ядром бинарного отношения $\rho \subset A \times B$ называется бинарное отношение $\rho \circ \rho^{-1} \subset A^2$.

Пусть $\rho \subset A^2$. Бинарное отношение называется

1. *рефлексивным*, если $\forall a \in A : a\rho a$;
2. *иррефлексивным*, если $\forall a \in A : \neg a\rho a$;
3. *симметричным*, если $\forall a, b \in A : (a\rho b) \Rightarrow (b\rho a)$;
4. *антисимметричным*, если $\forall a, b \in A : (a\rho b) \wedge (b\rho a) \Rightarrow a = b$;
5. *асимметричным*, если $\forall a, b \in A : (a\rho b) \Rightarrow \neg b\rho a$;
6. *транзитивным*, если $\forall a, b, c \in A : (a\rho b) \wedge (b\rho c) \Rightarrow a\rho c$;
7. *полным* (линейным), если $\forall a, b \in A : a \neq b \Rightarrow a\rho b \vee b\rho a$;

8. *плотным*, если $\forall a, b \in A : (a \neq b \wedge a\rho b) \Rightarrow \exists c \in A : a \neq c \neq b : a\rho c \wedge c\rho b$

Теорема 1.3. *Пусть $\rho \subset A^2$. Тогда:*

1. *ρ рефлексивно $\Leftrightarrow I \subset \rho$;*
2. *ρ симметрично $\Leftrightarrow \rho = \rho^{-1}$;*
3. *ρ транзитивно $\Leftrightarrow \rho \circ \rho \subset \rho$;*
4. *ρ антисимметрично $\Leftrightarrow \rho \cap \rho^{-1} \subset I$;*
5. *ρ иррефлексивно $\Leftrightarrow \rho \cap I = \emptyset$;*
6. *ρ полно $\Leftrightarrow \rho \cup I \cup \rho^{-1} = A^2$;*

Доказательство. Используя определения свойств отношений, получим:

1. $\Rightarrow$: $\forall a \in A : a\rho a \Rightarrow \forall a \in A : (a, a) \in \rho \Rightarrow I \subset \rho$
   $\Leftarrow$: $I \subset \rho \Rightarrow \forall a \in A : (a, a) \in \rho \Rightarrow \forall a \in A : a\rho a$
2. $\Rightarrow$: $(\forall a, b \in A : (a\rho b) \Rightarrow (b\rho a)) \Rightarrow (\forall a, b \in A : (a, b) \in \rho \Rightarrow (b, a) \in \rho \Rightarrow (a, b) \in \rho^{-1}) \Rightarrow \rho \subset \rho^{-1}$ Аналогично $\rho^{-1} \subset \rho$.
   $\Leftarrow$: $\rho = \rho^{-1} \Rightarrow (((a, b) \in \rho) \Rightarrow (b, a) \in \rho^{-1} \Rightarrow (b, a) \in \rho)$
3. $\Rightarrow$: $\forall a, b, c \in A : (a\rho b) \wedge (b\rho c) \Rightarrow a\rho c; a(\rho \circ \rho)c \Rightarrow \exists b : a\rho b \wedge b\rho c \Rightarrow a\rho c \Rightarrow \rho \circ \rho \subset \rho$.
   $\Leftarrow$: $\rho \circ \rho \subset \rho \Rightarrow \forall a, c \in A : (a\rho \circ \rho c \Rightarrow a\rho c) \Rightarrow \forall a, b, c(a\rho b \wedge b\rho c \Rightarrow a\rho c)$
4. $\Rightarrow$: $\forall a, b \in A : (a\rho b \wedge b\rho a \Rightarrow a = b) \Rightarrow (b\rho^{-1}a \wedge b\rho a \Rightarrow a = b) \Rightarrow \rho \cap \rho^{-1} \subset I$
   $\Leftarrow$: $\rho \cap \rho^{-1} \subset I \Rightarrow \forall a, b \in A : (a\rho b \wedge a\rho^{-1}b \Rightarrow a = b) \Rightarrow (a\rho b \wedge b\rho a \Rightarrow a = b)$
5. $\Rightarrow$: От противного: $\rho \cap I \neq \emptyset \Rightarrow \exists a \in A : a\rho a \Rightarrow \rho$ не иррефлексивно
   $\Leftarrow$: $\rho \cap I = \emptyset \Rightarrow \neg\exists a \in Aa\rho a \Rightarrow \forall a \neg a\rho a$
6. $\Rightarrow$: $\forall a, b \in A : true = ((a \neq b) \Rightarrow (a\rho b \vee b\rho a)) = ((a = b) \vee a\rho b \vee a\rho^{-1}b) \Rightarrow A^2 = \rho \cup \rho^{-1} \cup I$
   $\Leftarrow$: $\forall a, b \in A^2 : (a, b) \in A^2 = \rho \cup \rho^{-1} \cup I \Rightarrow (a\rho b \vee b\rho a \vee a = b) = ((a \neq b) \Rightarrow a\rho b \vee b\rho a)$.

□

Некоторые комбинации свойств бинарных отношений очень часто встречаются на практике и потому заслуживают отдельного рассмотрения.

Определение 1.4. Бинарное отношения на некотором множестве называют отношением:

Таблица 1.1: Бинарные отношения и их свойства

	Рефлексивное	Иррефлексивное	Симметричное	Антисимметричное	Асимметричное	Транзитивное	Полное (линейное)	Плотное
Определение	$a \rho a$	$\neg a \rho a$	$(a \rho b) \Rightarrow (b \rho a)$	$(a \rho b) \wedge (b \rho a) \Rightarrow a = b$	$(a \rho b) \Rightarrow \neg b \rho a$	$(a \rho b) \wedge (b \rho c) \Rightarrow a \rho c$	$a \neq b \Rightarrow a \rho b \vee b \rho a$	$(a \neq b \wedge a \rho b) \Rightarrow \exists c \in A : a \neq c \neq b : a \rho c \wedge c \rho b$
Критерий	$I \subset \rho$	$\rho \cap I = \emptyset$	$\rho = \rho^{-1}$	$\rho \cap \rho^{-1} \subset I$		$\rho \circ \rho \subset \rho$	$\rho \cup I \cup \rho^{-1} = A^2$	
Классы отношений								
эквивалентность	✔		✔			✔		
толерантность	✔		✔					
(нестрогий) порядок	✔			✔		✔	линейный/ частичный	
предпорядок	✔					✔		
строгий порядок		✔		✔		✔		
строгий предпорядок		✔				✔		

1. *эквивалентности*, если оно рефлексивно, симметрично и транзитивно;
2. *толерантности*, если оно рефлексивно и симметрично;
3. порядка (или *частичного* порядка), если оно рефлексивно, антисимметрично и транзитивно;
4. *предпорядка* (квазипорядка), если оно рефлексивно и транзитивно;
5. *строгого* порядка, если оно иррефлексивно, антисимметрично и транзитивно;
6. *строгого* предпорядка, если оно иррефлексивно и транзитивно.

Сводка этих свойств и определений приведена в табл. 1.1.

Пример 1.1. Бинарное отношение сравнимости по модулю $a \equiv b \bmod N \Leftrightarrow (a - b) = cN, a, b, c \in \mathbb{Z}$ является отношением эквивалентности.

Пример 1.2. Бинарное отношение ρ на множестве всех непустых подмножеств некоторого множества U, для которого $A \rho B \Leftrightarrow A \cap B \neq \emptyset$, является толерантностью. Смысл названия становится ясен, если множества A, B, C (см. рис. 1.2) рассматривать как множество утверждений (взглядов) различных политических партий, идеологические платформы которых иногда имеют общие черты, что позволяет им сравнительно мирно сосуществовать.

Пример 1.3. Естественный числовой порядок $\geq$ является отношением нестрогого порядка.

Пример 1.4. На множестве натуральных чисел $\mathbb{N}$ зададим бинарное отношение $a|b$ (a делит b). Оно является отношением нестрогого порядка. Если его распространить на $\mathbb{Z}$, то теряется свойство антисимметричности, т.е. оно становится предпорядком.

Пример 1.5. Отношение включения множества $\subset$ является отношением порядка.

Пример 1.6. Отношения $<$ и $>$ есть отношения строгого порядка.

Пример 1.7. Отношение "быть степенью" $a\rho b \Leftrightarrow b = a^k, a \neq 1, k \in \mathbb{N}_{2+}$ является строгим предпорядком.

Можно сказать, что эквивалентность есть транзитивная толерантность или симметричный предпорядок, а порядок — асимметричный предпорядок.

Целесообразно изучить свойства абстрактных отношений, что позволит в дальнейшем автоматически распространять их на конкретные примеры бинарных отношений. Примером применения такого подхода является реализация библиотеки STL в C++. Здесь будут подробно рассмотрены отношения эквивалентности и порядка.

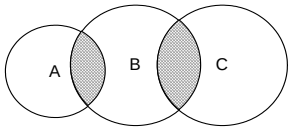


Рис. 1.2: Отношение толерантности $A\rho B$, $B\rho C$

1.2.4 Отношения эквивалентности

Определение 1.5. Пусть $\equiv$ — *отношение эквивалентности* на множестве A и $x \in A$. Подмножество элементов A, эквивалентных x, называется *классом эквивалентности* для x

$$[x]_{\equiv} = \{y \in A | y \equiv x\}$$

Лемма 1.2. $\forall a \in A [a] \neq \emptyset$

Доказательство. $a \equiv a \Rightarrow a \in [a]$. □

Лемма 1.3. $a \equiv b \Rightarrow [a] = [b]$

Доказательство. $\forall x : x \in [a] \Rightarrow x \equiv a \wedge a \equiv b \Rightarrow x \equiv b \Rightarrow x \in [b] \Rightarrow [a] \subset [b]$. $\forall x : x \in [b] \Rightarrow x \equiv b \wedge a \equiv b \Rightarrow x \equiv b \wedge b \equiv a \Rightarrow x \equiv a \Rightarrow x \in [a] \Rightarrow [b] \subset [a]$. □

Лемма 1.4. $a \not\equiv b \Rightarrow [a] \cap [b] = \emptyset$

Доказательство. От противного: $\exists c \in [a] \cap [b] \Rightarrow c \in [a] \wedge c \in [b] \Rightarrow c \equiv a \wedge c \equiv b \Rightarrow a \equiv b$ □

Определение 1.6. Пусть $\mathcal{A} = \{A_i\}, A_i \subset A$ — некоторое семейство подмножеств. Семейство $\mathcal{A}$ называется *покрытием*множества A, если каждый элемент A принадлежит хотя бы одному из A_i, т.е.

$$A = \bigcup_i A_i.$$

Семейство $\mathcal{A}$ называется дизъюнктным, если его элементы не пересекаются. Дизъюнктное покрытие называется *разбиением* множества A.

Теорема 1.4. *Всякое отношение эквивалентности на множестве A определяет его разбиение, причем среди элементов разбиения нет пустых. Обратно, всякое разбиение, не содержащее пустых элементов, определяет отношение эквивалентности.*

Доказательство. Разбиение с непустыми элементами может быть построено по отношению эквивалентности следующим образом:

1. Выбрать произвольный элемент $a \in A$.
2. Построить класс эквивалентности $[a]$.
3. $A := A \setminus [a]; \mathcal{A} := \mathcal{A} \cap \{[a]\}$.
4. Если A непусто, перейти к шагу 1.

Достаточность. Пусть $a \equiv b \Leftrightarrow \exists i : a \in A_i \wedge b \in A_i$

1. Рефлексивность: $A = \bigcup_i A_i \Rightarrow \forall a \in A \exists i : a \in A_i \wedge a \in A_i \Rightarrow a \equiv a$.
2. Симметричность: $a \equiv b \Rightarrow \exists i : a \in A_i \wedge b \in A_i \Rightarrow \exists i : b \in A_i \wedge a \in A_i \Rightarrow b \equiv a$.
3. Транзитивность: $a \equiv b \wedge b \equiv c \Rightarrow (\exists i : a \in A_i \wedge b \in A_i) \wedge (\exists j : b \in A_j \wedge c \in A_j)$. Т.к. $A_i \cap A_j = \emptyset, i \neq j$ получаем, что $i = j$, т.е. $\exists i : a \in A_i \wedge c \in A_i \Rightarrow a \equiv c$.

Следовательно, определенное таким образом $\equiv$ есть отношение эквивалентности.

Замечание 1.1. В книге Новикова транзитивность доказывается через $[a] = [b] = [c]$, что неверно.

□

Определение 1.7. Если ρ — отношение эквивалентности на множестве A, то множество классов эквивалентности называется *фактормножеством* множества A по эквивалентности ρ

$$A/\rho = \{[a]_\rho | a \in A\}$$

Фактор-множество является подмножеством булеана 2^A

1.2.5 Отношения порядка и упорядоченные множества

Рефлексивное антисимметричное транзитивное отношение называется отношением *нестрогого порядка*. Иррефлексинвное антисимметричное транзитивное отношение называется отношением *строгого порядка*. Отношение порядка может быть полным (линейным), и тогда оно называется отношением *полного* или *линейного* порядка.В противном случае говорят об отношении частичного порядка. Множество, на которым определено отношение полного/частичного порядка, называется полностью/частично *упорядоченным*.

Двойственным порядком называют отношение $\prec^{-1}$.

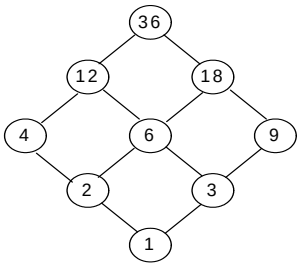


Рис. 1.3: Диаграмма Хассе отношения $a|b$

Определение 1.8. Для двух элементов $x, y \in A$ по определению $x \lhd y$ тогда и только тогда, когда $x \prec y \wedge \nexists z \in A : x \prec z \prec y$. Отношение $\lhd$ называется *отношением доминирования*. Если $x \lhd y$, то y доминирует над x.

Конечное упорядоченное множество удобно графически изображать в виде диаграммы Хассе. Элементы множества изображаются в виде кружочков, причем если $a \lhd b$, то b изображается выше a и соединяется с ним прямой линией. Иногда на линии из a в b изображается стрелка.

Отношение доминирования иррефлексивно, антисимметрично, но не транзитивно. Если исходный порядок является плотным, то отношение доминирования будет пустым. В этом случае отношение порядка может быть представлено в виде отрезка $[a, b]_{\prec} = \{x | a \prec x \wedge x \prec b\}$.

Пример 1.8. Рассмотрим отношение делимости $a|b$ на некотором подмножестве натуральных чисел. Ясно, что оно рефлексивно, антисимметрично и транзитивно, т.е. задает нестрогий частичный порядок. Диаграмма Хассе для множества чисел, являющихся делителями 36, представлена на рис. 1.3.

Определение 1.9. Элемент x множества A с отношением порядка $\preceq$ называется *минимальным*, если

$$\nexists y \in A : y \preceq x \wedge y \neq x.$$

Наименьшим элементом упорядоченного множества называется такой его элемент $x \in A$, что $\forall y \in A : x \preceq y$.

Наименьший элемент множества (если он существует), является единственным. Действительно, предположим, что $\exists x, x' \in A : \forall y \in A : (x \preceq y) \wedge (x' \preceq y) \Rightarrow (x \preceq x') \wedge (x' \preceq x) \Rightarrow x = x'$. Вместе с тем, минимальных элементов в множестве может быть сколько угодно. Например, введем отношение порядка на плоскости с некоторой фиксированной декартовой системой координат как $(a, b) \preceq (c, d) \Leftrightarrow (a \leq c) \wedge (b \leq d)$. Рассмотрим множество точек треугольника, вершинами которого являются точки $(1, 0), (1, 1), (0, 1)$. Все точки, лежащие на отрезке $(1, 0) - (0, 1)$ являются минимальными, а наименьшего элемента нет. Однако если у множества есть наибольший/наименьший элемент, то он является единственным максимальным/минимальным.

Последовательность $x_i, i \in \mathbb{N}$ называется неубывающей, если $\forall i \in \mathbb{N} : x_i \preceq x_{i+1}$.

Теорема 1.5. *Во всяком конечном непустом частично упорядоченном множестве существует минимальный элемент.*

Доказательство. От противного. Пусть $\neg(\exists x \in A : \nexists y \in A : y \preceq x \wedge y \neq x) \Rightarrow \forall x \in A \exists y \in A : y \preceq x \wedge y \neq x \Rightarrow \exists z_i \in A, i = 1..\infty : z_{i+1} \preceq z_i, z_{i+1} \neq z_i$. В силу конечности множества $\exists i, j : i < j \wedge z_i = z_j$. В силу транзитивности $z_i \succeq z_{i+1} \succeq \ldots \succ z_j \Rightarrow z_i \succ z_j$. Кроме того, $z_j \preceq z_{i+1} \Rightarrow z_i \preceq z_{i+1}$. Т.е. имеет место противоречие $z_{i+1} \neq z_i \wedge z_i \preceq z_{i+1} \wedge z_{i+1} \preceq z_i$. □

Теорема 1.6. *Всякий частичный порядок на конечном множестве может быть дополнен до полного (линейного).*

Доказательство. Линейный порядок элементов некоторого множества может быть задан списком элементов. В соответствии с теоремой 1.5, в конечном частично упорядоченном множестве существует минимальный элемент. Список элементов, задающий линейный порядок, может быть сформирован путем последовательного выбора (с удалением) из множества минимальных элементов (*топологическая сортировка*). □

Пусть $(A, \preceq)$ — упорядоченное множество и $B \subset A$. Элемент $a \in A$ называется верхней (нижней) гранью множества B, если $\forall x \in B : (x \preceq a)$ (или $x \succeq a$). Наименьший/наибольший элемент множества верхних/нижних граней является точной верхней/нижней гранью множества B и обозначается $\sup B$ или $\inf B$. Упорядоченное множество $(A, \preceq)$ называется *индуктивным*, если

- Оно содержит наименьший элемент;
- Всякая неубывающая последовательность элементов этого множества имеет точную верхнюю грань.

Определение 1.10. Пусть $(M_1, \preceq)$ и $(M_2, \sqsubseteq)$ — индуктивные упорядоченные множества. Отображение $f : M_1 \longrightarrow M_2$ называется *непрерывным*, если для любой неубывающей последовательности $\alpha_1, \ldots, \alpha_n, \ldots$ элементов множества M_1 образ ее точной верхней грани равен точной верхней грани последовательности образов $f(\alpha_1), \ldots, f(\alpha_n), \ldots$, т.е. $f(\sup \alpha_n) = \sup f(\alpha_n)$.

Определение 1.11. Отображение $f : M_1 \longrightarrow M_2$ упорядоченных множеств называется *монотонным*, если $\forall a, b \in M_1 : a \preceq b \Rightarrow f(a) \preceq f(b)$.

Это определение не следует путать с определением монотонных функций, используемым в матанализе. Функция $f : \mathbb{R} \longrightarrow \mathbb{R}$, монотонная в смысле матанализа, будет являться монотонной в смысле определения 1.11, только если она является неубывающей.

Теорема 1.7. *Всякое непрерывное отображение одного индуктивного упорядоченного множества в другое монотонно.*

Доказательство. Пусть $f : M_1 \longrightarrow M_2$ непрерывно, M_1, M_2 — индуктивные множества. Пусть $a, b \in M_1, a \preceq b$. Образуем неубывающую последовательность $\{x_n\}_{n\in\mathbb{N}}, x_1 = a, x_n = b, n \geq 2$. Для нее $\sup x_n = b$. В силу непрерывности $f(b) = f(\sup x_n) = f(\sup\{a, b\}) = \sup\{f(a), f(b)\} \Rightarrow f(a) \preceq f(b)$. □

Пример 1.9. Функция $f = \begin{cases} 0,5x, & 0 \leq x < 0.5 \\ 0,5 + 0,5x, & 0.5 \leq x \leq 1 \end{cases}$ является монотонной, но не непрерывной.

Определение 1.12. Элемент $a \in A$ называется *неподвижной точкой* отображения $f : A \longrightarrow A$, если $f(a) = a$. Элемент a называется наименьшей неподвижной точкой отображения $f : A \longrightarrow A$, если он является наименьшим элементом множества неподвижных точек f.

Теорема 1.8 (О неподвижной точке)**.** *Любое непрерывное отображение f индуктивного упорядоченного множества $(M, \preceq)$ в себя имеет наименьшую неподвижную точку.*

Доказательство. Пусть $\mu \in M$ — наименьший элемент множества M. Пусть $f^0(x) = x, f^n(x) = f(f^{n-1}(x)), n > 0$. Рассмотрим последовательность элементов M

$$\{f^n(\mu)\}_{n\geq 0} = \{\mu, f(\mu), \ldots, f^n(\mu), \ldots\}. \tag{1.6}$$

Докажем, что эта последовательность неубывающая. Т.к. μ — наименьший элемент, $\mu \preceq f(\mu)$. Пусть для некоторого $n\, f^{n-1}(\mu) \preceq f^n(\mu)$. Т.к. f непрерывно, по теореме 1.7 оно является монотонным, следовательно $f^n(\mu) = f(f^{n-1}(\mu)) \preceq f(f^n(\mu)) = f^{n+1}(\mu)$. Таким образом, последовательность (1.6) является неубывающей. Тогда по определению индуктивного множества она имеет точную верхнюю грань $a = \sup_{n\geq 0} f^n(\mu)$, т.е. $\forall n : f^n(\mu) \preceq a$. Ясно, что a является также верхней гранью для любой подпоследовательности (1.6), в т.ч. для $n \geq k > 0$. Пусть b — какая-то иная верхняя грань такой последовательности с усеченным началом, т.е. $\forall n \geq k : x_n \preceq b$. Т.к. исходная последовательность неубывающая, $x_p \preceq x_k, p = 0..k-1$, а следовательно $x_p \preceq x_k \preceq b$. Т.к. $a = \sup_{n\geq 0} f^n(\mu)$, $a \preceq b$, т.е. является точной верхней гранью любой последовательности вида (1.6) с усеченным началом.

В силу непрерывности f имеем

$$f(a) = f(\sup_{n\geq 0} f^n(\mu)) = \sup_{n\geq 0} f(f^n(\mu)) = \sup_{n\geq 0} f^{n+1}(\mu)$$

Но

$$\sup_{n\geq 0} f^{n+1}(\mu) = \sup\{f^1(\mu), f^2(\mu), \ldots\} = \sup_{n\geq 1} f^n(\mu) = a,$$

т.е. a является неподвижной точкой.

Докажем минимальность этой неподвижной точки. Пусть $\exists y \in M : f(y) = y$. Т.к. $\mu \preceq y$, а f, будучи непрерывным, монотонно, то $f(\mu) \preceq f(y) = y, f^2(\mu) \preceq f^2(y) = y$ и т.д., т.е. $\forall n \geq 0 : f^n(\mu) \preceq y$, т.е. y является верхней гранью последовательности $\{f^n(\mu)\}$. Т.к. a является точной верхней гранью, $a \preceq y$, т.е. a — наименьшая неподвижная точка. □

Пример 1.10. Пусть $f(x) = \frac{1}{2}x + \frac{1}{4} : [0,1] \longrightarrow [0,1]$. Применяя метод, использованный в доказательстве теоремы, получим $f^0(0) = 0, f^1(0) = 1/4, f^2(0) = 3/8, f^n(0) = \frac{2^n-1}{2^{n+1}} \underset{n\to\infty}{\longrightarrow} 1/2$.

1.2.6 Замыкания отношений

Замкнутость означает, что многократное выполнение допустимых шагов не выводит за определенные границы.

Отношение ρ' называется *замыканием* ρ относительно свойства C, если

1. ρ' обладает свойством C: $C(\rho')$;
2. $\rho \subset \rho'$;
3. ρ' является наименьшим: $C(\rho'') \wedge (\rho \subset \rho'') \Rightarrow \rho' \subset \rho''$

Для любого бинарного отношения ρ можно построить отношения

$$\rho^+ = \bigcup_{i=1}^{\infty} \rho^i \tag{1.7}$$

$$\rho^* = \bigcup_{i=0}^{\infty} \rho^i. \tag{1.8}$$

Теорема 1.9. *ρ^+ является транзитивным замыканием ρ.*

Доказательство. 1. $a\rho^+b \wedge b\rho^+c \Rightarrow \exists m,n \in \mathbb{N} : a\rho^m b \wedge b\rho^n c \Rightarrow \exists a_1,\ldots,a_{m-1},c_{n-1},\ldots,c_1 : a\rho a_1\rho\ldots\rho a_{m-1}\rho b\rho c_{n-1}\rho\ldots\rho c_1\rho c \Rightarrow a\rho^{m+n-1}c \Rightarrow a\rho^+c.$

2. $\rho = \rho^1 \subset \rho^+$.

3. $a\rho^+b \Rightarrow \exists k : a\rho^k b \Rightarrow \exists c_1,\ldots,c_k : a\rho c_1\rho\ldots\rho c_{k-1}\rho b; \rho \subset \rho'' \Rightarrow a\rho'' c_1\rho''\ldots\rho'' c_{k-1}\rho'' b.$ Т.к. ρ'' транзитивно, $a\rho'' b$. Следовательно, $\rho^+ \subset \rho''$.

□

Теорема 1.10. *ρ^* — рефлексивное и транзитивное замыкание ρ*

Доказательство. Очевидно. □

Транзитивное замыкание произвольного отношения ρ на множестве $M : |M| = n$, заданного матрицей R, может быть найдено с помощью алгоритма Уоршалла.

TRANSITIVECLOSURE(R)

```
1  S := R;
2  for i := 1 to n
3      do for j := 1 to n
4              do for k := 1 to n
```

5 **do** $T[j,k] := S[j,k] \vee S[j,i] \wedge S[i,k]$
6 $S := T$;
7 **return** (T);

Действительно, на каждом шаге цикла по i в транзитивное замыкание добавляются пары элементов (j,k), для которых существуют последовательность промежуточных элементов с номерами в диапазоне от 1 до i, связанных отношением ρ.

1.3 Основные результаты

1. Теорема 1.2: если некоторая пара (a,b) принадлежит какой-либо степени отношения ρ на множестве A мощности n, то эта пара принадлежит и степени ρ не выше $n-1$.

2. Теорема 1.4 Всякое отношение эквивалентности на множестве A определяет его разбиение, причем среди элементов разбиения нет пустых. Обратно, всякое разбиение, не содержащее пустых элементов, определяет отношение эквивалентности.

3. Теорема 1.6: всякий частичный порядок на конечном множестве может быть дополнен до полного (линейного).

4. Теорема 1.8: любое непрерывное отображение f индуктивного упорядоченного множества $(M, \preceq)$ в себя имеет наименьшую неподвижную точку.

Упражнения

1. Доказать, что $A \times (B \cap C) = (A \times B) \cap (A \times C)$
2. Доказать, что $(A \cap B) \times (C \cap D) = (A \times C) \cap (B \times D)$
3. Доказать, что $\overline{A \times B} \neq \overline{A} \times \overline{B}$
4. Доказать, что $(A \subset X) \wedge (B \subset Y) \Rightarrow (A \times B) \subset (X \times Y)$
5. Доказать, что $(A \setminus B) \times C = (A \times C) \setminus (B \times C)$
6. Доказать, что для любой функции имеют место соотношения $f(A \cup B) = f(A) \cup f(B); f(A \cap B) \subset f(A) \cap f(B); f(A) \setminus f(B) \subset f(A \setminus B)$. При каких условиях имеют место равенства?
7. Доказать, что для любой функции имеют место соотношения $f^{-1}(A \cup B) = f^{-1}(A) \cup f^{-1}(B), f^{-1}(A \cap B) = f^{-1}(A) \cap f^{-1}(B), f^{-1}(\overline{A}) = \overline{f^{-1}(A)}, f^{-1}(A \setminus B) = f^{-1}(A) \setminus f^{-1}(B)$
8. Какими свойствами обладает отношение $Q = \{(m,n) | m,n \in \mathbb{N}^2, m = n^2\}$?
9. Построить графики и графы следующих бинарных отношений, заданных на множестве $\{1,2,3,4,5,6\}$:

(a) $x_1 \phi x_2 \Leftrightarrow x_1 \leq x_2$

(b) $x_1 \phi x_2 \Leftrightarrow x_1 + x_2$ — четное

10. Доказать, что если $\equiv$ — отношение эквивалентности на конечном множестве M, то $|M| = |M/\equiv| \Rightarrow \forall x \in M |[x]_{\equiv}| = 1$.

11. Пусть A — вполне упорядоченное множество с отношением порядка ρ. Введем отношение на множестве кортежей элементов из a

$$(a_1, \ldots, a_m)\tau(b_1, \ldots, b_n) := (m \leq n \wedge \forall i = 1..m : a_i = b_i) \vee (\exists i \in 1..m : (a_i \rho b_i) \wedge \forall j < i : a_j = b_j)$$

Такое отношение называется лексикографическим (алфавитным) порядком. Доказать, что оно является полным порядком на множестве кортежей $A^+ = \bigcup_{i=1}^{\infty} A^i$.

Глава 2

Алгебраические системы

2.1 Понятие алгебры

2.1.1 Основные определения

Определение 2.1. Пусть A — произвольное непустое множество и $n \in \mathbb{N}$. Любое *отображение*

$$\omega : A^n \to A$$

называется *n-арной операцией* на множестве A.

Компоненты кортежа $(a_1, \ldots, a_n)$ называются аргументами операции, а $b = \omega(a_1, \ldots, a_n)$ — результатом применения операции к аргументам $a_1, \ldots, a_n$. n-арная операция обозначается также как $a_1 \ldots a_n \omega$. Бинарная операция обозначается также как $a_1 \omega a_2$. При $n = 1$ и $n = 2$ говорят об унарной и бинарной операции. *Нульарной операцией* на множестве A называется любой фиксированный элемент $a \in A$.

Наибольший интерес преджставляют бинарные операции. Пусть * — бинарная операция. Ее называют

1. Ассоциативной, если $\forall x, y, z \in A : (x * y) * z = x * (y * z)$.
2. Коммутативной, если $\forall x, y \in A : x * y = y * x$.
3. Идемпотентной, если $\forall x \in A : x * x = x$.

Элемент $\mathbf{0} \in A$ называется левым (правым) *нулем* относительно данной операции $*$, если $\forall x \in A : \mathbf{0} * x = \mathbf{0}$ ($\forall x \in A : x * \mathbf{0} = \mathbf{0}$). Если $\mathbf{0}'$ — левый нуль, а $\mathbf{0}''$ — правый нуль, то они совпадают. Действительно, $\mathbf{0}' = \mathbf{0}' * \mathbf{0}'' = \mathbf{0}''$. В этом случае говорят о нуле относительно операции *. Нуль операции является единственным.

Пример 2.1. На множестве целых чисел нулем относительно операции умножения является число 0.

На множестве квадратных матриц вида $\begin{pmatrix} a & 0 \\ b & 1 \end{pmatrix}, a, b \in \mathbb{R}$ любая матрица вида $\begin{pmatrix} 0 & 0 \\ d & 1 \end{pmatrix}$ будет правым нулем относительно операции умножения, т.к.

$$\begin{pmatrix} a & 0 \\ b & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ d & 1 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ d & 1 \end{pmatrix}.$$

Однако левого нуля в этом множестве нет, т.к. иначе он совпадал бы с правым нулем и был единственным. Но правых нулей имеется много.

Элемент $\mathbf{1} \in A$ называется левым (правым) *нейтральным* относительно операции *, если $\forall x \in A : \mathbf{1} * x = x$ ($\forall x \in A : x * \mathbf{1} = x$). Если существуют левый и правый нейтральные элементы, то они совпадают: $\mathbf{1}' = \mathbf{1}' * \mathbf{1}'' = \mathbf{1}''$. В этом случае нейтральный элемент единственен.

Пример 2.2. Нейтральным элементом относительно операции умножения целых чисел является 1. Нейтральным элементом относительно операции сложения является 0.

На множестве квадратных матриц вида $\begin{pmatrix} a & 0 \\ b & 0 \end{pmatrix}, a, b \in \mathbb{R}$ любая матрица вида $\begin{pmatrix} 1 & 0 \\ d & 0 \end{pmatrix}$ является правым нейтральным элементом относительно операции умножения, т.к.

$$\begin{pmatrix} a & 0 \\ b & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ d & 0 \end{pmatrix} = \begin{pmatrix} a & 0 \\ b & 0 \end{pmatrix}.$$

Т.к. правых нейтральных элементов несколько, левого нейтрального элемента не существует.

Не для всякой операции существуют нулевые и нейтральные элементы.

Пример 2.3.

1. Пусть U — некоторый универсум. Операции $\cup, \cap$ на 2^U являются идемпотентными, ассоциативными и коммутативными, причем $\emptyset$ является нулем относительно $\cap$ и нейтральным элементом относительно $\cup$. Операция $\setminus$ не является ассоциативной, т.к. $A \setminus (B \setminus C) \neq (A \setminus B) \setminus C$.

2. На множестве всех бинарных отношений операция композиции отношений является ассоциативной, но не коммутативной, а диагональ множества A является нейтральным элементом относительно этой операции.

3. Пусть $X : |X| \geq 2$. На множестве всех отображений из X в X с операцией композиции отображений постоянное отображение $\phi_a(x) = a, a \in A$ будет правым нулем, но не будет левым нулем. Действительно $f \circ \phi_a = \phi_a(f(x)) = a = \phi_a$, но $\phi_a \circ f = f(\phi_a(x)) = f(a) \neq a = \phi_a$

Определение 2.2. *Алгебра* (универсальная алгебра, Ω-алгебра) $\mathcal{A} = (A, \Omega)$ считается заданной, если заданы некоторое множество A, называемое носителем этой алгебры, и некоторое множество операций Ω на A, называемое сигнатурой алгебры. Алгебра, носителем которой является конечное множество, называется *конечной*.

Пример 2.4. Примеры алгебр:

1. $\mathcal{A}_1 = (2^M, \{\cup, \cap, \setminus, \bigtriangleup, \text{-}, \emptyset, M\})$
2. алгебра бинарных отношений: $\mathcal{A}_2 = (2^{M\times M}, \{\cap, \circ, ^{-1}\})$
3. $\mathcal{A}_3 = (\mathbb{R}, \{+, \cdot, 0, 1\})$
4. Алгебра с бесконечной сигнатурой $\mathcal{A}_4 = (\mathbb{R}, \{\uparrow^n | n \geq 2\})$
5. Множество векторов с операцией скалярного произведения алгеброй не является.

Определение 2.3. Две алгебры $\mathcal{A}_1 = (A_1, \Omega_1)$ и $\mathcal{A}_2 = (A_2, \Omega_2)$ называют однотипными, если существует биекция $f : \Omega_1 \longrightarrow \Omega_2$, при которой для всех n n-арная операция из Ω_1 переходит в n-арную операцию из Ω_2.

Пример 2.5. Алгебра $\{2^M, \cup, \cap, \emptyset, M\}$, заданная на 2^M, и алгебра $\{\mathbb{R}, +, \cdot, 0, 1\}$ однотипны. Биекция может быть определена. например, как $\cup \longrightarrow +, \cap \longrightarrow \cdot, \emptyset \longrightarrow 1, M \longrightarrow 1$. Эта алгебра не является однотипной с $\mathcal{A}_1$ из примера 2.4.

Некоторые часто встречающиеся свойства операций имеют свои названия. Пусть задана $\mathcal{A} = (A, \{\circ, *\})$. Тогда

1. Ассоциативность $(a * b) * c = a * (b * c)$;
2. Коммутативность $a * b = b * a$;
3. Дистрибутивность слева $a * (b \circ c) = (a * b) \circ (a * c)$
4. Дистрибутивность справа $(a * b) \circ c = (a \circ c) * (b \circ c)$
5. Поглощение $(a * b) \circ a = a$
6. Идемпотентность $a * a = a$

2.1.2 Морфизмы

Определение 2.4. Пусть $\mathcal{A} = (A, \{\phi_1, \ldots, \phi_m\})$ и $\mathcal{B} = (B, \{\psi_1, \ldots, \psi_m\})$ две однотипные алгебры. Если существует $f : A \longrightarrow B$, такая что

$$\forall i = 1..m : f(\phi_i(a_1, \ldots, a_{n_i})) = \psi_i(f(a_1), \ldots, f(a_{n_i})),$$

то f называется *гомоморфизмом* из A в B. Гомоморфизм, являющийся биекцией, называется *изоморфизмом*.

Понятие гомоморфизма можно проиллюстрировать с помощью коммутативной диаграммы:

$$\begin{array}{ccc} A & \xrightarrow{\phi_i} & A \\ {\scriptstyle f}\downarrow & & {\scriptstyle f}\downarrow \\ B & \xrightarrow{\psi_i} & B \end{array}$$

Диаграмма называется коммутативной потому, что условие гомоморфизма может быть переписано как $\phi \circ f = f \circ \psi$.

Пример 2.6. Пусть $\mathcal{A} = (\mathbb{N}, +), \mathcal{B} = (\mathbb{N}_{10}, +_{10})$, где $\mathbb{N}_{10} = \{0, 1, \ldots, 9\}, +_{10}(x, y) = x + y \bmod 10$. Тогда $f = a \bmod 10$ является гомоморфизмом из $\mathcal{A}$ в $\mathcal{B}$.

Лемма 2.1. *Если $f : A \longrightarrow B$ — изоморфизм, то $f^{-1} : B \longrightarrow A$ — тоже изоморфизм.*

Доказательство. Пусть ϕ — произвольная операция из сигнатуры $\mathcal{A}$, а ψ — соответствующая ей операция из сигнатуры $\mathcal{B}$. Т.к. f — изоморфизм, то он является и гомоморфизмом, т.е.

$$f(\phi(a_1, \ldots, a_n)) = \psi(f(a_1), \ldots, f(a_n)). \tag{2.1}$$

Т.к. f биективна, пусть $b_i = f(a_i), a_i = f^{-1}(b_i)$. Применяя к обеим частям (2.1) f^{-1}, получим

$$\phi(f^{(-1)}(b_1), \ldots, f^{(-1)}(b_n)) = f^{-1}(\psi(b_1, \ldots, b_n))$$

□

Лемма 2.2. *Если $f : A \longrightarrow B$ и $g : B \longrightarrow C$ — гомоморфизмы, то $f \circ g : A \longrightarrow C$ также является гомоморфизмом.*

Доказательство. □

Теорема 2.1. *Отношение изоморфизма на множестве однотипных алгебр является эквивалентностью.*

Доказательство. 1. Рефлексивность: $\mathcal{A} \overset{f}{\sim} \mathcal{A}, f = I$.

2. Симметричность: $\mathcal{A} \overset{f}{\sim} \mathcal{B} \Rightarrow \mathcal{B} \overset{f^{-1}}{\sim} \mathcal{A}$

3. Транзитивность: $\mathcal{A} \overset{f}{\sim} \mathcal{B} \wedge \mathcal{B} \overset{g}{\sim} \mathcal{C} \Rightarrow \mathcal{A} \overset{f \circ g}{\sim} \mathcal{C}$

□

Пример 2.7. Алгебра натуральных чисел $(\mathbb{N}, +)$ изоморфна алгебре четных чисел: $(\mathbb{N}, +) \overset{\times 2}{\sim} (\{n | n = 2k, k \in \mathbb{N}\}, +)$. Алгебра вещественных чисел с операцией умножения изоморфна алгебре вещественных чисел с операцией сложения : $(\mathbb{R}, \cdot) \overset{\ln}{\sim} (\mathbb{R}, +)$

Свойство изоморфизма позволет переносить свойства (формулы), полученные для одной алгебры, на другую алгебру. Поэтому принято рассматривать алгебраические структуры с точностью до изоморфизма, т.е. классы эквивалентности по отношению изоморфизма.

2.2 Группоиды, полугруппы, группы

2.2.1 Основные понятия

Рассмотрим алгебры с сигнатурами, состоящими из одной бинарной операции.

Определение 2.5. *Группоидом* называют любую алгебру $\mathcal{G} = (G, \cdot)$, сигнатура которой состоит из одной бинарной операции, на которую не накладывается никаких ограничеинй. Группоид называется *полугруппой*, если его операция ассоциативна, т.е. $a \cdot (b \cdot c) = (a \cdot b) \cdot c$.

Пример 2.8. Множество натуральных чисел $\mathbb{N}$ вместе с операцией возведения в степень является группоидом, т.к. $\left(a^b\right)^c \neq a^{(b^c)}$. Множество натуральных чисел $\mathbb{N}$ вместе с операцией сложения является полугруппой.

Определение 2.6. Группоид называется *моноидом*, если его операция ассоциативна и относительно нее существует нейтральный элемент, т.е. $\mathbf{1} \cdot a = a \cdot 1 = a$.

Т.к. нейтральный элемент является единственным, его можно внести в сигнатуру моноида как нульарную операцию.

Полугруппы с коммутативной операцией называются коммутативными полугруппами.

Пример 2.9. Множество всех бинарных отношений на множестве A с операцией композиции является моноидом, нейтральным элементом которого является диагональ A.

Определение 2.7. Группоид $\mathcal{G} = (G, \circ)$ называется *группой*, если операция ассоциативна, существует нейтральный элемент $\mathbf{1}$ и $\forall x \in G \exists x^{-1} : x \circ x^{-1} = x^{-1} \circ x = \mathbf{1}$. Элемент x^{-1} называется *обратным* к x.

Теорема 2.2. *В любой группе для любого элемента x обратный к нему элемент единственен.*

Доказательство. Предположим, что для некоторого элемента a нашлись два обратных a', a''. Тогда $a' = a' \circ \mathbf{1} = a' \circ (a \circ a'') = (a' \circ a) \circ a'' = \mathbf{1} \circ a'' = a''$. □

Таким образом, в сигнатуру группы может быть введена унарная операция нахождения обратного элемента. Наиболее популярны аддитивная и мультипликативная формы описания групп. В таблице 2.1 приведено сравнение этих способов.

Теорема 2.3. *В любой группе $\mathcal{G} = (G, \circ)$ выполняются следующие соотношения:*

1. $(a \circ b)^{-1} = b^{-1} \circ a^{-1}$
2. $a \circ b = a \circ c \Rightarrow b = c$
3. $b \circ a = c \circ a \Rightarrow b = c$

Таблица 2.1: Аддитивный и мультипликативный способы задания группы

	Аддитивный	Мультипликативный
Символ операции	$+$	$\cdot$
Название	сложение	умножение
Нейтральный элемент	$\mathbf{0}$	$\mathbf{1}$
Элемент, обратный к a	$-a$	a^{-1}

4. $(a^{-1})^{-1} = a$

Доказательство. 1. $(a \circ b) \circ (b^{-1} \circ a^{-1}) = a \circ (b \circ b^{-1}) \circ a^{-1} = a \circ \mathbf{1} \circ a^{-1} = a \circ a^{-1} = \mathbf{1}$.

2. $a \circ b = a \circ c \Rightarrow a^{-1} \circ (a \circ b) = a^{-1} \circ (a \circ c) \Rightarrow (a^{-1} \circ a) \circ b = (a^{-1} \circ a) \circ c \Rightarrow \mathbf{1} \circ a = \mathbf{1} \circ b \Rightarrow a = b$.

3. $b \circ a = c \circ a \Rightarrow (b \circ a) \circ a^{-1} = (c \circ a) \circ a^{-1}$. Далее — аналогично.

4. $(a^{-1}) \circ a = a^{-1} \circ a = \mathbf{1}$.

□

Теорема 2.4. *Единственным решением уравнения $a \circ x = b$ в любой группе является $x = a^{-1} \circ b$.*

Доказательство. $a \circ x = b \Rightarrow a^{-1} \circ (a \circ x) = a^{-1} \circ b \Rightarrow (a^{-1} \circ a) \circ x = a^{-1} b \Rightarrow \mathbf{1} \circ x = a^{-1} b \Rightarrow x = a^{-1} b$, т.е. $a^{-1} b$ действительно является решением уравнения. Предположим, что $a \circ x' = b$ и $a \circ x'' = b$. Тогда $a \circ x' = a \circ x'' \Rightarrow x' = x''$, т.е. решение единственно. □

Группы, бинарная операция которых коммутативна, называются коммутативными (*абелевыми*) группами. В этом случае при использовании аддитивной формы записи уравнение имеет вид $a + x = b$, а ее решение $x = b + (-a)$ называют разностью и обозначают $x = b - a$. При использовании мультипликативной формы записи решение $x = b \cdot a^{-1}$ называют частным от деления b на a и обозначают $x = \frac{b}{a}$.

Пример 2.10. Рассмотрим группу S_n подстановок n-й степени всех биекций n-элементного множества $\{1, 2, \ldots, n\}$. Произвольная биекция может быть записана в виде

$$\begin{pmatrix} 1 & 2 & \ldots & n \\ \alpha_1 & \alpha_2 & \ldots & \alpha_n \end{pmatrix}$$

Биекцию множества $\{1, 2, \ldots, n\}$ на себя называют подстановкой этого множества. Подстановка, отображающая α_1 в α_2, α_2 в α_3, $\ldots$, α_k в α_1 называют циклом длины k и обозначают $(\alpha_1, \ldots, \alpha_k)$. Например, подстановка $\begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 2 & 4 & 1 \end{pmatrix}$ может быть записана как $(1, 3, 4), (2)$. Циклы длины 1 иногда не указывают. Для всякой подстановки может быть определена обратная.

Решим в группе S_3 уравнение

$$\begin{pmatrix}1&2&3\\3&1&2\end{pmatrix}\circ X\circ\begin{pmatrix}1&2&3\\2&3&1\end{pmatrix}=\begin{pmatrix}1&2&3\\3&2&1\end{pmatrix}$$

Умножим обе части слева на $\begin{pmatrix}1&2&3\\3&1&2\end{pmatrix}^{-1}=\begin{pmatrix}1&2&3\\2&3&1\end{pmatrix}$. Получим

$$X\circ\begin{pmatrix}1&2&3\\2&3&1\end{pmatrix}=\begin{pmatrix}1&2&3\\2&1&3\end{pmatrix}$$

Умножим полученное уравнение справа на $\begin{pmatrix}1&2&3\\2&3&1\end{pmatrix}^{-1}=\begin{pmatrix}1&2&3\\3&1&2\end{pmatrix}$. Получим $X=\begin{pmatrix}1&2&3\\1&3&2\end{pmatrix}$.

В полугруппе в общем случае законы сокращения и разрешимости уравнения не имеют место. Например, в полугруппе квадратных матриц $AX=AY$ может выполняться и для различных X,Y.

В полугруппе любой элемент можно умножать на себя произвольное число раз, причем в силу ассоциативности операции элемент $a^n=\underbrace{a\circ a\circ\ldots\circ a}_{n\text{ раз}}$ определен однозначно. Отрицательные степени могут быть определены как $a^{-n}=(a^{-1})^n$.

Теорема 2.5. *Для любой полугруппы*

$$a^m\circ a^n=a^{m+n},$$
$$(a^m)^n=a^{mn}.$$

Теорема 2.6. *Для любой группы*

$$a^{-n}=(a^n)^{-1},$$
$$a^m\circ a^n=a^{m+n},$$
$$(a^m)^n=a^{mn}.$$

Определение 2.8. Полугруппу (в частности, группу) называют *циклической*, если существует такой элемент a, что любой элемент полугруппы является некоторой целой степенью элемента a, называемого образующим элементом полугруппы (группы).

Пример 2.11. Полугруппа $(\mathbb{N}_0,+,0)$ является циклической с образующим элементом 1. Всякий элемент может быть получен возведением 1 в некоторую степень n, что обозначают как $n\cdot 1$. Полугруппа $(\mathbb{Z},+,0)$ также является циклической. Образующими элементами могут быть 1 и -1. Тогда $0=0\cdot 1, n\cdot 1=\underbrace{1+1+1+\ldots+1}_{n\text{ раз}}=n>0, (-n)\cdot 1=n\cdot(-1)=\underbrace{(-1)+(-1)+(-1)+\ldots+(-1)}_{n\text{ раз}}=-n<0$. Группа $(\mathbb{Z}_3,\oplus_3,0)$ является циклической, причем любой ее ненулевой элемент является образующим.

Определение 2.9. *Порядком конечной группы* называется число элементов в этой группе. *Порядком элемента* в группе называется наименьшее положительное n, такое что $a^n = \mathbf{1}$.

Теорема 2.7. *Порядок образующего элемента конечной циклической группы равен порядку самой группы.*

Доказательство. Пусть $\mathcal{G} = (G, \circ)$ — конечная циклическая группа с образующим элементом a, порядок которого равен n. Рассмотрим бинарное отношение $\rho = \{(x, y) \in G^2 | y = x \circ a\}$. Согласно теореме 2.4, $\forall y \in G \exists! x \in G : y = x \circ a$, т.е. $\forall y \in G : \exists!(x, y) \in \rho$. Рассмотрим последовательность $L = (\mathbf{1}, a, a^2, \ldots)$. Ясно, что $\mathbf{1}\,\rho a \rho a^2 \rho \ldots$. По определению образующего элемента, $\forall x \in G \exists k \geq 0 : \mathbf{1}\,\rho^k x$. В соответствии с теоремой 1.2, если некоторая пара $(x, y) \in \rho^k$, то она также принадлежит $\rho^l, l \leq |G| - 1$. Таким образом, все элементы G встречаются среди первых $|G|$ членов последовательности L ровно один раз, т.е. $n \geq |G|$. Предположим, что $n > |G|$. Но тогда $a^{|G|} = a^k, k > 0 \Rightarrow a^{|G|-k} = \mathbf{1}$, т.е. в последовательности L несколько раз встречается $\mathbf{1}$. Из полученного противоречия следует, что $|G| = n$. □

Из этой теоремы следует, что в бесконечной циклической группе не существует такого $n > 0 : a^n = \mathbf{1}$.

2.2.2 Подалгебры

Определение 2.10. Подмножество $X \subset A$ называется *замкнутым* относительно операции ϕ, если

$$\forall x_1, \ldots, x_n \in A : \phi(x_1, \ldots, x_n) \in X.$$

Если X замкнуто относительно всех операций из Ω, то (X, Ω_X) называется *подалгеброй*, где $\Omega_x = \{\phi_i^X | \phi_i^X = \phi_i|_{X^k}, \phi_i \in \Omega, k = n_i\}$, n_i указывает арность операции ϕ_i.

Теорема 2.8. *Непустое пересечение подалгебр образует подалгебру.*

Доказательство. Пусть (X_1, Ω_{X_1}) и (X_2, Ω_{X_2}) замкнуты. Следовательно, $\forall j \forall x_1, \ldots, x_{n_j} \in X_k : \phi_j^{X_k}(x_1, \ldots, x_{n_j}) \in X_k$, где $k = 1, 2$. Следовательно, $\forall j \forall x_1, \ldots, x_{n_j} \in X_1 \cap X_2 : \phi_j^{X_1 \cap X_2}(x_1, \ldots, x_{n_j}) \in X_1 \cap X_2$. □

Замыканием множества $X \subset A$ относительно сигнатуры Ω ($|X|_\Omega$) называется множество всех элементов, которые могут быть получены из X с помощью операций из Ω, в т.ч. элементы самого X. Множество $A' \subset A$ называется системой образующих алгебры (A, Ω), если $|A'|_\Omega = A$. Если алгебра имеет конечную систему образующих, то она называется *конечно порожденной*. Бесконечные алгебры могут иметь конечную систему образующих.

Пример 2.12. В алгебре целых чисел $(\mathbb{Z}, +, \cdot)$ замыканием числа 2 является множество четных чисел. Алгебра натуральных чисел $(\mathbb{N}, +)$ имеет конечную систему образующих $\{1\}$.

Если на множестве G определена единственная операция, то всякое его подмножество, замкнутое относительно нее, называется подгруппоидом исходного группоида $\mathcal{G}$. Если эта операция ассоциативна, то это свойство сохраняется и при ограничении на подмножество; в этом случае говорят о подполугруппе исходной полугруппы $\mathcal{G}$. Но если $\mathcal{G}$ является моноидом, не всякий его подгруппоид будет являться подмоноидом, т.к. при ограничении на подмножество может потеряться нейтральный элемент. Например, аддитивная группа целых чисел $(\mathbb{Z}, +)$ при ограничении на подмножество натуральных чисел лишается нейтрального элемента 0. Следовательно, полугруппа $(\mathbb{N}, +)$ даже не является моноидом. Если ограничение исходного моноида выполнено так, что сохранен нейтральный элемент, говорят о подмоноиде. Аналогично, если при ограничении группы на подмножество для каждого оставшегося элемента остался также обратный к нему, гворят о подгруппе. Заметим, что сохранение нейтрального элемента следует из замкнутости множества.

Определение 2.11. Подгруппу группы $\mathcal{G}$, заданную на множестве степеней фиксированного элемента $a \in \mathcal{G}$, называют *циклической подгруппой* группы $\mathcal{G}$, порожденной элементом a.

Пример 2.13. Рассмотрим группу $\mathbb{Z}_{13}^*$ (мультипликативную группу вычетов по модулю 13). Построим циклическую подгруппу, порожденную элементом 5:

$$5^0 = 1, 5^1 = 5, 5^2 \equiv 12 \bmod 5, 5^3 \equiv 8 \bmod 13, 5^4 \equiv 1 \bmod 13.$$

Таким образом, порядок этой циклической подгруппы равен 4.

Определение 2.12. Левым *смежным классом* подгруппы $\mathcal{H} = (H, \circ, \mathbf{1})$ по элементу $a \in G$ называется множество

$$aH = \{y | y = a \circ h, h \in H\}.$$

Правым смежным классом подгруппы $\mathcal{H} = (H, \circ, \mathbf{1})$ по элементу $a \in G$ называется множество

$$Ha = \{y | y = h \circ a, h \in H\}.$$

Ясно, что в коммутативной группе $aH = Ha$. При использовании аддитивной записи смежные классы обозначаются $H + a$. Заметим, что если $a \in H$, то $aH = H$.

Введем бинарное отношение

$$\sim_H = \{(a, b) \in G^2 | aH = bH\}.$$

Лемма 2.3. *Бинарное отношение $\sim_H$ есть эквивалентность на G, причем класс эквивалентности произвольного $a \in F$ совпадает со смежным классом aH.*

Доказательство. Т.к. $\forall a \in G : aH = aH$, отношение рефлексивно. Т.к. $a \sim_H b \Rightarrow aH = bH \Rightarrow bH = aH \Rightarrow b \sim_H a$, отношение симметрично. Т.к. $a \sim_H b \wedge b \sim_H c \Rightarrow$

$aH = bH \wedge bH = cH \Rightarrow aH = cH \Rightarrow a \sim_H c$, отношение транзитивно. Следовательно, $\sim_H$ является отношением эквивалентности.

Пусть $x \in [a]_{\sim_H} \Rightarrow x \sim_H a \Rightarrow xH = aH \Rightarrow \forall y = ah, h \in H : y = xh_1, h_1 \in H \Rightarrow ah = xh_1 \Rightarrow x = ahh_1^{-1}$. Т.к. $\mathcal{H}$ — подгруппа, $h_2 = hh_1^{-1} \in H \Rightarrow x = ah_2 \in aH \Rightarrow [a]_{\sim_H} \subset aH$. Пусть $x \in aH \Rightarrow x = ah, h \in H \Rightarrow xH = ahH$. Т.к. $hH = H$, справедливо $xH = aH \Rightarrow x \sim_H a \Rightarrow x \in [a]_{\sim_H} \Rightarrow aH \subset [a]_{\sim_H}$. □

Лемма 2.4. *Всякий левый смежный класс подгруппы $\mathcal{H}$ равномощен ей.*

Доказательство. Для произвольного $a \in G$ зададим отображение $\phi_a : H \longrightarrow aH$ следующим образом: $\phi_a(h) = ah$. Это отображение является сюръекцией, т.к. если $x \in aH$, то $\exists h \in H : x = ah = \phi_a(h)$. Кроме того, ϕ_a — инъекция, т.к. из равенства $ah_1 = ah_2$ в силу свойств группы $\mathcal{G}$ следует $h_1 = h_2$. Следовательно, ϕ_a — биекция и $|aH| = |H|$. □

Доказанные утверждения справедливы для любой группы. Но в случае конечных групп из них вытекает следующий важный результат.

Теорема 2.9 (Лагранжа)**.** *Порядок конечной группы делится на порядок любой ее подгруппы.*

Доказательство. Выберем произвольную подгруппу $\mathcal{H}$ группы $\mathcal{G}$. Согласно Лемме 2.3, все левые смежные классы образуют разбиение множества G на непересекающиеся подмножества, равномощные в силу Леммы 2.4 подгруппе $\mathcal{H}$. Т.к. группа $\mathcal{G}$ конечна, число элементов разбиения k также конечно. Следовательно, $|G| = k|H|$. Следовательно, порядок $\mathcal{G}$ делится на порядок произвольной ее подгруппы $\mathcal{H}$. □

Следствие 2.1. *Любая группа простого порядка является циклической.*

Доказательство. Рассмотрим подгруппы группы $\mathcal{G}$, порядок которой является простым числом. Рассмотрим циклическую подгруппу $\mathcal{H}$, образующий элемент которой отличен от нейтрального. Тогда эта подгруппа содержит по крайней мере два элемента. Но т.к. ее порядок должен быть делителем $|\mathcal{G}|$, $|\mathcal{H}| = |\mathcal{G}|$. Следовательно, $\mathcal{G}$ является циклической. □

Необходимо отметить, что группа, порядок которой не является простым числом, также может быть циклической. Примером является $\mathbb{Z}_4^+$ — аддитивная группа вычетов целых чисел по модулю 4 с образующим элементом 1.

Группу называют *неразложимой*, если она не имеет нетривиальных подгрупп. Тривиальными подгруппами является подгруппа, состоящая из одного нейтрального элемента, и подгруппа, совпадающая со всей группой.

Следствие 2.2. *Конечная группа неразложима тогда и только тогда, когда она является циклической группой, порядок которой есть простое число.*

Доказательство. Если порядок циклической группы — простое число, то, согласно теореме Лагранжа, каждая ее подгруппа имеет порядок равный или 1, или порядку всей группы, т.е. группа неразложима.

Пусть конечная группа $\mathcal{G} = (G, \cdot, \mathbf{1})$ неразложима, Выберем элемент $a \in G : a \neq \mathbf{1}$. Тогда циклическая подгруппа с образующим элементом a совпадает с $\mathcal{G}$. Предположим, что $|G|$ — составное число, т.е. $|G| = kl, k, l \neq 1$. Тогда циклическая подгруппа с образующим элементом $b = a^k$ не совпадает с $\mathcal{G}$, т.к. $b^l = a^{kl} = \mathbf{1}$, а в соответствии с теоремой 2.7, порядок образующего элемента совпадает с порядком подгруппы. Из полученного противоречия следует, что $|G|$ является простым числом. □

Следствие 2.3. *В конечной группе $\mathcal{G}$ для любого элемента $a \in G$ имеет место* $a^{|G|} = \mathbf{1}$

Доказательство. Если группа $\mathcal{G}$ циклическая и a — ее образующий элемент, то утверждение непосредственно следует из теоремы 2.7. Если элемент a является образующим некоторой циклической подгруппы $\mathcal{H}$ порядка k, то $|G| = kl$ и $a^k = \mathbf{1}$, откуда следует, что $a^{|G|} = \mathbf{1}$ □

Пример 2.14. Малая теорема Ферма: если $(n, p) = 1$, то

$$n^{p-1} \equiv 1 \bmod p.$$

Действительно, пусть $n = rp + k, 0 < k < p$. Ясно, что $n^{p-1} \equiv k^{p-1} \bmod p$. Рассмотрим мультипликативную группу $\mathbb{Z}_p^*$ вычетов по модулю p. Порядок этой группы равен $p-1$. В соответствии со следствием 2.3, $\forall a \in \mathbb{Z}_p^* : a^{|\mathbb{Z}_p^*|} = \mathbf{1}$, т.е. $\forall k : 0 < k < p : k^{p-1} = 1 \bmod p$.

Пример 2.15. Рассмотрим аддитивную группу вычетов целых чисел по модулю 15. В этой группе могут быть выделены следующие подгруппы:

1. $\{0\}$ — образующий элемент 0.
2. $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14\}$ — образующий элемент 1.
3. $\{0, 5, 10\}$ — образующий элемент 5.
4. $\{0, 3, 6, 9, 12\}$ — образующий элемент 3.

2.2.3 Кольца, тела, поля

Определение 2.13. *Кольцом* называется алгебра

$$\mathcal{R} = (R, +, \cdot, \mathbf{0}, \mathbf{1}),$$

сигнатура которой состоит из двух бинарных и двух нульарных операций, для которых выполняются равенства (аксиомы кольца):

1. $a + (b + c) = (a + b) + c$;
2. $a + b = b + a$;
3. $a + \mathbf{0} = a$;
4. $\forall a \in R : \exists a' : a + a' = 0$;
5. $a \cdot (b \cdot c) = (a \cdot b) \cdot c$;
6. $a \cdot \mathbf{1} = \mathbf{1} \cdot a = a$;
7. $a \cdot (b + c) = a \cdot b + a \cdot c, (b + c) \cdot a = b \cdot a + c \cdot a$.

Операции $+$ и $\cdot$ называются сложением и умножением, $\mathbf{0}$ и $\mathbf{1}$ — нулем и единицей кольца. Можно заметить, что алгебра $(R, \mathbf{0}, +)$ является абелевой группой, называемой аддитивной группой кольца $\mathcal{R}$. При этом обратный элемент $a' : a + a' = \mathbf{0}$ обозначают как $-a$, а операцию сложения с ним называют вычитанием $a+(-b) = a-b$. С другой стороны, алгебра $(R, \mathbf{1}, \cdot)$ является мультипликативным моноидом кольца. Связь между сложением кольца и умножением кольца устанавливается аксиомой дистрибутивности. Кольцо называется коммутативным, если его операция умножения коммутативна.

Пример 2.16. Алгебра $(\mathbb{Z}, +, \cdot, 0, 1)$ является коммутативным кольцом. При этом $(\mathbb{N}_0, +, \cdot, 0, 1)$ кольцом не является, т.к. $(\mathbb{N}_0, +)$ — коммутативный моноид, но не группа.

Пример 2.17. Множество всех квадратных матриц фиксированного порядка с операциями сложения и умножения является некоммутативным кольцом.

Теорема 2.10. *В любом кольце выполняются следующие тождества:*

1. $\mathbf{0} \cdot a = a \cdot \mathbf{0} = \mathbf{0}$
2. $(-a) \cdot b = -(a \cdot b) = a \cdot (-b)$
3. $(a - b) \cdot c = a \cdot c - b \cdot c, c \cdot (a - b) = c \cdot a - c \cdot b$

Доказательство. 1. $a + \mathbf{0} \cdot a = \mathbf{1} \cdot a + \mathbf{0} \cdot a = (\mathbf{1} + \mathbf{0}) \cdot a = \mathbf{1} \cdot a = a \Rightarrow \mathbf{0} \cdot a = a - a = \mathbf{0}$.
Равенство $a \cdot \mathbf{0} = \mathbf{0}$ доказывается аналогично.

2. $a \cdot (-b) + a \cdot b = a \cdot ((-b) + b) = a \cdot \mathbf{0} = \mathbf{0} \Rightarrow a \cdot (-b) = -(a \cdot b)$
3. $a \cdot (b - c) = a \cdot (b + (-c)) = a \cdot b + a \cdot (-c) = a \cdot b - a \cdot c$.

□

Ненулевые элементы a, b кольца называются *делителями нуля*, если $a \cdot b = \mathbf{0} \vee b \cdot a = \mathbf{0}$. Примерами кольца с делителями нуля являются кольцо вычетов целых чисел по модулю составного числа ($2 * 3 \equiv 0 \bmod 6$) и кольцо квадратных матриц порядка не ниже 2. Коммутативное кольцо без делителей нуля называется *областью целостности*.

Определение 2.14. *Полукольцом* называется алгебра

$$\mathcal{S} = (S, +, \cdot, \mathbf{0}, \mathbf{1}),$$

такая, что для произвольных элементов a, b, c множества S выполняются следующие равенства (аксиомы полукольца):

1. $a + (b + c) = (a + b) + c$;
2. $a + b = b + a$;
3. $a + \mathbf{0} = a$;
4. $a \cdot (b \cdot c) = (a \cdot b) \cdot c$;
5. $a \cdot \mathbf{1} = \mathbf{1} \cdot a = a$;
6. $a \cdot (b + c) = a \cdot b + a \cdot c$;
7. $(b + c) \cdot a = b \cdot a + c \cdot a$;
8. $a \cdot \mathbf{0} = \mathbf{0} \cdot a = \mathbf{0}$;

Видно, что кольцо — частный случай полукольца. Если кольцо является по сложению абелевой группой, то полукольцо — лишь коммутативный моноид.

Пример 2.18. Алгебра логики $(\{0, 1\}, \vee, \wedge)$ является полукольцом.

Поставим вопрос: в каких случаях кольцо по умножению будет группой? Ясно, что все элементы кольца, в котором $\mathbf{0} \neq 1$, не могут образовывать группу по умножению, т.к. $\mathbf{0}$ не имеет обратного. Действительно, если существует $\mathbf{0}^{-1} : \mathbf{0} \cdot \mathbf{0}^{-1}$, то $\mathbf{0} = \mathbf{0} \cdot \mathbf{0}^{-1} = \mathbf{1}$. Таким образом, нулевой элемент не может входить в мультипликативную группу. Если в кольце имеются делители нуля, то подмножество ненулевых элементов не может быть группой по умножению, т.к. оно не замкнуто.

Определение 2.15. Кольцо, в котором множество всех ненулевых элементов образует группу по умножению, называется *телом*. Коммутативное тело называется *полем*. Группа ненулевых элементов тела (поля) по умножению называется *мультипликативной группой* тела (поля).

Таким образом, в любом поле выполняются следующие тождества:

1. $a + (b + c) = (a + b) + c$;
2. $a + b = b + a$;
3. $a + \mathbf{0} = a$;
4. $\forall a \in R : \exists a' : a + a' = 0$;

5. $a \cdot (b \cdot c) = (a \cdot b) \cdot c$;

6. $a \cdot b = b \cdot a$; в случае тела имеет место лишь $a \cdot \mathbf{1} = \mathbf{1} \cdot a = a$;

7. $a \cdot (b + c) = a \cdot b + a \cdot c, (b + c) \cdot a = b \cdot a + c \cdot a$.

8. $\forall a \neq \mathbf{0}\, \exists a^{-1} : a \cdot a^{-1} = a^{-1} \cdot a = \mathbf{1}$

Пример 2.19. Алгебра $(\mathbb{Q}, \cdot, +, 0, 1)$ называется полем рациональных чисел. Алгебра $(\mathbb{R}, \cdot, +, 0, 1)$ называется полем вещественных чисел.

Теорема 2.11. *Конечная область целостности является полем.*

2.3 Основные результаты

1. Теорема 2.1: отношение изоморфизма на множестве однотипных алгебр является эквивалентностью.

2. Теорема 2.7: Порядок образующего элемента конечной циклической группы равен порядку самой группы.

3. Теорема 2.9 (Лагранжа): порядок конечной группы делится на порядок любой ее подгруппы.

4. Следствие 2.3 из теоремы Лагранжа: в конечной группе $\mathcal{G}$ для любого элемента $a \in G$ имеет место $a^{|G|} = \mathbf{1}$

Упражнения

1. Ассоциативна ли операция $\odot$ на множестве M, если

   (a) $M = \mathbb{N}, x \odot y = 2xy$;

   (b) $M = \mathbb{Z}, x \odot y = x^2 + y^2$;

   (c) $M = \mathbb{R}, x \odot y = \sin x \sin y$;

   (d) $M = \mathbb{R} \setminus \{0\}, x \odot y = xy^{x/|x|}$;

   (e) $M = \mathbb{R}, x \odot y = x - y$?

2. Пусть $\mathcal{A} = (A, \cdot), \mathcal{B} = (B, *)$ — некоторые алгебры с бинарными операциями. Тогда алгебра $\mathcal{C} = (C, *) = \mathcal{A} \times \mathcal{B}$ называется прямым произведением $\mathcal{A}$ и $\mathcal{B}$, если $C = A \times B$ и операция $*$ определена как $(a_1, b_1) * (a_2, b_2) = (a_1 \cdot a_2), b_1 * b_2)$.

   Пусть $\mathcal{C} = \mathcal{A} \times \mathcal{A}$. Доказать, что существуют гомоморфизмы $\alpha : \mathcal{A} \longrightarrow \mathcal{C}$, $\beta : \mathcal{C} \longrightarrow \mathcal{A}$, такие что $\alpha \circ \beta : \mathcal{A} \longrightarrow \mathcal{A}$ — тождественная функция.

3. Доказать, что множество линейных функций $L = \{ax+b | a, b \in \mathbb{R}\}$ с операцией линейной замены переменной, определенной как $(ax+b) \circ (cx+d) = a(cx+d)+b = (ac)x+(ad+b)$, образует группу.

4. На множестве M определена бинарная операция $\circ : x \circ y = x$. Доказать, что $(M, \circ)$ — полугруппа. Что можно сказать о ее нейтральных элементах? В каких случаях она является группой?

5. Доказать теорему 2.7, используя только определения группы и образующего элемента.

6. В группе S_4 решить уравнения

   (a) $\begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 2 & 1 & 3 \end{pmatrix} X \begin{pmatrix} 1 & 2 & 3 & 4 \\ 3 & 2 & 1 & 4 \end{pmatrix} = (1\,2)$

   (b) $(1\,2)(3\,4)X(1\,3) = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 4 & 2 & 1 & 3 \end{pmatrix}$

7. Является ли полем множество всех чисел вида $x+\sqrt{2}y, x, y \in \mathbb{Q}$ с обычными операциями сложения и умножения?

Глава 3

Алгебраическая теория алгоритмов

Иногда это также называется получисленными алгоритмами [9].

3.1 Операции над матрицами

3.1.1 Умножение произвольных матриц

Рассмотрим задачу вычисления $Z = XY, X \in \mathbb{K}^{m\times n}, Y \in \mathbb{K}^{n\times s}$, где $\mathbb{K}$ — некоторое кольцо. Известно, что

$$z_{ik} = \sum_{j=1}^{n} x_{ij}y_{jk}, i = 1..m, k = 1..s$$

Непосредственное вычисление требует mns умножений и $ms(n-1)$ сложений. *Шмуэль Виноград* предложил способ замены половины умножений сложениями (предполагается коммутативность умножения):

$$z_{ik} = \sum_{j=1}^{n/2}(x_{i,2j} + y_{2j-1,k})(x_{i,2j-1} + y_{2j,k}) - a_i - b_k + (x_{in}y_{nk}\,\mathbf{1}(n \text{ нечетно}))$$

$$a_i = \sum_{j=1}^{n/2} x_{i,2j}x_{i,2j-1};\ b_k = \sum_{j=1}^{n/2} y_{2j-1,k}y_{2j,k}. \qquad (3.1)$$

Этот метод требует $\lceil n/2\rceil ms + \lfloor n/2\rfloor(m+s)$ умножений и $(n+2)ms + (\lfloor n/2\rfloor - 1)(ms + m + s)$ сложений или вычитаний.

В общем случае число умножений может быть еще более уменьшено с помощью *алгоритма Штрассена*, который не требует коммутативности умножения. Пусть необходимо перемножить две 2×2 матрицы. Модификация Винограда алгоритма Штрассена:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} A & C \\ B & D \end{pmatrix} = \begin{pmatrix} aA + bB & w + v + (a + b - c - d)D \\ w + u + d(B + C - A - D) & w + u + v \end{pmatrix},$$
$$u = (c - a)(C - D), v = (c + d)(C - A), w = aA + (c + d - a))(A + D - C), \quad (3.2)$$

требует всего 7 умножений и 15 сложений. Этот подход может быть применен рекурсивно, что дает асимптотическое число умножений $O(n^{\log_2 7}) = O(n^{2.8074})$.

3.1.2 Умножение двоичных матриц

Рассмотрим задачу умножения двоичных $n \times n$ матриц над полукольцом $(\{0,1\}, \vee, \wedge)$. Алгоритм Штрассена не может быть непосредственно применен к решению этой задачи, т.к. он требует, чтобы множество с операцией сложения образовывали абелеву группу. Тем не менее, множество $(\{0,1\}$ может быть *погружено в кольцо* целых чисел по модулю $n+1$ с заменой операций $\vee \to +, \wedge \to *$. После этого умножение может быть выполнено с помощью алгоритма Штрассена над кольцом. Затем результат должен быть отображен в бинарное множество путем замены всех ненулевых элементов полученной матрицы на 1.

Можно также построить специализированный алгоритм перемножения двоичных матриц, который, однако, асимптотически менее эффективен. Разобьем матрицы на $n/\log_2 n$ групп строк и столбцов: $X = (X_1|X_2|\ldots|X_{n/\log_2 n})$, $Y = \begin{pmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_{n/\log_2 n} \end{pmatrix}$. Тогда $XY = \bigvee_{i=1}^{n/\log_2 n} X_i Y_i$. Заметим, что каждая строка матрицы X_i содержит $\log_2 n$ элементов, равных 0 или 1. Наличие 1 в строке означает, что соответствующая строка матрицы Y_i должна быть включена в дизъюнкцию, т.е. каждая строка матрицы X_i задает некоторое подмножество строк Y_i. Эти подмножества можно перебирать таким образом, чтобы очередное подмножество отличалось от *какого-либо из предыдущих добавлением одного* элемента. Таким образом, произведение $X_i Y_i$ может быть вычислено не более чем за n^2 операций, а общая сложность составляет $O(n^3/\log_2 n)$ операций. Естественно, что этот метод может быть использован и в том случае, когда Y является вектором-столбцом. Данный алгоритм был разработан В.Л. Арлазаровым, Е.А. Диницем, М.А. Кронродом и И.А. Фараджевым и в зарубежной литературе носит название *алгоритма четырех русских*.

Если операцию $\vee$ заменить на $\oplus$ (т.е. сложение по модулю 2), то рассматриваемая алгебра станет кольцом (на самом деле полем $GF(2)$). Это дает возможность рассматривать последовательность строк матриц X_i, различающихся друг от друга *добавлением или удалением* ровно одного элемента, т.е. строки могут быть переупорядочены в соответствии с кодом Грея.

3.2 Операции над многочленами

3.2.1 Вычисление значений многочленов

Предположим, что задан некоторый многочлен $u(x) = u_n x^n + u_{n-1} x^{n-1} + \ldots + u_0$ и необходимо вычислить его значение в одной или нескольких точках. В том случае, когда подавляющее большинство коэффициентов равны нулю, вычисления могут выполняться непосредственно, причем могут быть использованы быстрые алгоритмы возведения в степень. Однако в общем случае намного более эффективным методом является схема Горнера:

$$u(x) = (\ldots((u_n x + u_{n-1})x + u_{n-2})x + \ldots)x + u_0 \tag{3.3}$$

Таким образом, требуется всего n умножений и n сложений.

В том случае, когда точки, в которых вычисляется значение многочлена, известны заранее, сложность может быть уменьшена за счет использования каких-либо их специальных свойств.

3.2.2 Билинейные формы

Рассмотрим задачу вычисления

$$z_k = \sum_{i=1}^{m} \sum_{j=1}^{n} t_{ijk} x_i y_j, 1 \le k \le s,$$

где t_{ijk} — некоторые коэффициенты. Это выражение называется *билинейной формой*, задаваемой тензором (t_{ijk}). Несложно заметить, что оно линейно относительно y и z. Ограничимся нормальными вычислительными схемами, в которых все умножения производятся между линейными комбинациями x и линейными комбинациями y. Таким образом, мы строим r произведений

$$w_l = (a_{1l}x_1 + a_{2l}x_2 + \ldots + a_{ml}x_m)(b_{1l}y_1 + b_{2l}y_2 + \ldots + b_{ml}y_m), l = 1..r,$$

а результат вычисляем как

$$z_k = \sum_{l=1}^{r} c_{kl} w_l.$$

Сопоставляя эти выражения, получим, что нормальная схема вычислений корректна тогда и только тогда, когда

$$t_{ijk} = \sum_{l=1}^{r} a_{il} b_{jl} c_{kl}.$$

Заметим, что алгоритм Штрассена перемножения матриц не является нормальной вычислительной схемой, т.к. там x и y в ходе вычислений смешиваются. Ненулевой

тензор называют тензором ранга 1, если $\exists(a_1,\ldots,a_m),(b_1,\ldots,b_n),(c_1,\ldots,c_s):\forall i,j,k:$ $t_{ijk}=a_ib_jc_k$. Рангом тензора называется такое минимальное число r, что (t_{ijk} выражается в виде суммы r тензоров ранга 1. Ранг тензора есть минимальное число умножений в цепочке при нормальном вычислении соответствующей билинейной формы.

Ограничимся рассмотрением задач вычисления линейной и циклической свертки. *Линейной сверткой* многочленов (или векторов их коэффициентов) $a(x)=\sum_{i=0}^{n-1}a_ix^i$ и $b(x)=\sum_{i=0}^{n-1}b_ix^i$ называется многочлен (или вектор его коэффициентов)

$$c(x)=a(x)b(x)=\sum_{i=0}^{2n-2}x^i\sum_{j=0}^{n-1}a_jb_{i-j}.$$

Циклической сверткой многочленов (или векторов их коэффициентов) называется многочлен (или вектор его коэффициентов)

$$c(x)=a(x)b(x)\pmod{x^n-1}=\sum_{i=0}^{n-1}x^i\sum_{j=0}^{n-1}a_jb_{((i-j))},$$

где $((i))\equiv i \bmod n$.

3.2.3 Алгоритмы Карацубы и Тоома-Кука вычисления свертки

Рассмотрим задачу вычисления линейной свертки $c_0+c_1x+c_2x^2=(a_0+a_1x)(b_0+b_1x)$. Результат является многочленом второй степени и однозначно определяется своими значениями в трех различных точках. В случае вещественных чисел в качестве таких точек удобно выбрать $c(-1),c(0),c(1)$. Тогда $c(-1)=a(-1)b(-1)=(a_0-a_1)(b_0-b_1),c(0)=a(0)b(0)=a_0b_0,c(1)=a(1)b(1)=(a_0+a_1)(b_0+b_1)$. Следовательно, $c_(x)=c(0)(1-x^2)+\frac{1}{2}c(1)(x^2+x)+\frac{1}{2}c(-1)(x^2-x)$. Константы $\frac{1}{2}$ удобно внести в выражения для $c(1)$ и $c(-1)$. *Алгоритм быстрого умножения Тоома-Кука* может быть записан как

$$\begin{pmatrix}c_0\\c_1\\c_2\end{pmatrix}=\begin{pmatrix}1&0&0\\0&1&-1\\-1&1&1\end{pmatrix}\left(\begin{pmatrix}1&0\\1/2&1/2\\1/2&-1/2\end{pmatrix}\begin{pmatrix}a_0\\a_1\end{pmatrix}\cdot\begin{pmatrix}1&0\\1&1\\1&-1\end{pmatrix}\begin{pmatrix}b_0\\b_1\end{pmatrix}\right).$$

Видно, что данный алгоритм требует 3 умножений и 7 сложений. Во многих случаях многочлен $a(x)$ является фиксированным, что позволяет вычислить $a(0),a(1),a(-1)$ заранее. В некоторых случаях одно из этих значений оказывается нулевым, что позволяет дополнительно сократить число операций.

Рассмотрим модификацию алгоритма Тоома-Кука, демонстрирующую важный прием, используемый при построении быстрых алгоритмов. Заметим, что $c_{2n-2}=a_{n-1}b_{n-1}$. Тогда $c'(x)=a(x)b(x)-x^{2n-2}a_{n-1}b_{n-1}$ имеет степень, на единицу меньшую, чем $c(x)$. Для случая линейной 2×2 свертки это приводит к $c'(0)=a(0)b(0)=a_0b_0$, $c'(1)=a(1)b(1)-a_1b_1=(a_0+a_1)(b_0+b_1)-a_1b_1$. Таким образом, $c_0=c'_0=a_0b_0,c_1=$

$c_1' = (a_0 + a_1)(b_0 + b_1) - a_1 b_1 - a_0 b_0, c_2 = a_1 b_1$. Описанный *алгоритм Карацубы* может быть записан в следующем виде:

$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix} \left(\begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \end{pmatrix} \right).$$

Таким образом, требуется 3 умножения и 4 сложения. Этот алгоритм может быть легко адаптирован для вычисления циклической свертки:

$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 \\ -1 & 1 & -1 \end{pmatrix} \left(\begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \end{pmatrix} \right).$$

Основная идея алгоритма Тоома-Кука и его аналогов состоит в выборе таких интерполяционных точек, вычисление значений многочленов (и последующая интерполяция) в которых сводятся к элементарным действиям типа сложения и вычитания. В случае поля вещественных чисел эту идею можно обобщить на случай многочленов большей степени, используя точки $\pm 2, \pm 3$ и т.д., причем умножения на эти числа должны заменяться на сложения. Однако достаточно быстро этот метод становится чрезмерно громоздким. Более того, не во всяких полях оказывается возможным найти достаточное количество удобных для вычислений точек, необходимых для построения алгоритма. В связи с этим, как правило, практикуется использование т.н. гнездовых методов.

3.2.4 Алгоритм Винограда

Предположим, что необходимо вычислить

$$c(x) = a(x)b(x) \bmod m(x).$$

Алгоритмы Тоома-Кука и Карацубы основывались на вычислении значений многочленов $a(x)$ и $b(x)$ в наборе точек, перемножении этих значений и восстановлении результата с помощью интерполяции. Вычисление значений многочлена в отдельных точках можно представить как $a(x_i) \equiv a(x) \bmod (x - x_i)$. Это позволяет рассматривать интерполяцию как частный случай формулы обращения китайской теоремы об остатках.

Теорема 3.1 (Китайская теорема об остатках для многочленов)**.** *Пусть $m^{(1)}(x), \ldots, m^{(r)}(x)$ — многочлены от одной переменной x с коэффициентами из некоторого поля. Пусть $m(x) = m^{(1)}(x) \cdot \ldots \cdot m^{(r)}(x)$ и пусть также $u^{(1)}(x), \ldots, u^{(r)}(x)$ — некоторые многочлены. Тогда существует ровно один многочлен $u(x)$:*

$$(0 \le \deg u(x) < \deg m(x)) \wedge (u(x) \equiv u^{(j)}(x) \bmod m^{(j)}(x), j = 1..r,$$

причем

$$u(x) = ((u^{(1)}(x)M_1(x) + \ldots + u^{(r)}(x)M_r(x)) \bmod m(x)), \tag{3.4}$$

где $M_j(x) = n^{(j)}(x)((n^{(j)}(x))^{-1} \bmod m^{(j)}(x)), n^{(j)}(x) = m(x)/m^{(j)}(x)$.

Пусть $m(x) = \prod_{j=1}^{r} m^{(j)}(x)$. Вычислим $a^{(j)}(x) \equiv a(x) \bmod m^{(j)}(x), b^{(j)}(x) \equiv b(x) \bmod m^{(j)}(x)$. Искомый многочлен может быть восстановлен по $c^{(j)}(x) \equiv a^{(j)}(x)b^{(j)}(x) \bmod m^{(j)}(x)$. Снижение сложности достигается в том случае, когда коэффициенты многочленов $m^{(j)}(x)$ являются "вычислительно простыми". Например, это могут быть небольшие по модулю целые или рациональные числа. Данный метод носит название *алгоритма Винограда*.

В том случае, когда необходимо вычислить циклическую свертку, многочлен $m(x)$ однозначно задан и равен $x^n - 1$. В случае необходимости вычисления линейной свертки многочлен $m(x)$ должен быть выбран таким образом, чтобы приведение по модулю не влияло на результат, т.е. он должен иметь степень не менее $\deg a(x) + \deg b(x) + 1$. Это оставляет достаточно большую свободу для выбора, причем вычислительные алгоритмы, построенные на основе различных многочленов, как правило, имеют несколько различающуюся сложность.

Матрицы билинейной формы для алгоритма Винограда формируются на основе матриц вычисления остатка от деления на многочлены $m^{(j)}(x)$, матриц вычисления произведений $c^{(j)}(x)$ и матрицы, соответствующей операции восстановления результата по формуле (3.4).

Пример 3.1. Рассмотрим построение быстрого алгоритма линейной свертки 3-точечного вектора (т.е. многочлена второй степени) с двухточечным. Пусть $a(x) = a_1x + a_0, b(x) = b_2x^2 + b_1x + b_0$. Непосредственное вычисление $c(x) = a(x)b(x)$ требует 6 умножений и 2 сложений. Пусть $m(x) = x(x-1)(x^2+1) = x^4 - x^3 + x^2 - x$. Вычеты равны

$$\begin{array}{rr} a^{(1)}(x) = a_0 & b^{(1)}(x) = b_0 \\ a^{(2)}(x) = a_1 + a_0 & b^{(2)}(x) = b_2 + b_1 + b_0 \\ a^{(3)}(x) = a_1x + a_0 & b^{(3)}(x) = b_1x + (b_0 - b_2) \end{array}$$

Следовательно,

$$\begin{aligned} c^{(1)}(x) &= a_0b_0 \\ c^{(2)}(x) &= (a_1 + a_0)(b_2 + b_1 + b_0) \\ c^{(3)}(x) &\equiv (a_1x + a_0)(b_1x + (b_0 - b_2)) \bmod (x^2 + 1) \end{aligned}$$

Вычисление первых двух вычетов требует двух операций умножения. Вычисление третьего вычета совпадает со структурой умножения комплексных чисел. Непосредственное его вычисление может быть выполнено как

$$c^{(3)}(x) = x(a_0^{(3)}b_1^{(3)} + a_1^{(3)}b_0^{(3)}) + (a_0^{(3)}b_0^{(3)} - a_1^{(3)}b_1^{(3)}).$$

Однако вычисления могут быть упрощены, если воспользоваться аналогом алгоритма Карацубы. Вычислим $a^{(3)}(x)b^{(3)}(x) - x^2a_1^{(3)}b_1^{(3)} = a_0^{(3)}b_0^{(3)} + x((a_0^{(3)} + a_1^{(3)})(b_0^{(3)} + b_1^{(3)}) - a_1^{(3)}b_1^{(3)} - a_0^{(3)}b_0^{(3)})$. Дополняя это выражение членом, образующимся при приведении по модулю $x^2 + 1$, получим $c^{(3)}(x) = a_0^{(3)}b_0^{(3)} - a_1^{(3)}b_1^{(3)} + x((a_0^{(3)} + a_1^{(3)})(b_0^{(3)} + b_1^{(3)}) -$

$a_1^{(3)}b_1^{(3)} - a_0^{(3)}b_0^{(3)}$). Окончательный результат должен быть восстановлен в соответствии с китайской теоремой об остатках:

$$c(x) = -(x^3-x^2+x-1)c^{(1)}(x)+\frac{1}{2}(x^3+x)c^{(2)}(x)+\frac{1}{2}(x^3-2x^2+x)c^{(3)}(x) \bmod (x^4-x^3+x^2-x)$$

Это равенство может быть перезаписано как

$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 1 & 1 \\ 1 & 0 & -2 & 0 \\ -1 & 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} c_0^{(1)} \\ \frac{1}{2}c_0^{(2)} \\ \frac{1}{2}c_0^{(3)} \\ \frac{1}{2}c_1^{(3)} \end{pmatrix}$$

Окончательно собирая алгоритм, получим

$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & 1 & 1 \\ 1 & 0 & -2 & 0 \\ -1 & 1 & 1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & -1 & 1 & -1 \end{pmatrix} \left(\begin{pmatrix} 1 & 0 \\ 1/2 & 1/2 \\ 1/2 & 0 \\ 1/2 & 1/2 \\ 0 & 1/2 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & -1 \\ 1 & 1 & -1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix} \right)$$

Перемножая матрицы постсложений, получим

$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 1 & -2 \\ 1 & 0 & -2 & 0 & 2 \\ -1 & 1 & 2 & -1 & 0 \end{pmatrix} \left(\begin{pmatrix} 1 & 0 \\ 1/2 & 1/2 \\ 1/2 & 0 \\ 1/2 & 1/2 \\ 0 & 1/2 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & 0 & -1 \\ 1 & 1 & -1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix} \right)$$

Полученный алгоритм можно улучшить, выбрав многочлен $m(x)$ несколько меньшей степени, чем это требуется, аналогично тому, как было сделано при построении алгоритма Карацубы. С помощью этого приема можно уменьшить число умножений и сложений.

Реализация алгоритма Винограда требует наличия эффективных процедур вычисления произведений многочленов по модулю неприводимых многочленов, т.е. $c^{(k)}(x) = a^{(k)}(x)b^{(k)}(x) \bmod m^{(k)}(x)$. Наиболее прямой метод состоит в вычислении линейной свертки и последующем приведении ее по модулю $m^{(k)}(x)$. Если степень испходных многочленов равна $n-1$, это требует не менее $2n-1$ умножений. Это позволяет использовать ранее построенные быстрые алгоритмы линейной свертки, модифицировав в них матрицы постсложений.

Какой-либо общей теории минимизации числа сложений не разработано, поэтому в этой части приходится полагаться на эвристические приемы. Применение алгоритма Винограда к многочленам больших степеней, как правило, приводит к большому числу сложений. Поэтому его целесообразно комбинировать с гнездовыми алгоритмами.

Всякий быстрый алгоритм представляет собой некоторое тождество, опирающееся на свойства ассоциативности, коммутативности и дистрибутивности, справедливые для того поля, над которым он был построен. В связи с этим построенный быстрый алгоритм может использоваться и в любом расширении заданного поля. Однако алгоритм, построенный специально для расширенного поля, может оказаться более эффективным.

Сформулируем некоторые фундаментальные свойства операций над многочленами:

1. Никакой алгоритм вычисления линейной свертки двух многочленов длин L и N не может содержать число умножений, меньшее чем $L + N - 1$.

2. Если число простых делителей многочлена $p(x)$ равно t, никакой алгоритм вычисления произведения многочленов (достаточно большой степени) по модулю $p(x)$ не может содержать число умножений, меньшее чем $2n - t$.

Хорошие алгоритмы линейных и циклических сверток табулированы [6, 5].

3.2.5 Перенос алгоритмов на поля другой природы

В некоторых случаях удается воспользоваться алгоритмами, построенными для полей иной природы. Например, рассмотрим задачу вычисления свертки (линейной или циклической), возникающую во многих задачах цифровой обработки сигналов. В большинстве сигнальных процессоров используется представление вещественных чисел с фиксированной запятой, т.е. в виде целых чисел. Если можно гарантировать, что результат выполнения операции не превысит некоторого достаточно большого значения p', то можно найти наименьшее простое число $p > p'$ и выполнять все вычисления по модулю p, т.е. в поле $GF(p)$, для которого могут существовать более эффективные вычислительные алгоритмы, чем для поля вещественных или кольца целых чисел.

Если построен алгоритм линейной или циклической свертки над полем вещественных чисел, то в некоторых случаях его удается адаптировать для конечного поля $GF(p^m)$. Пусть $\mathcal{S}(a, b)$ — билинейная форма, соответствующая определению свертки (которое не зависит от поля), а $\mathcal{F}(a, b)$ — билинейная форма, задающая некоторый быстрый алгоритм. Домножим тождество $\mathcal{F}(a, b) = \mathcal{S}(a, b)$ на минимальное число L так, чтобы избавиться от знаменателей в $\mathcal{F}(a, b)$. Полученное тождество приведем по модулю p. Если $L \neq 0 \bmod p$, полученное выражение задает быстрый алгоритм свертки для поля $GF(p^m)$.

Пример 3.2. Рассмотрим алгоритм трехточечной циклической свертки над вещественными числами. Он имеет вид

$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & -1 \\ 1 & -1 & -1 & 2 \\ 1 & 0 & 1 & -1 \end{pmatrix} \left(\begin{pmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ 1 & 0 & -1 \\ 0 & 1 & -2 \\ \frac{1}{3} & \frac{1}{3} & -\frac{2}{3} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & -1 \\ 0 & 1 & -1 \\ 1 & 1 & -2 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix} \right)$$

Домножая это тождество на 3 и приводя результат по модулю 2, получим быстрый алгоритм циклической свертки для поля $GF(2^m)$:

$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix} \left(\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \end{pmatrix} \right)$$

С другой стороны, лучший алгоритм двухточечной циклической свертки над вещественным полем имеет вид

$$\begin{pmatrix} c_0 \\ c_1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \left(\begin{pmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & -\frac{1}{2} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \end{pmatrix} \right).$$

Т.к. он содержит четные знаменатели, перенос в поле характеристики два невозможен. Причина этого состоит в том, что в этом поле $x^2 - 1 = (x+1)^2$, что делает невозможным использование алгоритма Винограда. В связи с этим приходится использовать алгоритм с тремя умножениями:

$$\begin{pmatrix} c_0 \\ c_1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \left(\begin{pmatrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \end{pmatrix} \right).$$

В зависимости от свойств многочлена $x^n - 1$ сложность алгоритма циклической свертки над конечным полем может быть как меньше, так и больше чем в случае вещественных чисел.

Для построения быстрых алгоритмов вычислений над комплексным полем могут использоваться расширенные конечные поля $GF(p^2), p$ — простое число.

Вычисления в конечном поле можно вложить в вещественное, комплексное или какое-либо иное подходящее поле. Для этого представим элементы поля $GF(p^m)$ в виде многочленов по модулю неприводимого многочлена, т.е. $a_i = \sum_{j=0}^{m-1} a_{ij} z^j$. Тогда коэффициенты линейной свертка может быть представлена как $c_i = \sum_{k=0}^{n-1} a_k b_{i-k} = \sum_{k=0}^{n-1} \sum_{l=0}^{m-1} \sum_{l'=0}^{m-1} a_{kl} b_{i-k,l'} z^{l+l'} \pmod p \pmod{p(x)}$. Это выражение представляет собой двумерную свертку, которую можно вычислять вышеописанными методами.

Кроме того, быстрые алгоритмы справедливы в некоторых кольцах, образованных на основе соответствующих полей.

3.2.6 Гнездовые алгоритмы свертки

Двумерная свертка представляет собой операцию, задаваемую на паре двумерных таблиц. Примером использования двумерной свертки является операция двумерной фильтрации, используемая при обработке изображений. Двумерная линейная свертка таблицы данных $b_{k',k''}, k' = 0..N'-1, k'' = 0..N''-1$ и таблицы фильтра $a_{j',j''}, j' = 0..L'-1, j'' = 0..L''-1$ называется таблица значений $c_{i',i''} = \sum_{k'=0}^{N'-1} \sum_{k''=0}^{N''-1} a_{i'-k',i''-k''} b_{k',k''}$.

Это определение может быть обобщено как на случай циклической свертки, так и на случай больших размерностей.

Заметим, что каждую из таблиц можно представить или как многочлен с векторными коэффициентами, или как многочлен от двух переменных $a(x,y) = \sum_{j'=0}^{L'-1}\sum_{j''=0}^{L''-1} b_{j'j''}x^{j'}y^{j''} = \sum_{j'=0}^{L'-1} b_{j'}(y)x^{j'}$. Это дает возможность определить двумерную линейную свертку или как как произведение многочленов от двух переменных, или как линейную свертку многочленов от одной переменной:

$$c(x,y) = a(x,y)b(x,y)$$

. Аналогично, двумерная циклическая свертка может быть определена как

$$c(x,y) = a(x,y)b(x,y) \pmod{x^{n'}-1} \pmod{y^{n''}-1}.$$

Непосредственное вычисление линейной свертки требует $L'L''N'N''$ умножений, что абсолютно неприемлемо для реализации.

Быстрые алгоритмы, построенные для одномерных сверток над полями, могут быть применены и для вычисления многомерных сверток. Для этого необходимо упорядочить переменные некоторым образом и рассмотреть задачу вычисления свертки многочленов, скажем, от переменной x, коэффициентами которого являются многочлены от переменной y. Применение быстрого алгоритма вычисления свертки приведет к необходимости сложения и умножения коэффициентов этого многочлена. Т.к. они также являются многочленами, вычисление их сверток может быть снова выполнено с помощью соответствующих быстрых алгоритмов, но теперь уже умножения будут являться обычными умножениями над полем.

3.2.7 Итеративные алгоритмы

В некоторых случаях задачу вычисления одномерной свертки удается решить путем ее сведения к многомерной свертке. Это может быть полезно в тех случаях, когда применение специализированного алгоритма линейной свертки (например, Винограда) приводит к слишком большому числу сложений. *Алгоритм Агарвала-Кули* основывается на китайской теореме об остатках для целых чисел.

Пусть надо вычислить компоненты циклической свертки

$$c_i = \sum_{k=0}^{n-1} a_{((i-k))}b_k$$

. Предположим, что $n = n'n'', (n',n'') = 1$. Заменим индексы i,k на пары (i',i'') : $i' \equiv i \bmod n', i'' \equiv i \bmod n'', (k',k'') : k' \equiv k \bmod n', k'' \equiv k \bmod n''$. В соответствии с китайской теоремой об остатках, старые индексы можно восстановить по формулам $i = N''n''i' + N'n'i'' \bmod n, k = N''n''k' + N'n'k'' \bmod n, N'n' + N''n'' = 1$. Тогда циклическую свертку можно записать как

$$c_{N''n''i'+N'n'i''} = \sum_{k=0}^{n-1} a_{N''n''(i'-k')+N'n'(i''-k'')}b_{N''n''k'+N'n'k''}.$$

Определим вспомогательные переменные

$$a_{i',i''} = a_{N''n''i'+N'n'i''}, i' = 0..n'-1, i'' = 0..n''-1$$
$$b_{k',k''} = a_{N''n''k'+N'n'k''}, k' = 0..n'-1, k'' = 0..n''-1$$
$$c_{j',j''} = c_{N''n''j'+N'n'j''}, j' = 0..n'-1, j'' = 0..n''-1$$

Тогда циклическая свертка может быть записана как

$$c_{i',i''} = \sum_{k'=0}^{n'-1} \sum_{k''=0}^{n''-1} a_{((i'-k')),((i''-k''))} b_{k'k''}$$

Здесь индексы вычисляются по модулям n' и n''. Таким образом, получена двумерная циклическая свертка. Заметим, что до настоящего времени никакого снижения сложности не получено. Число умножений по прежнему составляет $(n'n'')^2$.Но если теперь воспользоваться каким-либо быстрым алгоритмом двумерной свертки (построенном на основе более коротких алгоритмов одномерной свертки), то сложность может существенно снизиться. Она будет равна

$$A(n) = n'A(n'') + M(n'')A(n')$$
$$M(n) = M(n')M(n''),$$

где $M(n)$ и $A(n)$ — число умножений и сложений, требуемое для вычисления n-точечной циклической свертки. Заметим, что

- Общее число умножений не зависит от порядка выбора чисел n', n''.
- Общее число сложений зависит от порядка их выбора.
- Использование элементарного одномерного вычислительного алгоритма с меньшим числом умножений может привести к снижению как общего числа умножэеений, так и общего числа сложений.

Пример 3.3. Рассмотрим построение алгоритма 6-точечной циклической свертки $c(x) = a(x)b(x) \bmod (x^6 - 1)$ над $GF(2^m)$. Перегруппируем коэффициенты: $a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5 \longrightarrow (a_0 + a_4z + a_2z^2) + y(a_3 + a_1z + a_5z^2)$. Воспользуемся одним из быстрых алгоритмов двумерной циклической свертки $u(y) = w(y)v(y) \bmod (y^2 - 1)$:

$$\begin{pmatrix} u_0 \\ u_1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \left(\begin{pmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} w_0 \\ w_1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} v_0 \\ v_1 \end{pmatrix} \right)$$

Подставляя перегруппированные многочлены, получим

$$\begin{pmatrix} c_0 + c_4z + c_2z^2 \\ c_3 + c_1z + c_5z^2 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \left(\begin{pmatrix} (a_0 + a_3) + (a_1 + a_4)z + (a_2 + a_5)z^2 \\ a_3 + a_1z + a_5z^2 \\ a_0 + a_4z + a_2z^2 \end{pmatrix} \cdot \begin{pmatrix} b_0 + b_4z + b_2z^2 \\ (b_0 + b_3) + (b_1 + b_4)z + (b_2 + b_5)z^2 \\ (b_0 + b_3) + (b_1 + b_4)z + (b_2 + b_5)z^2 \end{pmatrix} \right)$$

Воспользуемся теперь быстрым алгоритмом трехточечной циклической свертки $p(z) = q(z)r(z) \bmod (z^3 - 1)$

$$\begin{pmatrix} p_0 \\ p_1 \\ p_2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix} \left(\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} q_0 \\ q_1 \\ q_2 \end{pmatrix} \cdot \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_0 \\ r_1 \\ r_2 \end{pmatrix} \right)$$

В результате получим

$$\begin{pmatrix} c_0 \\ c_4 \\ c_2 \\ c_3 \\ c_1 \\ c_5 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix} \left(\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{pmatrix} \right),$$

Рассмотрим еще один способ сведения одномерной свертки к многомерной. Пусть необходимо вычислить $c(x) = a(x)b(x), \deg a(x) = MN - 1, \deg b(x) = ML - 1$. Преобразуем исходные многочлены следующим образом:

$$a(y, z) = \sum_{i=0}^{N-1} \left(\sum_{k=0}^{M-1} a_{Ml+k} y^k \right) z^i.$$

Преобразование $b(x)$ выполняется аналогично. Заметим, что $a(x) = a(x, x^M)$. Тогда $c(y, z) = a(y, z)b(y, z)$ может быть вычислено с помощью быстрого гнездового алгоритма и $c(x) = x(x, x^M)$. Число умножений в полученном алгоритме равно произведению

числа умножений двух используемых элементарных алгоритмов: линейной свертки последовательностей длины L и N и двух последовательностей длины M. В том случае, когда степени исходных многочленов не удается разложить указанным образом, они могут быть дополнены нулями до удобных значений. В этом случае необходимо проверить получившиеся матрицы пред- и постсложений на предмет наличия нулевых и повторяющихся столбцов и строк.

Пример 3.4. Рассмотрим вычисление линейной одномерной 4-точечной свертки, т.е. $c(x) = (a_0 + a_1x + a_2x^2 + a_3x^3)(b_0 + b_1x + b_2x^2 + b_3x^3)$ Это сводится к $c_0(y)z^0 + c_1(y)z^1 + c_2(y)z^2 = ((a_0 + a_1y)z^0 + (a_2 + a_3y)z^1)((b_0 + b_1y)z^0 + (b_2 + b_3y)z^1)$. Проитерируем дважды алгоритм Карацубы:

$$\begin{pmatrix} c_0(y) \\ c_1(y) \\ c_2(y) \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix} \left(\begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} a_0 + a_1y \\ a_2 + a_3y \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} b_0 + b_1y \\ b_2 + b_3y \end{pmatrix} \right)$$

$$= \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix} \left(\begin{pmatrix} a_0 + a_1y \\ (a_0 + a_2) + (a_1 + a_3)y \\ a_2 + a_3y \end{pmatrix} \cdot \begin{pmatrix} b_0 + b_1y \\ (b_0 + b_2) + (b_1 + b_3)y \\ b_2 + b_3y \end{pmatrix} \right)$$

$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \\ c_6 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 1 & 0 & 0 & -1 & 0 & 0 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ 0 & 0 & -1 & 0 & 0 & 1 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \left(P \begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} \cdot P \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix} \right), P = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Данный алгоритм содержит 9 умножений и 20 сложений, в то время как оптимальный алгоритм требует 7 умножений, но большего числа сложений. Данный алгоритм можно проитерировать и получить алгоритм 16-точечной свертки, содержащий 81 умножение и т.д.

При вычислении циклической свертки не всегда удается воспользоваться алгоритмом Агарвала-Кули. В этом случае могут быть использованы аналогичные приемы. Однако в случае вычислений над конечными полями возникают определенные сложности. Рассмотрим вычисление циклической свертки $c(x) = a(x)b(x) \bmod x^n - 1$ длины $n = p^\lambda, \lambda \geq 1, p$ — простое число. Пусть вычисления производятся в поле $GF(p^m)$. В этом случае $(x^n - 1) = (x - 1)^n$ и разложение модуля на взаимно простые сомножители (т.е. использование алгоритма Винограда) невозможно. Пусть $a(x) = \sum_{i=0}^{p-1} x^i \sum_{k=0}^{n/p-1} a_{i+pk}x^{pk} = \sum_{i=0}^{p-1} x^i a_i(x^p), b(x) = \sum_{i=0}^{p-1} x^i b_i(x^p)$. Тогда $c(x) = \sum_{k=0}^{2p-2} x^k \sum_{i=0}^{k} a_{k-i}(x^p)b_i(x^p) \bmod (x^n - 1)$. Необходимо отметить, что индексы многочленов здесь не приводятся по модулю. Считается, что $a_i(x) = b_i(x) = 0, i \geq p$.

С учетом этого можно записать

$$\begin{aligned} c(x) = & \sum_{k=0}^{2p-1} x^k \sum_{i=0}^{k} a_{k-i}(x^p) b_i(x^p) \bmod (x^n - 1) \\ = c(z,x)|_{z=x^p} = & \sum_{k=0}^{p-1} x^k \left(\sum_{i=0}^{k} a_{k-i}(z) b_i(z) + z \sum_{i=k+1}^{p-1} a_{k-i+p}(z) b_i(z) \right) \bmod (z^{n/p} - 1) \end{aligned}$$

Таким образом, задача свелась к вычислению n/p-точечных циклических сверток многочленов от z, которые могут быть вычислены с помощью быстрого алгоритма меньшей размерности.

Пример 3.5. Построим четырехточечную циклическую свертку над полем $GF(2^m)$.

$$\begin{aligned} c(x) = & (a_0 + a_1 x + a_2 x^2 + a_3 x^3)(b_0 + b_1 x + b_2 x^2 + b_3 x^3) \bmod (x^4 - 1) \\ = & ((a_0 + a_2 x^2) + x(a_1 + a_3 x^2))((b_0 + b_2 x^2) + x(b_1 + b_3 x^2)) \bmod (x^4 - 1) \\ = & ((a_0 + a_2 z)(b_0 + b_2 z) + z(a_1 + a_3 z)(b_1 + b_3 z)) \\ & + x\left((a_1 + a_3 z)(b_0 + b_2 z) + (a_0 + a_2 z)(b_1 + b_3 z)\right) \bmod (z^2 - 1) \end{aligned}$$

Последнее произведение можно было вычислить с помощью алгоритма Карацубы:

$$\begin{aligned} c(x) = & ((a_0 + a_2 z)(b_0 + b_2 z) + z(a_1 + a_3 z)(b_1 + b_3 z)) \\ + x & (((a_0 + a_1) + (a_2 + a_3)z)((b_0 + b_1) + (b_2 + b_3)z) \\ & - (a_0 + a_2 z)(b_0 + b_2 z) - (a_1 + a_3 z)(b_1 + b_3 z)) \bmod (z^2 - 1) \end{aligned}$$

Окончательно получаем

$$\begin{pmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \left(\begin{pmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{pmatrix} \cdot P \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{pmatrix} \right), P = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

Используя другую форму быстрого алгоритма двухточечной циклической свертки $c_0 = a_0(b_0 + b_1) + (a_0 + a_1)b_1 + x(a_0(b_0 + b_1) + (a_0 + a_1)b_0)$, можно было бы получить другой алгоритм с 9 умножениями, в котором матрицы предсложений были бы разными [5]. При этом одна из этих матриц содержала бы большое число единиц. Как будет показано в дальнейшем, это свойство может быть очень полезно при построении других алгоритмов.

В поле характеристики 2 можно предложить еще один способ вычисления циклической свертки длины $n = 2^\lambda$, близкий к алгоритму Карацубы. Пусть $a(x) = a_0(x) + x^m a_1(x), b(x) = b_0(x) + x^m b_1(x), \deg a_i(x) < m, \deg b_i(x) < m, m = 2^{\lambda-1}$. Рассмотрим вычисление $c(x) \equiv a(x)b(x) \bmod (x^m - 1)^2$. Пусть $u_0(x) = a_0(x) + a_1(x), u_1(x) =$

$a_1(x), w_0(x) = b_0(x) + b_1(x), w_1(x) = b_1(x)$. Ясно, что $a(x) = u_0(x) + (x^m - 1)u_1(x), b(x) = w_0(x) + (x^m - 1)w_1(x)$. Тогда

$$a(x)b(x) \equiv u_0(x)w_0(x) + (x^m - 1)\left(u_0(x)w_1(x) + u_1(x)w_0(x)\right) \bmod (x^{2m} - 1)$$

Таким образом,

$$\begin{aligned} a(x)b(x) \bmod (x^{2m} - 1) = \quad & (a_0(x) + a_1(x))(b_0(x) + b_1(x)) \bmod (x^{2m} - 1) + \\ & (x^m - 1)\left((a_0(x) + a_1(x))b_1(x) + a_1(x)(b_0(x) + b_1(x)) \bmod (x^m - 1)\right) \end{aligned}$$

3.3 Дискретное преобразование Фурье

3.3.1 Преобразование Фурье в дискретном и непрерывном случаях

Предположим, что задана функция $f(x)$, периодическая на интервале $(-\pi, \pi)$. Такая функция может быть разложена в ряд Фурье

$$f(x) = \sum_{k=-\infty}^{\infty} A_k e^{ikx},$$

где

$$A_k = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(t)e^{-ikt}\mathrm{d}t.$$

Ясно, что это определение можно легко обобщить на случай других значений L периода функции. В пределе, при $L \to \infty$ ряд Фурье переходит в *преобразование Фурье*:

$$f(x) = \int_{-\infty}^{\infty} F(t)e^{2\pi itx}\mathrm{d}t$$

(обратное преобразование Фурье),

$$F(t) = \int_{-\infty}^{\infty} f(x)e^{-2\pi itx}\mathrm{d}x$$

(прямое преобразование Фурье). Иногда его также определяют как

$$h(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} H(\omega)e^{i\omega t}\mathrm{d}\omega$$

и

$$H(\omega) = \int_{-\infty}^{\infty} h(t)e^{-i\omega t}\mathrm{d}t.$$

Предположим, что функция $f(t)$ дискретна и задана своими отсчетами $f_k = f(k\Delta t), k = 0..n-1$. Это приводит к следующему определению *дискретного преобразования Фурье* последовательности f_k:

$$F_j = \sum_{k=0}^{n-1} f_k e^{-2\pi ijk/n}. \tag{3.5}$$

Обратное дискретное преобразование Фурье задается выражением

$$f_k = \frac{1}{n} \sum_{j=0}^{n-1} F_n e^{2\pi ikj/n} \tag{3.6}$$

Как обычно, множитель $1/n$, а также -1 в показателе степени может перераспределяться между этими выражениями. Пусть $\alpha = e^{-2\pi i/n}$, или, в более общем случае, α — некоторый элемент порядка n. Тогда

$$\begin{aligned} F_j &= \sum_{k=0}^{n-1} f_k \alpha^{jk} \\ f_k &= \frac{1}{n} \sum_{j=0}^{n-1} F_j \alpha^{-jk} \end{aligned}$$

Величина α называется *ядром преобразования*. Чтобы убедиться в справедливости этих соотношений, заметим, что элемент порядка n должен удовлетворять равенству $\alpha^n - 1 = 0$, а также $\alpha^{rn} - 1 = 0$. Над любым полем $x^n - 1 = (x-1)(x^{n-1} + \ldots + 1) = 0$. Следовательно, для всех $r \neq 0 \bmod n$ элемент α^r является корнем последнего многочлена в этом разложении. Это означает, что

$$\sum_{j=0}^{n-1} \alpha^{rj} = 0, r \neq 0 \bmod n.$$

Если $r = 0 \bmod n$, то

$$\sum_{j=0}^{n-1} \alpha^{rj} = n.$$

Это не равно нулю, если n не кратно характеристике поля. Таким образом,

$$\sum_{j=0}^{n-1} \alpha^{-jk} \sum_{l=0}^{n-1} \alpha^{lj} f_l = \sum_{l=0}^{n-1} f_l \sum_{j=0}^{n-1} \alpha^{(l-k)j} = n f_k.$$

Вычисление ДПФ можно рассматривать как умножение вектора $(f_0, \ldots, f_{n-1})^T$ на *матрицу Вандермонда*, которая, как известно, невырождена.

В поле комплексных чисел ДПФ существует для всех $n > 0$. В конечных полях $GF(p^m)$ это не так. Действительно, ненулевые элементы поля образуют группу по умножению, а все степени α (не путать α с примитивным элементом поля!) образуют некоторую циклическую конечную подгруппу. В соответствии с теоремой 2.9, порядок этой подгруппы должен быть делителем порядка исходной группы, т.е. $p^m - 1$. Например, в поле $GF(2^4)$ существуют ДПФ длины $3, 5, 15$, но не существует ДПФ длины 8. Если ДПФ нужной длины не существует, можно взять некоторое расширение поля. Однако во многих случаях это крайне непрактично.

Необходимо отметить, что прямое и обратное преобразование Фурье симметричны. Действительно, $F_{n-j} = \sum_{k=0}^{n-1} f_k \alpha^{(n-j)k} = \sum_{k=0}^{n-1} f_k \alpha^{-jk}$. Кроме того, можно заметить, что F_j равны значению многочлена $f(x) = \sum_{k=0}^{n-1} f_k x^k$ в точках α^j. Следовательно, ОДПФ соответствует операции интерполяции. Это свойство используется для поиска корней многочленов над конечными полями. Действительно, если F_i (i-я спектральная компонента) оказывается равным нулю, то $f(\alpha^i) = 0$ т.е. α^i является корнем.

Теорема 3.2 (О свертке)**.** *Пусть существует ДПФ длины n и*

$$e_i = f_i g_i, i = 0..n-1.$$

Тогда

$$E_j = \frac{1}{n} \sum_{k=0}^{n-1} F_{((j-k))} G_k.$$

Доказательство. Вычислим ДПФ вектора с компонентами $e_i = f_i g_i$:

$$E_j = \sum_{i=0}^{n-1} \alpha^{ij} f_i g_i = \sum_{i=0}^{n-1} \alpha^{ij} f_i \frac{1}{n} \sum_{k=0}^{n-1} \alpha^{-ik} G_k = \frac{1}{n} \sum_{k=0}^{n-1} G_k \left(\sum_{i=0}^{n-1} \alpha^{i(j-k)} f_i \right) = \frac{1}{n} \sum_{k=0}^{n-1} G_k F_{((j-k))}$$

□

Это тождество может быть использовано для быстрого вычисления циклической свертки.

ДПФ применяется практически в огромном числе вычислительных процедур, например:

- Цифровая обработка сигналов;
- Цифровая связь (ADSL, WiMAX,...)
- Сжатие и обработка изображений
- Помехоустойчивое кодирование

Непосредственное вычисление ДПФ требует n^2 умножений. Но за счет использования алгебраических свойств элемента α и поля, в котором происходят вычисления,

сложность удается существенно сократить. Различные алгоритмы быстрого вычисления ДПФ получили название *быстрого преобразования Фурье* (БПФ). Некоторые из этих приемов реализованы аппаратно в цифровых сигнальных процессорах. Тем не менее, во многих случаях оказывается необходимым вычисление ДПФ на длинах, не поддерживаемых аппаратно. В этом случае возникает задачу необходимо свести к набору вычислительных примитивов, поддерживаемых аппаратно.

3.3.2 Общие алгоритмы быстрого преобразования Фурье

Алгоритм Кули-Тьюки

Предположим, что необходимо вычислить ДПФ длины $n = n'n''$. Пусть $i = i' + n'i'', k = n''k' + k''$. Тогда

$$F_k = F_{n''k'+k''} = \sum_{i=0}^{n-1} f_i \alpha^{ki} = \sum_{i''=0}^{n''-1} \sum_{i'=0}^{n'-1} \alpha^{(n''k'+k'')(i'+n'i'')} f_{i'+n'i''}.$$

Пусть $\gamma = \alpha^{n'}, \beta = \alpha^{n''}$. Заметим, что $\forall k', i'' : \alpha^{n'n''k'i''} = 1$. Пусть $f_{i',i''} = f_{i'+n'i''}, F_{k',k''} = F_{n''k'+k''}$. Тогда

$$F_{k',k''} = \sum_{i'=0}^{n'-1} \beta^{k'i'} \left(\alpha^{i'k''} \sum_{i''=0}^{n''-1} \gamma^{i''k''} f_{i',i''} \right)$$

Заметим, что выражение во внутренних скобках зависит только от i', k'', а внутреннее выражение представляет собой набор ДПФ с ядром γ, результат которых после нескольких дополнительных умножений передается на вход другим ДПФ с ядром β. Таким образом, ДПФ может быть вычислено следующим образом (*алгоритм Кули-Тьюки*):

1. Вычислить $\sum_{i''=0}^{n''-1} \gamma^{i''k''} f_{i',i''}, i' = 0..n'-1, k'' = 0..n''-1$. Непосредственное выполнение этого шага требует $n'n''n''$ умножений и $n'n''(n''-1)$ сложений.

2. Домножить результат предыдущего шага на $\alpha^{i'k''}, i' = 0..n'-1, k'' = 0..n''-1$. Это требует $n'n''$ умножений. Пусть $t_{i'k''}$ — результат этого шага.

3. Вычислить $F_{k',k''} = \sum_{i'=0}^{n'-1} \beta^{k'i'} t_{i'k''}$. Аналогично, этот шаг требует $n'n''n'$ умножений и $n'n''(n'-1)$ сложений.

Таким образом, общая сложность вычислений составляет $M(n) = n(n''+n'+1)$ умножений и $A(n) = n(n''+n'-2)$ сложений. В некоторых случаях величины $\beta^{k'i'}$ имеют специальный вид, облегчающий вычисления. Дальнейшее снижение сложности может быть получено за счет использования для вычисления ДПФ малой длины других быстрых алгоритмов, необязательно Кули-Тьюки. Тогда $M(n) = n'M(n'') + n''M(n') + n, A(n) = n'A(n'') + n''A(n')$.

Во многих приложениях длина преобразования над полем комплексных чисел оказывается равной $n = 2^m$ (БПФ по основанию 2). В этом случае $n' = 2, n'' = 2^{m-1}$ (прореживание по времени) или $n'' = 2^{m-1}, n' = 2$ (прореживание по частоте). В первом случае $\beta = \alpha^{n/2} = -1$. Тогда получим

$$\begin{aligned} F_k &= \sum_{i=0}^{n/2-1} \alpha^{2ik} f_{2i} + \alpha^k \sum_{i=0}^{n/2-1} \alpha^{2ik} f_{2i+1} \\ F_{k+n/2} &= \sum_{i=0}^{n/2-1} \alpha^{2ik} f_{2i} - \alpha^k \sum_{i=0}^{n/2-1} \alpha^{2ik} f_{2i+1}, k = 0..n/2-1 \end{aligned}$$

Таким образом, входной вектор распадается на два подвектора, элементы которых записаны через одну позицию. Если процедура БПФ для этих векторов длины $n/2$ сохраняет результат во входном массиве, то и рассматриваемая процедура БПФ длины n может сохранить результат во входном векторе, воспользовавшись всего одной временной переменной.

Аналогичная ситуация возникает и при вычислении БПФ с прореживанием по частоте. Сложность описанных алгоритмов удовлетворяет рекуррентным соотношениям $M(n) = 2M(n/2) + n/2 = O((n/2)\log_2 n), A(n) = 2A(n/2) + n = n\log_2 n$. Сложность алгоритма можно немного уменьшить, если заметить, что на самом внутреннем шаге алгоритма все умножения производятся на ± 1; на предпоследнем шаге умножения производятся на константы $i^k, k = 0..3$. Некоторые упрощения возможны и на последующих шагах. В связи с этим, как правило, БПФ малых длин реализуют в виде линейных программ, в которых реализуются все возможные упрощения. Кроме того, это позволяет экономить на издержках, связанных с организацией циклов и т.п. управляющими конструкциями.

Популярны также алгоритмы БПФ по основанию 4.

Алгоритм Кули-Тьюки был предложен в 1805 году Гауссом и переоткрыт в 1965 году Кули (Cooley) и Тьюки (Tukey),

Алгоритм Гуда-Томаса

Предположим, что $n = n'n'', (n', n'') = 1$. Пусть $N'n' + N''n'' = 1$. Представим числа $i = 0..n-1$ в виде пары вычетов по модулям n', n'':

$$i' \equiv i \bmod n', i'' \equiv i \bmod n'', i \equiv i'N''n'' + i''N'n' \bmod n.$$

Выходные индексы представим так:

$$k' = N''k \bmod n', k'' = N'k \bmod n'', k = n''k' + n'k''.$$

Действительно, $n''(N''k + Q'n') + n'(N'k + Q''n'') = k(N'n' + N''n'') \equiv k \bmod n$. Тогда

$$F_k = F_{k',k''} = \sum_{i'=0}^{n'-1} \sum_{i''=0}^{n''-1} \alpha^{(n''k'+n'k'')(i'N''n''+i''N'n')} f_{i'N''n''+i''N'n'} = \sum_{i'=0}^{n'-1} \beta^{i'k'} \left(\sum_{i''=0}^{n''-1} \gamma^{i''k''} f_{i',i''} \right),$$

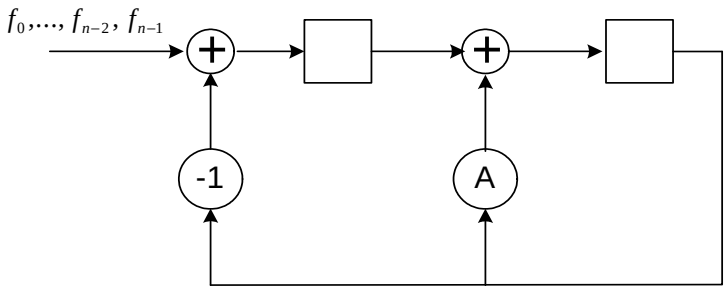


Рис. 3.1: Авторегрессионный фильтр с передаточной функцией $x^2 - Ax + 1$

где $\beta = \alpha^{N''(n'')^2}, \gamma = \alpha^{N'(n')^2}$. *Алгоритм Гуда-Томаса* требует $n'n''n'' + n''n'n' = n(n' + n'')$ умножений. Число сложений совпадает с алгоритмом Кули-Тьюки.

Требование взаимной простоты сомножителей ограничивает возможность построения рекурсивных алгоритмов. В связи с этим алгоритм Гуда-Томаса обычно комбинируют с другими алгоритмами.

Алгоритм Герцеля

В некоторых случаях требуется вычислить небольшое количество элементов вектора ДПФ. Пусть необходимо найти (возможно, среди прочих) компоненту $F_k = \sum_{i=0}^{n-1} \alpha^{ik} f_i = f(\alpha^k)$. Построим многочлен $\phi(x)$ минимальной степени с "вычислительно простыми" коэффициентами, имеющий α^k своим корнем. В случае комплексного ДПФ этот многочлен имеет вид $\phi(x) = (x - \alpha^k)(x - \alpha^{-k}) = x^2 - 2\cos(2\pi k/n) + 1$ (вещественные коэффициенты!). В случае конечных полей воспользуемся минимальным многочленом α^k. Пусть $f(x) = Q(x)\phi(x) + r(x)$. Ясно, что $r(\alpha^k) = f(\alpha^k)$. Вычисление остатка от деления можно реализовать с помощью авторегрессионного фильтра (см. рис. 3.1).

Оценим сложность для случая вычислений в комплексном поле. Если бы в качестве многочлена $\phi(x)$ был взят полином с комплексными коэффициентами, деление (т.е. вычисление по схеме Горнера) потребовало бы $2(n-2)$ вещественных умножения. За счет использования многочлена с вещественными коэффициентами сложность уменьшается до $n-2$ умножений.

В случае конечных полей экономия может быть намного более существенной, т.к. умножения в простом поле существенно проще умножений в расширенном поле.

Преобразование Фурье как свертка

Рассмотрим *алгоритм Блюстейна*. Воспользуемся тождеством $ik = \frac{1}{2}\left(i^2 + k^2 - (i-k)^2\right)$. Тогда $F_k = \sum_{i=0}^{n-1} \alpha^{ik} f_i = \alpha^{k^2/2} \sum_{i=0}^{n-1} \alpha^{i^2/2} f_i \alpha^{-(i-k)^2/2}$. Произведем замену переменных $G_k = \alpha^{-k^2/2} F_k, g_i = \alpha^{i^2/2} f_i, h_{i-k} = \alpha^{-(i-k)^2/2}$. Тогда задача вычисления ДПФ сводится к вычислению

$$\begin{pmatrix} G_0 \\ G_1 \\ \vdots \\ G_{N-1} \end{pmatrix} = \begin{pmatrix} h_0 & h_1 & h_2 & \dots & h_{N-1} \\ h_1 & h_0 & h_1 & \dots & h_{N-2} \\ \dots & & & & \\ h_{N-1} & h_{N-2} & h_{N-3} & \dots & h_0 \end{pmatrix} \begin{pmatrix} g_0 \\ g_1 \\ \vdots \\ g_{N-1} \end{pmatrix}$$

Полученная матрица носит название *теплицевой* (все ее диагонали содержат одинаковые элементы). Пока это привело лишь к увеличению сложности. Однако полученная $N \times N$ матрица может быть расширена до $2N-2$ *циркулянтной* (т.е. такой матрицы, все строки которой являются циклическими сдвигами друг друга). Например, 6×6 матрица может быть преобразована следующим образом:

$$\begin{pmatrix} h_0 & h_1 & h_2 & h_3 & h_4 & h_5 \\ h_1 & h_0 & h_1 & h_2 & h_3 & h_4 \\ h_2 & h_1 & h_0 & h_1 & h_2 & h_3 \\ h_3 & h_2 & h_1 & h_0 & h_1 & h_2 \\ h_4 & h_3 & h_2 & h_1 & h_0 & h_1 \\ h_5 & h_4 & h_3 & h_2 & h_1 & h_0 \end{pmatrix} \longrightarrow \begin{pmatrix} h_0 & h_1 & h_2 & h_3 & h_4 & h_5 & h_4 & h_3 & h_2 & h_1 \\ h_1 & h_0 & h_1 & h_2 & h_3 & h_4 & h_5 & h_4 & h_3 & h_2 \\ h_2 & h_1 & h_0 & h_1 & h_2 & h_3 & h_4 & h_5 & h_4 & h_3 \\ h_3 & h_2 & h_1 & h_0 & h_1 & h_2 & h_3 & h_4 & h_5 & h_4 \\ h_4 & h_3 & h_2 & h_1 & h_0 & h_1 & h_2 & h_3 & h_4 & h_5 \\ h_5 & h_4 & h_3 & h_2 & h_1 & h_0 & h_1 & h_2 & h_3 & h_4 \\ h_4 & h_5 & h_4 & h_3 & h_2 & h_1 & h_0 & h_1 & h_2 & h_3 \\ h_3 & h_4 & h_5 & h_4 & h_3 & h_2 & h_1 & h_0 & h_1 & h_2 \\ h_2 & h_3 & h_4 & h_5 & h_4 & h_3 & h_2 & h_1 & h_0 & h_1 \\ h_1 & h_2 & h_3 & h_4 & h_5 & h_4 & h_3 & h_2 & h_1 & h_0 \end{pmatrix}$$

Умножение на такую матрицу эквиваленетно вычислению циклической свертки. Таким образом, задача вычисления ДПФ свелась к вычислению циклической свертки дополненного нулями преобразованного входного вектора g с фиксированной последовательностью h, отбрасыванию лишних компонентов полученного вектора и делению их на соответствующие константы.

БПФ простой длины

Алгоритм Рейдера также позволяет свести задачу вычисления ДПФ простой длины над некоторым полем $\mathbb{F}$ к циклической свертке. Пусть $F_k = \sum_{i=0}^{n-1} \alpha^{ik} f_i$. Тогда

$$\begin{aligned} F_0 &= \sum_{i=0}^{n-1} f_i \\ F_k &= f_0 + \sum_{i=1}^{n-1} \alpha^{ik} f_i \end{aligned}$$

Если n — простое число, то существует поле $GF(n)$ с примитивным элементом π. Пусть $r(i) : \pi^{r(i)} = i$. Тогда $F_{\pi^{r(k)}} = f_0 + \sum_{i=1}^{n-1} \alpha^{\pi^{r(i)+r(k)}} f_{\pi^{r(i)}}$ Сделаем замену переменных $l = r(k), j = n - 1 - r(i)$. Тогда

$$F_{\pi^l} = f_0 + \sum_{j=1}^{n-1} \alpha^{\pi^{l-j}} f_{\pi^{n-1-j}} \tag{3.7}$$

Таким образом, задача вычисления ДПФ свелась к задаче вычисления циклической свертки переставленной входной последовательности и фиксированной последовательности. Применяя быстрый алгоритм Винограда вычисления циклической свертки, получим *алгоритм Винограда БПФ*. Существенным недостатком алгоритма Винограда является большое число нерегулярных сложений, что затрудняет векторизацию вычислений (SIMD).

С другой стороны, для вычисления циклической свертки может использоваться алгоритм БПФ длины $n-1$, которое не является простым числом, т.е. для его вычисления могут быть использованы другие алгоритмы БПФ.

3.3.3 Алгоритмы БПФ в конечных полях

В конечных полях могут быть применены все вышеописанные общие алгоритмы БПФ. Однако применение некоторых специфических свойств позволяет существенно снизить сложность вычислений. Как правило, на практике приходится комбинировать специализированные и общие алгоритмы.

Некоторые свойства конечных полей

Определение 3.1. [11] *Линеаризованными многочленами* над $GF(p^m)$ называются многочлены вида

$$L(x) = \sum_i l_i x^{p^i}, l_i \in GF(p^m)$$

В том случае, когда $l_i \in GF(p)$, $L(x)$ называется p-полиномом.

Несложно показать, что для линеаризованных многочленов выполняется

$$L(a + b) = L(a) + L(b),\ a, b \in GF(p^m). \tag{3.8}$$

Из этого свойства вытекает следующая теорема, представленная здесь в немного модифицированном виде:

Теорема 3.3 ([4])**.** *Пусть $y \in GF(p^m)$ и элементы $\beta_0, \beta_1, ..., \beta_{m-1}$ образуют некоторый базис этого поля. Если*

$$y = \sum_{i=0}^{m-1} y_i \beta_i, y_i \in GF(p),$$

то

$$L(y) = \sum_{i=0}^{m-1} y_i L(\beta_i).$$

Таким образом, линеаризованные многочлен однозначно задается своими значениями в базисных элементах поля. Т.к. множество его значений входит в $GF(p^m)$, вектор этих значений может рассматриваться также как матрица $\mathcal{L}$ над $GF(p)$. Это дает возможность находить корни линеаризованных многочленов путем решения однородной системы линейных уравнений $y\mathcal{L} = 0$. Отсюда также следует то, что корни линеаризованных многочленов образуют линейное подпространство в $GF(p)^m$.

Обобщением линеаризованных многочленов являются *аффинные многочлены*, определяемые как $A(x) = a + L(x)$, где $a \in GF(p^m)$ и $L(x)$ — линеаризованный многочлен.

Определение 3.2. Следом элемента $\delta \in GF(p^m)$ называется величина

$$\mathrm{Tr}(\delta) = \sum_{i=0}^{m-1} \delta^{p^i}$$

Заметим, что $\mathrm{Tr}(\delta)^p = \sum_{i=0}^{m-1} \delta^{p^{i+1}} = \sum_{i=1}^{m} \delta^{p^i} = \sum_{i=0}^{m-1} \delta^{p^i} = \mathrm{Tr}(\delta) \Rightarrow \mathrm{Tr}(\delta) \in GF(p)$.

Если β — примитивный элемент конечного поля $GF(p^m)$, то всякий его элемент может быть представлен как $x = \sum_{i=0}^{m-1} x_i \beta^i, x_i \in GF(p)$. Множество $\{\beta^0, \ldots, \beta^{m-1}\}$ называется *стандартным базисом* конечного поля. Т.к. поле $GF(p^m)$ может рассматриваться как векторное пространство $GF(p)^m$, существуют и другие базисы, причем для любого базиса $(\pi_0, \ldots, \pi_{m-1})$ справедливо

$$\begin{pmatrix} 1 \\ \beta \\ \vdots \\ \beta^{m-1} \end{pmatrix} = C \begin{pmatrix} \pi_0 \\ \pi_1 \\ \vdots \\ \pi_{m-1} \end{pmatrix}.$$

Часто бывает полезен *нормальный базис* $\{\gamma^{p^0}, \gamma^{p^1}, \ldots, \gamma^{p^{m-1}}\}$. Можно показать, что в любом конечном поле существует нормальный базис. Например, в поле $GF(2^2)$ нормальный базис совпадает со стандартным $\{\beta, \beta^2\}$, где β — примитивный элемент. В поле $GF(2^4)$, задаваемом многочленом $x^4 + x + 1$, нормальный базис образован элементами $\beta^3, \beta^6, \beta^{12}, \beta^9$. Ясно, что образующий элемент γ нормального базиса должен удовлетворять $\mathrm{Tr}(\delta) \neq 0$.

Пусть $(n, p) = 1$. Рассмотрим многочлен $x^n - 1$. Его корни лежат в некотором расширенном поле $GF(p^m)$. Вычислим формальную производную многочлена $x^n - 1$. Она равна nx^{n-1}. Т.к. $(n, p) = 1$, многочлены nx^{n-1} и $x^n - 1$ взаимно просты, т.е. $x^n - 1$ не имеет кратных корней. Все его корни $\alpha_0, \ldots, \alpha^{n-1}$ называются корнями n-й степени из единицы.

Из вышесказанного следует, что если необходимо вычислить ДПФ вектора $(f_0, \ldots, f_{n-1}), f_i \in GF(p^l)$ и $n \nmid p^l - 1$, то корни $x^n - 1$ лежат в некотором другом

поле $GF(p^m)$, а следовательно, вычисления надо производить в некотором третьем расширенном поле $GF(p^{lcm(l,m)})$, содержащем как поле элементов вектора, так и корни из единицы. Результат ДПФ также будет принадлежать этому полю. Вычисления в больших расширениях могут быть весьма трудоемки, поэтому таких ситуаций следует по возможности избегать.

Циклотомическим классом по модулю n над $GF(p)$ называется множество

$$C_s = \{s, sp, sp^2, \ldots, sp^{m_s-1}\}, sp^{m_s} \equiv s \bmod n.$$

Ясно, что множество целых чисел $o \leq i \leq n-1$ разбивается на набор циклотомических классов, т.е.

$$\{0, \ldots, n-1\} = \bigcup_s C_s,$$

где s пробегает множество представителей циклотомических классов. Заметим, что циклотомические классы являются классами эквивалентности по бинарному отношению $\rho = \{(x, y) | \exists l : x \equiv yp^l \bmod n\}$.

Пример 3.6. Циклотомическими классами по модулю 9 над $GF(2)$ являются $\{0\}, \{1, 2, 4, 8, 7, 5\}, \{3, 6\}$.

Минимальным многочленом элемента α^s является многочлен с коэффициентами из $GF(p)$, равный

$$M^{(s)}(x) = \prod_{i \in C_s} (x - \alpha^i).$$

Ясно, что $x^n - 1 = \prod_s M^{(s)}(x)$, где s пробегает множество представителей ЦТК. Максимальный размер ЦТК по модулю n над $GF(p)$ (или, эквивалентно, максимальная степень неприводимого над $GF(p)$ сомножителя $x^n - 1$) указывает то расширение поля, в котором существуют корни степени n из 1. Элементы $\beta^i, i \in C_s$ (или являющиеся корнями одного минимального многочлена) называются сопряженными. Заметим, что это же определение справедливо и в поле комплексных чисел, где элементы $\pm i$ являются корнями $x^2 + 1$.

Далее, если не указано иное, будут рассматриваться поля характеристики 2.

Алгоритм Ванга-Жу

Классический алгоритм Герцеля заменяет задачу вычисления значений многочлена $f(x)$ в точках α^i вычислением значений остатков от деления $f(x)$ на минимальные многочлены элементов α^i, т.е. остатков от деления на $(x - \alpha^i)$. Это требует $O(n \log_2 n)$ умножений и $O(n^2)$ сложений. В [15] было предложено обобщение этого подхода, состоящее в том, что деление может производиться на некоторые многочлены с корнями α^i, от которых требуется лишь, чтобы число ненулевых (или вычислительно сложных) коэффициентов в них было небольшим относительно их степени. Такому критерию удовлетворяют линеаризованные и аффинные многочлены.

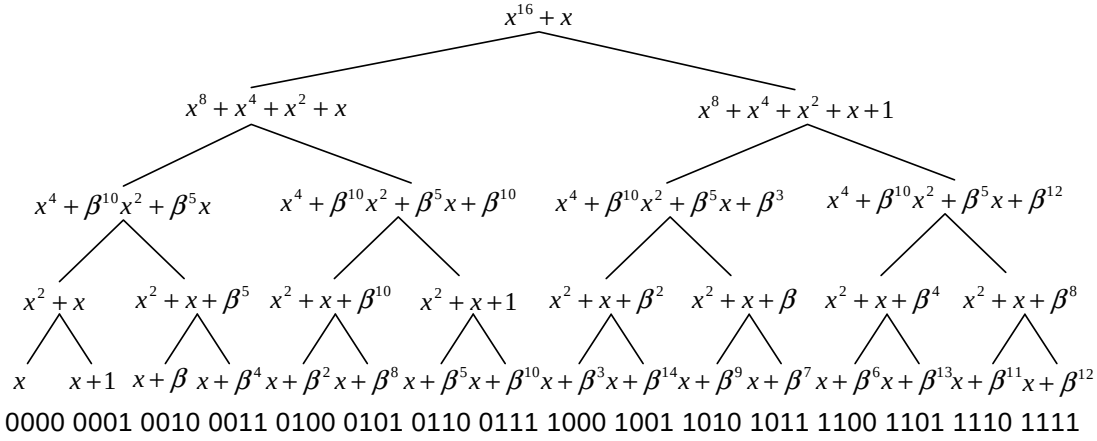


Рис. 3.2: Рекурсивное разбиение Ванга-Жу

Пример 3.7. Рассмотрим вычисление ДПФ над $GF(2^4)$. В качестве ядра преобразования воспользуемся примитивным элементом поля, т.е. $\alpha = \beta$. Будем вычислять расширенное преобразование, т.е. дополним вектор $(F_0, \ldots, F_{n-1})$ компонентой $F_{-\infty} = f(0)$. Пример подобного разбиения представлен на рис. 3.2. На первом шаге алгоритма вычисляются остатки от деления $f(x) = \sum_{i=0}^{15} f_i x^i$ на многочлены $x^8 + x^4 + x^2 + x$ и $x^8 + x^4 + x^2 + x + 1$. Далее осуществляется рекурсивное вычисление остатков от деления на их сомножители до тех пор, пока не получается остаток нулевой степени, т.е. компоненты ДПФ.

Как было показано выше, множество корней линеаризованного многочлена образует линейное подпространство $GF(2^m)$. В частности, множество корней $x^{2^m} - x$ совпадает с линейным пространством $GF(2^m)$. Пусть задан линеаризованный многочлен $L_j(x)$ с j-мерным пространством корней $\mathcal{L}_j$. Выберем какое-нибудь $j-1$-мерное подпространство $\mathcal{L}_{j-1}$ его корней. Ясно, что оно соответствует некоторому другому линеаризованному многочлену $\mathcal{L}_{j-1}$. Оставшиеся корни могут представлены как (смежный класс $\mathcal{L}_{j-1}$) $\alpha_i = \hat{\alpha} + \alpha_k, \alpha_k \in \mathcal{L}_{j-1}, \hat{\alpha} \notin \mathcal{L}_{j-1}$. Тогда $\prod_{\alpha_i \notin \mathcal{L}_{j-1}} (x - \alpha_i) = \prod_{\alpha_k \in \mathcal{L}_{j-1}} (x - \hat{\alpha} - \alpha_k) =$ $L_{j-1}(x - \hat{\alpha}) = \sum_{s=0}^{j-1} l_{j-1,s}(x - \hat{\alpha})^{2^s} = \sum_{s=0}^{j-1} l_{j-1,s} x^{2^s} + \sum_{s=0}^{j-1} l_{j-1,s} \hat{\alpha}^{2^s} = L_{j-1}(x) + a$, что есть аффинный многочлен. Таким образом, произвольный линеаризованный многочлен может быть разложен на произведение линеаризованного и аффинного многочленов. Это означает, что корни n-й степени из единицы (и 0) должны быть сгруппированы по смежным классам линейных подпространств. Такое расположение элементов можно получить, упорядочив их в соответствии с двоичным значением их векторного представления. Однако такое представление не является самым эффективным. Же-

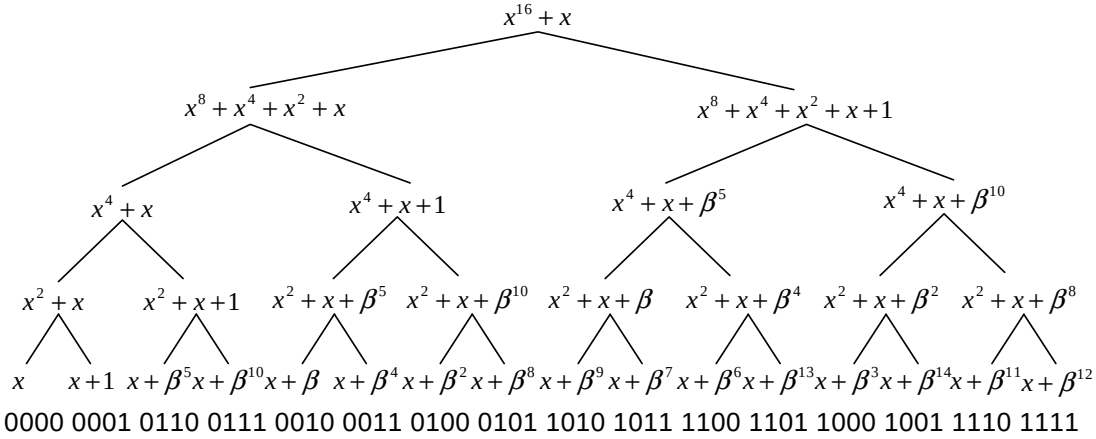


Рис. 3.3: Рекурсивное разбиение Ванга-Жу: группировка по модулям

лательно максимизировать число 2-полиномов в дереве разбиения. Корни 2-полинома обладают важным дополнительным свойством: если $L(\mu) = 0$, то $0 = L^2(\mu) = L(\mu^2)$. Подпространство с таким свойством называется модулем. Таким образом, желательно группировать элементы поля в модули. Однако до настоящего времени не ясно, всегда ли возможно такое разбиение. Пример его приведен на рис. 3.3 . Можно показать, что число сложений и умножений не превосходит $O(n \log^2 n)$.

Циклотомический алгоритм БПФ

Другой способ использования факта существования линеаризованных многочленов был предложен в [14]. Рассмотрим набор циклотомических классов по модулю n над $GF(2)$:

$$\{0\}, \{k_1, k_1 2, k_1 2^2, \ldots, k_1 2^{m_1-1}\}, \ldots, \{k_l, k_l 2, k_l 2^2, \ldots, k_l 2^{m_l-1}\},$$

где $k_i \equiv k_i 2^{m_i} \bmod n$.

Многочлен $f(x) = \sum_{i=0}^{n-1} f_i x^i, f_i \in GF(2^m)$ может быть разложен как

$$f(x) = \sum_{i=0}^{l} L_i(x^{k_i}), \quad L_i(y) = \sum_{j=0}^{m_i-1} f_{k_i 2^j \bmod n} y^{2^j}. \tag{3.9}$$

Действительно, выражение (3.9) представляет собой способ группировки чисел $s \in [0, n-1]$ по циклотомическим классам: $s \equiv k_i 2^j \bmod n$. Очевидно, что такое разбиение существует всегда. Заметим, что при $k_i = 0$ свободный член f_0 мы можем записать как значение многочлена $L_0(y) = f_0 y$ при $y = x^0$.

Выражение (3.9) будем называть циклотомическим разложением многочлена $f(x)$.

Пример 3.8. Многочлен $f(x) = \sum_{i=0}^{6} f_i x^i, f_i \in GF(2^3)$ представляется как

$$\begin{aligned} f(x) = &\quad L_0(x^0) + L_1(x) + L_2(x^3); \\ L_0(y) = &\quad f_0 y, \\ L_1(y) = &\quad f_1 y + f_2 y^2 + f_4 y^4, \\ L_2(y) = &\quad f_3 y + f_6 y^2 + f_5 y^4. \end{aligned}$$

В соответствии с разложением (3.9) запишем $f(\alpha^j) = \sum_{i=0}^{l} L_i(\alpha^{j\,k_i})$. Как известно [11], элемент α^{k_i} является корнем соответствующего минимального многочлена степени m_i и, следовательно, лежит в подполе $GF(2^{m_i})$, $m_i \mid m$. Таким образом, все величины $(\alpha^{k_i})^j$ принадлежат полю $GF(2^{m_i})$ и могут быть разложены в каком-либо базисе $(\pi_{i,0}, \ldots, \pi_{i,m_i-1})$ этого поля: $\alpha^{j\,k_i} = \sum_{s=0}^{m_i-1} a_{ijs}\pi_{i,s}, a_{ijs} \in GF(2)$. Тогда значения каждого из линеаризованных многочленов могут быть вычислены в базисных точках соответствующего подполя по формуле

$$L_i(\pi_{i,s}) = \sum_{p=0}^{m_i-1} \pi_{i,s}^{2^p} f_{k_i 2^p}, \quad i \in [0, l], s \in [0, m_i - 1]. \tag{3.10}$$

Базисы $(\pi_{i,0}, \ldots, \pi_{i,m_i-1})$ для каждого из линеаризованных многочленов $L_i(y)$ могут выбираться независимо.

В соответствии с теоремой 3.3 компоненты преобразования Фурье многочлена $f(x)$ являются линейными комбинациями этих значений

$$F_j = f(\alpha^j) = \sum_{i=0}^{l} \sum_{s=0}^{m_i-1} a_{ijs} L_i(\pi_{i,s}) = \tag{3.11}$$

$$\sum_{i=0}^{l} \sum_{s=0}^{m_i-1} a_{ijs} \left(\sum_{p=0}^{m_i-1} \beta_{i,s}^{2^p} f_{k_i 2^p} \right), \quad j \in [0, n-1].$$

Последнее выражение может быть записано в матричной форме как $F = ALf$, где $F = (F_0, F_1, \ldots, F_{n-1})^T$, $f = (f_0, f_{k_1}, f_{k_1 2}, f_{k_1 2^2}, \ldots, f_{k_1 2^{m_1-1}}, \ldots, f_{k_l}, f_{k_l 2}, f_{k_l 2^2}, \ldots, f_{k_l 2^{m_l-1}})^T$ есть перестановка вектора коэффициентов исходного многочлена $f(x)$, соответствующая разложению (3.9), A — матрица, составленная из элементов $a_{ijs} \in GF(2)$, L — блочно-диагональная матрица, составленная из элементов $\pi_{i,s}^{2^p}$. Очевидно, что для линеаризованных многочленов одинаковой степени m_i, входящих в разложение (3.9), можно выбрать одинаковые базисы $(\pi_{i,s})$ в подполях $GF(2^{m_i})$, вследствие чего матрица L будет содержать большое число одинаковых блоков.

Таким образом, задача БПФ разбивается на два этапа: умножение блочно-диагональной матрицы L на исходный вектор f и умножение двоичной матрицы A на полученный вектор $S = Lf$

$$F = ALf. \tag{3.12}$$

Рассмотрим более подробно первый этап преобразования Фурье — задачу вычисления произведения $S = Lf$. Блочно-диагональная матрица

$$L = \begin{pmatrix} L_0 & 0 & \dots & 0 \\ 0 & L_1 & \dots & 0 \\ \dots & \dots & \ddots & \dots \\ 0 & 0 & \dots & L_l \end{pmatrix}$$

состоит из блоков

$$L_i = \begin{pmatrix} \pi_{i,0} & \pi_{i,0}^2 & \dots & \pi_{i,0}^{2^{m_i-1}} \\ \pi_{i,1} & \pi_{i,1}^2 & \dots & \pi_{i,1}^{2^{m_i-1}} \\ \dots & \dots & \dots & \dots \\ \pi_{i,m_i-1} & \pi_{i,m_i-1}^2 & \dots & \pi_{i,m_i-1}^{2^{m_i-1}} \end{pmatrix}.$$

Выберем в качестве $(\pi_{i,0}, \dots, \pi_{i,m_i-1})$ нормальный базис $\left(\gamma_i, \gamma_i^2, \dots, \gamma_i^{2^{m_i-1}}\right)$. Тогда матрица L состоит из блоков вида

$$L_i = \begin{pmatrix} \gamma_i^{2^0} & \gamma_i^2 & \dots & \gamma_i^{2^{m_i-1}} \\ \gamma_i^2 & \gamma_i^4 & \dots & \gamma_i^{2^0} \\ \dots & \dots & \dots & \dots \\ \gamma_i^{2^{m_i-1}} & \gamma_i^{2^0} & \dots & \gamma_i^{2^{m_i-2}} \end{pmatrix}.$$

В силу блочно-диагональной структуры матрицы L вычисление произведения $S = Lf$ может быть представлено как $S = (b_0, b_1, \dots, b_l)^T = L\,(a_0, a_1, \dots, a_l)^T$, где $b_i = (b_{i,0}, b_{i,1}, \dots, b_{i,m_i-1})$ — подвектора искомого вектора S, $a_i = (a_{i,0}, a_{i,1}, \dots, a_{i,m_i-1})$ — подвектора исходного вектора f.

Представим вычисление $b_i^T = L_i a_i^T$ как циклическую свертку

$$b_i(x) = b_{i,0} + b_{i,m_i-1}x + \ldots + b_{i,1}x^{m_i-1} = (\gamma_i + \gamma_i^{2^{m_i-1}}x + \ldots + \gamma_i^2 x^{m_i-1})(a_{i,0} + a_{i,1}x + \ldots + a_{i,m_i-1}x^{m_i-1}) \bmod (x^{m_i} - 1).$$

Для ее вычисления могут быть применены известные алгоритмы [5, 7, 1]. При этом использование свойства нормального базиса $\gamma_i + \gamma_i^2 + \ldots + \gamma_i^{2^{m_i-1}} = 1$ позволяет заметно сократить число операций при вычислении циклической свертки. Отметим, что вычисление значений линеаризованных многочленов с помощью циклической свертки было описано в монографии [7].

Описанный подход позволяет свести задачу умножения блочно-диагональной матрицы L на исходный вектор f над $GF(2^m)$ к задаче вычисления $l + 1$ циклических сверток малой длины m_i. Существующие алгоритмы вычисления циклических сверток

$b_i(x) = \gamma_i(x)a_i(x) \bmod (x^{m_i} - 1)$ могут быть записаны в матричном виде как

$$b_i = \begin{pmatrix} b_{i,0} \\ b_{i,1} \\ \dots \\ b_{i,m_i-1} \end{pmatrix} = Q_i \left(\left(D_i \begin{pmatrix} \gamma_i \\ \gamma_i^{2^{m_i-1}} \\ \dots \\ \gamma_i^2 \end{pmatrix} \right) \cdot (P_i a_i) \right), \tag{3.13}$$

где Q_i, D_i и P_i являются двоичными матрицами, а $x \cdot y$ обозначает покомпонентное произведение векторов. Очевидно, что вектор $C_i = D_i \left(\gamma_i, \gamma_i^{2^{m_i-1}}, \dots, \gamma_i^2\right)^T$ может быть вычислен заранее. Таким образом, выражение (3.12) может быть переписано как

$$F = AQ\left(C \cdot (Pf)\right), \tag{3.14}$$

где Q — двоичная блочно-диагональная матрица объединенных последующих сложений для $l+1$ циклической свертки, C — объединенный вектор констант, P — двоичная блочно-диагональная матрица объединенных предварительных сложений.

Учитывая формулы (3.12) и (3.14), второй этап БПФ может рассматриваться как умножение двоичной матрицы AQ на вектор $C \cdot (Pf)$. Для вычисления произведения $(AQ)\left(C \cdot (Pf)\right)$ могут быть использованы модифицированный алгоритм "четырех русских" (В.Л.Арлазаров, Е.А.Диниц, М.А.Кронрод, И.А.Фараджев) для умножения булевых матриц со сложностью $O(n^2/\log n)$ сложений над элементами поля $GF(2^m)$ [2] или специализированный алгоритм, основанный на итеративном алгоритме декодирования низкоплотностных кодов.

Как было показано, задача вычисления БПФ многочлена над $GF(2^m)$ сводится к набору задач вычисления циклической свертки многочленов и умножению на двоичную матрицу. Предполагая, что сложность вычисления короткой циклической свертки в конечном поле асимптотически равна $C_{conv}(t) = O(m \log^t m), t \geq 2$, получим, что сложность всего алгоритма равна

$$\begin{aligned} C = \ & \textstyle\sum_{i=0}^{l} C_{conv}(m_i) + C_{bm} = \sum_{i=0}^{l} O(m_i \log^t m_i) + C_{bm} = \\ & O(\tfrac{n}{m} m \log^t m) + C_{bm} = O(n \log^t \log n) + C_{bm}. \end{aligned}$$

Здесь C_{bm} — сложность умножения на двоичную матрицу. Асимптотическая оценка сложности процедуры умножения на двоичную матрицу, порождаемой вышеописанным алгоритмом в настоячщее время неизвестна и требует дальнейших исследований.

Пример 3.9. Продолжим рассмотрение БПФ длины 7 над полем $GF(2^3)$. Пусть α — корень примитивного многочлена $x^3 + x + 1$. В качестве базиса поля $GF(2^3)$ выберем нормальный базис $(\gamma, \gamma^2, \gamma^4)$, где $\gamma = \alpha^3$. Разложим многочлен $f(x)$ как в примере 3.8 и представим компоненты преобразования Фурье в виде сумм

$$
\begin{aligned}
f(\alpha^0) = &\ L_0(\alpha^0) + L_1(\alpha^0) + L_2(\alpha^0) = &\ L_0(1) + L_1(\gamma) + L_1(\gamma^2) + L_1(\gamma^4) + \\
& & L_2(\gamma) + L_2(\gamma^2) + L_2(\gamma^4) \\
f(\alpha^1) = &\ L_0(\alpha^0) + L_1(\alpha) + L_2(\alpha^3) = &\ L_0(1) + L_1(\gamma^2) + L_1(\gamma^4) + L_2(\gamma) \\
f(\alpha^2) = &\ L_0(\alpha^0) + L_1(\alpha^2) + L_2(\alpha^6) = &\ L_0(1) + L_1(\gamma) + L_1(\gamma^4) + L_2(\gamma^2) \\
f(\alpha^3) = &\ L_0(\alpha^0) + L_1(\alpha^3) + L_2(\alpha^2) = &\ L_0(1) + L_1(\gamma) + L_2(\gamma) + L_2(\gamma^4) \\
f(\alpha^4) = &\ L_0(\alpha^0) + L_1(\alpha^4) + L_2(\alpha^5) = &\ L_0(1) + L_1(\gamma) + L_1(\gamma^2) + L_2(\gamma^4) \\
f(\alpha^5) = &\ L_0(\alpha^0) + L_1(\alpha^5) + L_2(\alpha) = &\ L_0(1) + L_1(\gamma^4) + L_2(\gamma^2) + L_2(\gamma^4) \\
f(\alpha^6) = &\ L_0(\alpha^0) + L_1(\alpha^6) + L_2(\alpha^4) = &\ L_0(1) + L_1(\gamma^2) + L_2(\gamma) + L_2(\gamma^2).
\end{aligned}
$$

Эти тождества могут быть записаны в матричной форме как

$$
F = \begin{pmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} L_0(1) \\ L_1(\gamma) \\ L_1(\gamma^2) \\ L_1(\gamma^4) \\ L_2(\gamma) \\ L_2(\gamma^2) \\ L_2(\gamma^4) \end{pmatrix} = AS.
$$

Тогда задачу БПФ можно переписать в виде

$$
F = A \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \gamma^1 & \gamma^2 & \gamma^4 & 0 & 0 & 0 \\ 0 & \gamma^2 & \gamma^4 & \gamma^1 & 0 & 0 & 0 \\ 0 & \gamma^4 & \gamma^1 & \gamma^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \gamma^1 & \gamma^2 & \gamma^4 \\ 0 & 0 & 0 & 0 & \gamma^2 & \gamma^4 & \gamma^1 \\ 0 & 0 & 0 & 0 & \gamma^4 & \gamma^1 & \gamma^2 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_4 \\ f_3 \\ f_6 \\ f_5 \end{pmatrix}.
$$

Первый этап алгоритма БПФ состоит в вычислении двух циклических сверток

$$
\begin{pmatrix} b_{i,0} \\ b_{i,1} \\ b_{i,2} \end{pmatrix} = \begin{pmatrix} \gamma^1 & \gamma^2 & \gamma^4 \\ \gamma^2 & \gamma^4 & \gamma^1 \\ \gamma^4 & \gamma^1 & \gamma^2 \end{pmatrix} \begin{pmatrix} a_{i,0} \\ a_{i,1} \\ a_{i,2} \end{pmatrix}, \quad i = 1, 2,
$$

где

$$
S = \begin{pmatrix} L_0(1) \\ L_1(\gamma) \\ L_1(\gamma^2) \\ L_1(\gamma^4) \\ L_2(\gamma) \\ L_2(\gamma^2) \\ L_2(\gamma^4) \end{pmatrix} = \begin{pmatrix} b_{0,0} \\ b_{1,0} \\ b_{1,1} \\ b_{1,2} \\ b_{2,0} \\ b_{2,1} \\ b_{2,2} \end{pmatrix}, \qquad f = \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_4 \\ f_3 \\ f_6 \\ f_5 \end{pmatrix} = \begin{pmatrix} a_{0,0} \\ a_{1,0} \\ a_{1,1} \\ a_{1,2} \\ a_{2,0} \\ a_{2,1} \\ a_{2,2} \end{pmatrix}.
$$

Используя алгоритм вычисления трехточечной циклической свертки $b_i(x) = b_{i,0} + b_{i,2}x + b_{i,1}x^2 = (\gamma + \gamma^4 x + \gamma^2 x^2)(a_{i,0} + a_{i,1}x + a_{i,2}x^2) \bmod (x^3 - 1)$, представленный в [5], получим

$$b_i = \begin{pmatrix} b_{i,0} \\ b_{i,1} \\ b_{i,2} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix} \left(\left[\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} \gamma \\ \gamma^4 \\ \gamma^2 \end{pmatrix} \right] \cdot \left[\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_{i,0} \\ a_{i,1} \\ a_{i,2} \end{pmatrix} \right] \right) = Q_i \left(C_i \cdot (P_i a_i) \right), \quad i = 1, 2.$$

С учетом $\gamma + \gamma^2 + \gamma^4 = 1$ видно, что алгоритм требует 3 умножений, 4 предварительных и 5 последующих сложений.

Теперь можно записать формулу (3.14) для рассматриваемого примера в матричной форме

$$F = \begin{pmatrix} F_0 \\ F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \end{pmatrix} = \left(\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \right) \times$$

$$\left(\begin{pmatrix} 1 \\ 1 \\ \gamma^2 + \gamma^4 \\ \gamma + \gamma^4 \\ \gamma + \gamma^2 \\ 1 \\ \gamma^2 + \gamma^4 \\ \gamma + \gamma^4 \\ \gamma + \gamma^2 \end{pmatrix} \cdot \left[\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ f_4 \\ f_3 \\ f_6 \\ f_5 \end{pmatrix} \right] \right) = (AQ)(C \cdot (Pf)).$$

Второй этап БПФ состоит в умножении двоичной матрицы AQ на вектор $C \cdot (Pf)$

$$F = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \end{pmatrix} (C \cdot (Pf)).$$

Этот этап может быть выполнен за 17 сложений.

Таким образом, БПФ длины 7 сводится к следующей последовательности дей-

ствий:

выполнение предварительных сложений $P \times f$

$$
\begin{array}{ll}
V_1 = f_2 + f_4 & V_8 = f_6 + f_5 \\
V_2 = f_1 + f_2 & V_9 = f_3 + f_6 \\
V_3 = f_1 + f_4 & V_{10} = f_3 + f_5 \\
V_4 = f_1 + V_1 & V_{11} = f_3 + V_8,
\end{array}
$$

выполнение умножений на константы $C \cdot (Pf)$

$$
\begin{array}{ll}
V_5 = V_1\,\alpha & V_{12} = V_8\,\alpha \\
V_6 = V_2\,\alpha^2 & V_{13} = V_9\,\alpha^2 \\
V_7 = V_3\,\alpha^4 & V_{14} = V_{10}\,\alpha^4,
\end{array}
$$

умножение матрицы AQ на вектор $C \cdot (Pf)$

$$
\begin{array}{ll}
T_{10} = V_{12} + V_{14} & F_2 = T_8 + T_{11} \\
T_{11} = f_0 + V_{11} & F_3 = T_7 + T_{14} \\
T_{14} = f_0 + V_4 & T_{12} = F_2 + T_{10} \\
T_{15} = V_5 + V_6 & T_{13} = F_3 + T_{15} \\
T_{16} = V_6 + V_{13} & F_4 = T_7 + T_{12} \\
F_0 = V_4 + T_{11} & F_5 = T_{10} + T_{13} \\
T_9 = V_{12} + T_{16} & F_6 = F_5 + T_7 \\
T_7 = V_7 + T_9 & F_1 = F_4 + T_8. \\
T_8 = V_5 + T_9 &
\end{array}
$$

Общая сложность алгоритма составляет $2 \times 3 = 6$ умножений и $2 \times 4 + 17 = 25$ сложений, что на одно сложение меньше, чем для алгоритма, представленного в работе [8].

3.4 Операции над целыми числами

3.4.1 Представление целых чисел в ЭВМ

Наибольшее распространение получили позиционные системы счисления. В позиционной системе по основанию b произвольное положительное целое число представимо как

$$a = \sum_{i \geq 0} a_i b^i, 0 \leq a_i < b$$

Существует несколько различных подходов к представлению отрицательных чисел, которые оказывают влияние на метод реализации вычислительных операций:

1. *Прямой код* или абсолютное значение со знаком, например -1234. Недостатком является существование “плюс нуля” и “минус нуля”, т.е. двух кодов, обозначающих одно и то же число.

2. *Дополнительный код*, в котором отрицательное число $-a, a > 0$ представляется как $b^n - a$, где n — используемая разрядность. Например, в десятиразрядном десятичном представлении $-1 = (99999\,99999)_{10}$, а в двоичном 16-разрядном $-1 = (1111\,1111\,1111\,1111)_2$. Это эквивалентно вычислению по модулю b^n. Недостатком является несимметричность относительно нуля: для числа $-2^{n-1} = (100\ldots0)_2$ невозможно представить обратное.

3. *Обратный код*, в котором каждый разряд a_i модуля отрицательного числа a заменяется на $b-1-a_i$, например $-1 = (99999\,99998)_{10}$. Это эквивалентно вычислениям по модулю b^n-1. Недостатком снова является наличие двух представлений нуля.

В большинстве современных ЭВМ используется дополнительный двоичный код. Основным преимуществом является то, что в этом случае старший бит указывает знак числа. При этом $-a = (-1-a)+1$. Операция $-1-a$ может быть выполнена путем побитового инвертирования. Основным преимуществом дополнительного кода является то, что при сложении не требуется анализировать знаки операндов. Однако при умножении и делении учет знаков необходим. Это отражено в структуре системы команд большинства процессоров. В связи с этим далее ограничимся рассмотрением только операций сложения и умножения целых неотрицательных чисел. Интерес представляет эффективная реализация операций над числами с числом разрядом, превышающим разрядность ЭВМ.

При работе с числами с многократной точностью иногда используется их представление в виде системы вычетов, основывающееся на китайской теореме об остатках.

Теорема 3.4 (Китайская теорема об остатках для целых чисел)**.** *Пусть $m_1, \ldots, m_r$ — положительные целые взаимно простые числа. Пусть $m = m_1 \cdot \ldots \cdot m_r$ и пусть также $a, u_1, \ldots, u_r$ — целые числа. Тогда существует ровно одно целое число u:*

$$a \le u < a + m \wedge u \equiv u_j \bmod m_j, j = 1..r,$$

причем

$$u = a + ((u_1 M_1 + \ldots + u_r M_r - a) \bmod m),$$

где $M_j = n_j(n_j^{-1} \bmod m_j), n_j = m/m_j$.

3.4.2 Сложение

Рассмотрим реализацию сложения неотрицательных n-разрядных целых чисел $(u_{n-1}, \ldots, u_0)_b$ и $(v_{n-1}, \ldots, v_0)_b$ по основанию b. Следующий алгоритм формирует их сумму $(w_n, w_{n-1}, \ldots, w_0)_b$, причем $w_n \in \{0, 1\}$:

```
ADD(u, v, n)
 1  j ← 0; k ← 0;
 2  while j < n
 3      do w_j ← u_j + v_j + k
 4          if w_j ≥ b
 5             then w_j ← w_j − b
 6                  k = 1
 7             else  k = 0
 8
 9          j ← j + 1w_n = k
10  return (w_n, ..., w_0)
```

Заметим, что при работе этого алгоритма всегда выполняются соотношения $u_j + v_j + k \le (b-1) + (b-1) + 1 < 2b$. Т.к. данный алгоритм выполняет ровно n итераций, длина его входа равна n, а длина выхода $n+1$, какие-либо существенные ускорения процедуры сложения предложить сложно.

С целью максимального использования аппаратных возможностей ЭВМ целесообразно выбирать в качестве основания системы счисления 2^m, где m — разрядность ЭВМ. При этом приведение по модулю и вычисление переноса осуществляются автоматически.

Заметим, что если число представлено в виде системы вычетов, то сложение может производиться путем покомпонентного сложения вектора вычетов (модулярная арифметика), т.е. имеет место изоморфизм. Это может быть использовано для распараллеливания вычислений (Single Instruction Multiple Data).

3.4.3 Умножение

Умножение "в столбик"

Рассмотрим реализацию умножения неотрицательных целых чисел $(u_{m-1}, \ldots, u_0)_b$ и $(v_{n-1}, \ldots, v_0)_b$ по основанию b. Следующий алгоритм формирует их произведение $(w_{m+n-1}, \ldots, w_0)_b$:

MULTIPLY(u, v, m, n)

```
 1  w_j ← 0, j = 0..m − 1;
 2  j ← 0;
 3  while j < n
 4      do if v_j > 0
 5            then i ← 0; k ← 0;
 6                 while i < m
 7                     do t ← u_i * v_j + w_{i+j} + k;
 8                        w_{i+j} ← t mod b;
 9                        k ← ⌊t/b⌋;
10                        i ← i + 1;
11                 w_{j+m} = k;
12            else  w_{j+m} ← 0;
13         j ← j + 1; ;
14  return (w_{m+n−1}, ..., w_0)
```

На каждом шаге алгоритма умножения выполняются неравенства

$$0 \le t < b^2, 0 \le k < b.$$

На основе этих свойств можно оценить размер регистра, требуемый для хранения промежуточных значений. Если не производить обнуление выходного массива, то можно получить более общий алгоритм умножения с накоплением $w = u * v + w'$.

Алгоритм Карацубы

Умножение целых чисел может быть существенно ускорено. Например, пусть $u = b^n U_1 + U_0, v = b^n V_1 + V_0$. Тогда (*алгоритм Карацубы*)

$$\begin{aligned} uv = & (b^{2n} + b^n)U_1V_1 + b^n(U_1 - U_0)(V_0 - V_1) + (b^n + 1)U_0V_0 = \\ = & U_0V_0 + b^{2n}U_1V_1 + b^n((U_0 + V_0)(U_1 + V_1) - U_0V_0 - U_1V_1). \end{aligned}$$

Эта процедура может быть применена рекурсивно. В этом случае число элементарных умножений асимптотически сходится к $O(n^{\log_2 3})$.

Пример 3.10.

$351 = 13 * 27 = 3 * 7 + 100 * 1 * 2 + 10 * ((1 + 3) * (2 + 7) - 3 * 7 - 1 * 2) = 21 + 100 * 2 + 10 * (36 - 21 - 2) = 21 + 200 + 130 = 351$

Алгоритм Тоома-Кука

Еще более общий алгоритм быстрого умножения можно получить, если представить целые числа $u = (u_{(r+1)(n-1)}, \ldots, u_1, u_0)_b$ как значения многочлена $U(x) = \sum_{i=0}^{r} U_i x^i$ в точке b^r. Тогда $w = uv = W(b^r), W(x) = U(x)V(x)$. В этом случае задача сводится к поиску эффективного способа нахождения коэффициентов $W(x)$. Они могут быть найдены путем вычисления $W(j) = U(j)V(j)$ в некоторых точках j, для которых это может быть сделано сравнительно легко, и последующего восстановления $W(x)$ с помощью интерполяционного полинома Ньютона

$$W(x) = W(j_1) + W(j_1; j_2)(x - j_1) + \ldots + W(j_1; \ldots; j_n)(x - j_1)\ldots(x - j_n),$$

где $W(j_1; j_2)$ и т.д. есть разделенные разности. Как правило, j выбирается равным $0, 1, 2, 3$ и т.п. малым числам, умножение на которые сводится к простым операциям.

Пример 3.11. Рассмотрим быстрое вычисление $2888794 = 1234 * 2341 = (0100\,1101\,0010)_2 * (1001\,0010\,0101)_2$. Аргументы могут быть представлены в виде многочленов $U(x) = 4x^2 + 13x + 2, V(x) = 9x^2 + 2x + 5$. Тогда имеем:

$$\begin{array}{ccccccc} U(0) = 2 & & U(1) = 19 & & U(2) = 44 & U(3) = 77 & U(4) = 118 \\ V(0) = 5 & & V(1) = 16 & & V(2) = 45 & V(3) = 92 & V(4) = 157 \\ W(0) = 10 & & W(1) = 304 & & W(2) = 1980 & W(3) = 7084 & W(4) = 18526 \end{array}$$

Построение интерполяционного полинома Ньютона:

$$\begin{array}{rrlll} 10 & & & & \\ & 249 & & & \\ 304 & & 1382/2 = 691 & & \\ & 1676 & & 1023/3 = 341 & \\ 1980 & & 3428/2 = 1714 & & 144/4 = 36 \\ & 5104 & & 1455/3 = 485 & \\ 7084 & & 6338/2 = 3169 & & \\ & 11442 & & & \\ 18526 & & & & \end{array}$$

Таким образом, $W(x) = (((36 * (x - 3) + 341) * (x - 2) + 691) * (x - 1) + 294) * x + 10 = 36x^4 + 125x^3 + 64x^2 + 69x + 10$, т.е. ответом будет $W(16)$. При этом необходимо отметить, что подстановка $x = 16$ сводится к умножению чисел малой разрядности и сдвигам.

Вышеописанный алгоритм носит название алгоритма *Тоома-Кука*.

Алгоритмы Карацубы и Тоома-Кука сводят задачу умножения двух чисел к нескольким задачам умножения чисел меньшей разрядности. Разбиение может осуществляться рекурсивно до тех пор, пока разрядность не уменьшится до поддерживаемой аппаратно. С целью избежания издержек, связанных с организацией рекурсии, промежуточные данные организуются в виде стека, обрабатываемого итеративно. Асимптотическая сложность алгоритма Тоома-Кука составляет $O(n2^{\sqrt{2\log n}}\log_2 n)$

Алгоритм Шёнхаге

Используя представление целых чисел в виде системы вычетов по специально подобранным модулям, можно построить алгоритм перемножения чисел со сложностью $O(n^{\log_3 6})$ (*алгоритм Шёнхаге*). При этом используется свойство изоморфизма $(uv) \equiv (u_j v_j) \bmod m_j$. В общем случае, наибольшую сложность при этом подходе представляет восстановление числа по набору вычетов. За счет использования специально подобранных модулей этот этап может быть реализован очень просто.

Умножение с помощью быстрого преобразования Фурье

В алгоритме Тоома-Кука интерполяционные точки $j \in \mathbb{Z}$ могут быть заменены на $x_j = \exp(2\pi ij/K), j = 0..K-1, i^2 = -1$. Оказывается, что в этом случае вычисление значений многочлена может быть выполнено весьма эффективно (см. п. 3.3.2). При этом неизбежно возникновение ошибок округления, которые, однако, оказываются весьма незначительными. Асимптотическая сложность этого метода (*Шёнхаге-Штрассена*) равна $O(n\log_2 n\log_2\log_2 n)$.

3.4.4 Деление

Основная сложность при реализации классического метода деления “в столбик” состоит в необходимости угадывать разряды частного. Этот процесс должен быть формализован. Прежде всего заметим, что деление m-разрядного числа на n-разрядное, $m > n$, сводится к последовательности делений $n+1$-разрядных чисел u на n-разрядное число v, причем $0 \le u/v < b$, где b — основание системы счисления. Таким образом, необходимо построить алгоритм для нахождения $q = \lfloor u/v \rfloor, u = (u_n, u_{n-1}, \ldots, u_0)_b, v = (v_{n-1}, \ldots, v_0)_b$.

Условие $u/v < b$ может быть переформулировано как $u/b < v$, т.е. $(u_n, u_{n-1}, \ldots, u_1)_b < (v_{n-1}, \ldots, v_0)_b$. q должно быть единственным целым числом, таким, что $0 \le u - qv < v$. Попытаемся угадать q как

$$\hat{q} = \min\left(\left\lfloor \frac{u_n b + u_{n-1}}{v_{n-1}} \right\rfloor, b-1\right). \tag{3.15}$$

Попытаемся оценить, насколько хорошо такое приближение.

Лемма 3.1. $\hat{q} \geq q$.

Доказательство. Т.к. $q \leq b-1$, утверждение верно при $\hat{q} = b-1$. В противном случае $\hat{q} \leq \frac{u_n b + u_{n-1}}{v_{n-1}} < \hat{q}+1 \Rightarrow \hat{q}v_{n-1} \geq u_n b + u_{n-1} - v_{n-1} + 1$. Следовательно, $u - \hat{q}v \leq u - \hat{q}v_{n-1}b^{n-1} \leq u_n b^n + \ldots + u_0 - (u_n b^n + u_{n-1}b^{n-1} - v_{n-1}b^{n-1} + b^{n-1}) = u_{n-2}b^{n-2} + \ldots + u_0 - b^{n-1} + v_{n-1}b^{n-1} < v_{n-1}b^{n-1} \leq v$. Тогда $\hat{q}v > u - v \geq qv - v = (q-1)v$, т.е. $\hat{q} > q-1$. $\square$

Докажем теперь, что если $\hat{q}$ превышает q, то превышение незначительно.

Теорема 3.5. *Если* $v_{n-1} \geq \lfloor b/2 \rfloor$, *то* $\hat{q} - 2 \leq q \leq \hat{q}$

Доказательство. Предположим, что это не так, т.е. $\hat{q} \geq q+3$. Тогда

$$\hat{q} \leq \frac{u_n b + u_{n-1}}{v_{n-1}} = \frac{u_n b^n + u_{n-1}b^{n-1}}{v_{n-1}b^{n-1}} \leq \frac{u}{v_{n-1}b^{n-1}} < \frac{u}{v - b^{n-1}}.$$

Случай $v = b^{n-1}$ невозможен, т.к. тогда $q = \hat{q}$. Т.к. $q+1 > u/v$,

$$3 \leq \hat{q} - q < \frac{u}{v - b^{n-1}} - \frac{u}{v} + 1 = \frac{u}{v}\left(\frac{b^{n-1}}{v - b^{n-1}}\right) + 1.$$

Тогда $\frac{u}{v} > 2\frac{v - b^{n-1}}{b^{n-1}} \geq 2(v_{n-1} - 1)$. Т.к. $b - 4 \geq \hat{q} - 3 \geq q = \lfloor u/v \rfloor \geq 2(v_{n-1} - 1)$, $v_{n-1} < \lfloor b/2 \rfloor$. $\square$

Условие этой теоремы носит название условия нормализации. Его можно обеспечить, домножив делимое и делитель на $\lfloor b/(v_{n-1}+1) \rfloor$.

Кроме того, можно показать, что если $\hat{q}v_{n-2} > b\hat{r} + u_{n-2}$, то $q < \hat{q}$, где $\hat{r} = u_n b + u_{n-1} - \hat{q}v_{n-1}$. В противном случае $q \in \{\hat{q}, \hat{q}-1\}$.

Рассмотрим вычисление частного и остатка от деления $(u_{m+n-1}, \ldots, u_0)_b$ на $(v_{n-1}, \ldots, v_0)$.

DIVIDE(u, v, m, n)

```
 1  d ← ⌊b/(v_{n-1} + 1)⌋;
 2  (u_{m+n}, u_{m+n-1} ..., u_0)_b ← d * (u_{m+n-1} ..., u_0)_b
 3  (v_{n-1} ..., v_0)_b ← d * (v_{n-1} ..., v_0)_b
 4  j ← m
 5  while j ≥ 0
 6      do q̂ ← ⌊(u_{j+n} b + u_{j+n-1})/v_{n-1}⌋
 7         r̂ ← (u_{j+n} b + u_{j+n-1}) mod v_{n-1}
 8         if (q̂ = b) ∨ (q̂ v_{n-2} > b r̂ + u_{j+n-2})
 9           then q̂ ← q̂ - 1
10                r̂ ← r̂ + v_{n-1}
11         if (r̂ < b) ∨ (q̂ v_{n-2} > b r̂ + u_{j+n-2})
12           then q̂ ← q̂ - 1
13                r̂ ← r̂ + v_{n-1}
14         (u_{j+n} ..., u_j)_b ← (u_{j+n} ... u_j)_b - q̂ (v_{n-1} ... v_1 v_0)_b
```

15 **if** $(u_{j+n} \dots u_j)_b < 0$
16 **then** $NegFlag \leftarrow true$
17 $(u_{j+n} \dots u_j)_b \leftarrow (u_{j+n} \dots u_j)_b + b^{n+1}$
18 **else** $NegFlag \leftarrow false$
19 $q_j = \hat{q}$
20 **if** $NegFlag = true$
21 **then** $q_j \leftarrow q_j - 1$
22 $(u_{j+n} \dots u_j)_b \leftarrow (u_{j+n} \dots u_j)_b + (0 v_{n-1} \dots v_1 v_0)_b$
23 $j \leftarrow j - 1$
24 **return** $((q_m \dots q_1 q_0)_b, (u_{n-1} \dots u_1 u_0)_b / d)$

Некоторые фрагменты этого алгоритма выполняются очень редко, что затрудняет отладку.

3.4.5 Возведение в степень

Рассмотрим задачу эффективного вычисления $y = x^n, n \in \mathbb{N}$. Пусть $n = \sum_{i \geq 0} n_i 2^i, n_i \in \{0, 1\}$. Тогда

$$y = x^{\sum_{i \geq 0} n_i 2^i} = \prod_{i \geq 0} x^{n_i 2^i} = \prod_{i \geq 0} \left(x^{2^i} \right)^{n_i}. \tag{3.16}$$

Заметим, что $x^{2^i} = x^{2^{i-1}} x^{2^{i-1}}$, а значение n_i указывает, должен ли сомножитель x^{2^i} учитываться при вычислении y. Таким образом, возведение в степень может быть выполнено за

$$\log_2(n) + \sum_{i \geq 0} n_i$$

операций умножения (*бинарный алгоритм возведения в степень*).

Двоичный метод возведения в степень не является оптимальным. Например, вычисление x^{15} требует 6 умножений, в то время как $z = x^3$ может быть вычислено с помощью 2 умножений, а $y = z^5$ — с помощью еще трех, т.е. всего необходимо 5 умножений. Таким образом, если $n = pq$, задача возведения в степень может быть разделена на две подзадачи, каждая из которых может быть решена двоичным методом. Как правило, этот метод несколько лучше двоичного, но не всегда (например, при $n = 33$).

3.5 Основные результаты

1. Алгоритм Штрассена перемножения матриц.
2. Алгоритмы Тоома-Кука и Карацубы вычисления свертки.
3. Алгоритм Винограда и границы сложности вычисления свертки.
4. Алгоритм Агарвала-Кули и итеративный метод вычисления свертки.

5. Алгоритмы Кули-Тьюки (Cooley-Tukey) и Гуда-Томаса БПФ.
6. Алгоритмы Блюстейна и Рейдера вычисления БПФ простой длины.
7. Применение свойств конечных полей для ускорения БПФ.
8. Представление целых чисел в ЭВМ
9. Применение алгоритмов свертки для умножения целых чисел
10. Возведение целых чисел в степень.

Упражнения

1. Построить алгоритм шеститочечной циклической свертки двумя способами над вещественным и конечным полями.
2. Написать программу, генерирующую билинейные формы алгоритма Винограда перемножения многочленов по модулю заданного многочлена $\pi(x)$.
3. Доказать, что если $\pi_0, \ldots, \pi_{m-1}$ — базис поля $GF(2^m)$, то матрица $\begin{pmatrix} \pi_0 & \pi_0^2 & \ldots & \pi_0^{2^{m-1}} \\ \pi_1 & \pi_1^2 & \ldots & \pi_1^{2^{m-1}} \\ \ldots & & & \\ \pi_{m-1} & \pi_{m-1}^2 & \ldots & \pi_{m-1}^{2^{m-1}} \end{pmatrix}$ обратима и ее определитель равен 1.
4. Построить алгоритм Кули-Тьюки БПФ длины 8 над комплексным полем.
5. Построить алгоритм Гуда-Томаса БПФ длины 6 над комплексным полем.
6. Построить алгоритм Рейдера БПФ длины 7 над комплексным полем и полем $GF(2^3)$.
7. Построить алгоритм Ванга-Жу БПФ длины 7 над полем $GF(2^3)$.
8. Построить циклотомический алгоритм БПФ длины 5 и 15 над полем $GF(2^4)$.
9. Вычислить 156616*879521 с помощью алгоритма Карацубы.

Глава 4

Введение в алгебраическую геометрию и коммутативную алгебру

В этом разделе представлен краткий обзор некоторых задач, возникающих при рассмотрении полиномиальных уравнений и связанных с ними объектов [10].

4.1 Идеалы и аффинные многообразия

4.1.1 Основные понятия

Определение 4.1. *Мономом* (одночленом) от переменных $x_1, \ldots, x_n$ называется произведение вида

$$x_1^{\alpha_1} x_2^{\alpha_2} \ldots x_n^{\alpha_n},$$

где показатели степеней — целые неотрицательные числа. Степенью монома называется $\sum_{i=1}^{n} \alpha_i$.

Если $\alpha = (\alpha_1, \ldots, \alpha_n)$ — вектор показателей степеней, то для удобства будем обозначать соответствующий моном как x^α.

Определение 4.2. *Полиномом* f от переменных $x_1, \ldots, x_n$ над полем $\mathbb{F}$ называется конечная линейная комбинация мономов с коэффициентами из $\mathbb{F}$:

$$f = \sum_{\alpha} a_\alpha x^\alpha.$$

a_α называется коэффициентом монома x^α. Если $a_\alpha \neq 0$, $a_\alpha x^\alpha$ называется членом полинома f. Степенью многочлена называется максимум степеней всех его членов.

Полином, не имеющий членов, называется нулевым. Несложно показать, что множество многочленов образует кольцо, обозначаемое как $\mathbb{F}[x_1, \ldots, x_n]$.

Определение 4.3. Пусть дано поле $\mathbb{F}$ и натуральное число n; тогда n-мерным *аффинным пространством* называется множество

$$\mathbb{F}^n = \{(a_1, \ldots, a_n) | a_i \in \mathbb{F}, i = 1..n\}.$$

$\mathbb{F}^1$ называется аффинной прямой, а $\mathbb{F}^2$ —аффинной плоскостью. Т.к. полином задает функцию $\mathbb{F}^n \longrightarrow F$, возникает связь между алгеброй и геометрией. Двойная природа полинома иногда приводит к странным явлениям. Например, вопрос "верно ли, что $f = 0$?" (другими словами, является ли f нулевым полиномом) не эквивалентен вопросу "равно ли $f(\alpha_1, \ldots, \alpha_n) = 0$?" (является ли он нулевой функцией). В качестве примера можно привести многочлен $x^2 - x$ их $GF(2)[x]$.

Теорема 4.1. *Пусть $\mathbb{F}$ — бесконечное поле и $f \in \mathbb{F}[x_1, \ldots, x_n]$. Тогда $f = 0$ в $\mathbb{F}[x_1, \ldots, x_n]$ тогда и только тогда, когда $f : \mathbb{F}^n \longrightarrow F$ является нулевой функцией.*

Доказательство. Если $f = 0$, то утверждение очевидно. В обратную сторону доказательство по индукции от противного. Пусть $n = 1$. Ненулевой полином степени m имеет не более m различных корней. Т.к. число членов многочлена конечно, число корней его также конечно. Следовательно, $\exists a \in \mathbb{F} : f(a) \neq 0$, что противоречит предположению о том, что f — нулевая функция. Если утверждение теоремы справедливо для многочленов от $n-1$ переменных и пусть $f \in \mathbb{F}[x_1, \ldots, x_n]$ обращается в нуль во всех точках $\mathbb{F}^n$. Тогда $\forall (a_1, \ldots, a_{n-1}) \in \mathbb{F}^{n-1}\, g(y) = f(a_1, \ldots, a_{n-1}, y) = \sum_i f_i(a_1, \ldots, a_{n-1}) x^i$ обращается в ноль для всех $y \in \mathbb{F}$. Следовательно, он является нулевым полиномом. Следовательно, $\forall a_1, \ldots, a_{n-1} : f_i(a_1, \ldots, a_{n-1}) = 0$. Тогда по индукционному предположению $f_i(x_1, \ldots, x_{n-1})$ являются нулевыми полиномами, а следовательно, и f является нулевым полиномом. □

Отсюда следует, что в бесконечном поле два полинома равны тогда и только тогда, когда они определяют одну и ту же функцию. Поле $\mathbb{F}$ называется *алгебраически замкнутым*, если любой непостоянный полином из $\mathbb{F}[x]$ имеет в нем корень. Примером алгебраически незамкнутого поля является поле вещественных чисел, а примером алгебраически замкнутого — поле комплексных чисел.

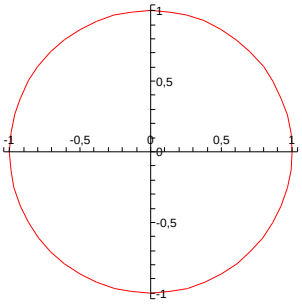
Определение 4.4. Пусть $f_1, ..., f_s \in \mathbb{F}[x_1, \ldots, x_n]$. *Аффинным многообразием*, определяемым полиномами $f_1, \ldots, f_s$ называется

$$V(f_1, \ldots, f_s) = \{(a_1, \ldots, a_n) \in \mathbb{F}^n \,|\, f_i(a_1, \ldots, a_n) = 0, i = 1..s\}.$$

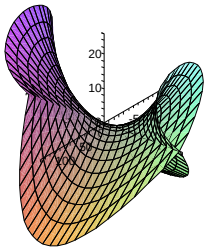
Пример 4.1. На рисунке 4.1 приведены примеры аффинных многообразий.

Определение 4.5. Подмножество $I \subset \mathbb{F}[x_1, \ldots, x_n]$ называется *идеалом*, если выполнены следующие условия:

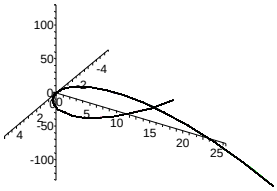
1. $0 \in I$



(a) $V(x^2+y^2-1)$



(b) $V(x^2-y^2z^2+z^3)$



(c) $V(y-x^2, z-x^3)$ (скрученная кубика)

Рис. 4.1: Примеры аффинных многообразий

2. $f, g \in I \Rightarrow f+g \in I$

3. $f \in I \wedge h \in \mathbb{F}[x_1, \ldots, x_n] \Rightarrow hf \in I$.

Лемма 4.1. *Пусть* $f_1, \ldots, f_s \in \mathbb{F}[x_1, \ldots, x_n]$. *Положим*

$$< f_1, \ldots, f_s >= \left\{ \sum_{i=1}^{s} h_i f_i | h_i \in \mathbb{F}[x_1, \ldots, x_n] \right\}$$

Множество $I =< f_1, \ldots, f_s >$ *является идеалом.*

Множество I называется идеалом, порожденным полиномами (базисом) $f_1, \ldots, f_s$.

Доказательство. Ясно, что $0 \in I$. Пусть $f, g \in I \Rightarrow f = \sum_{i=1}^{s} h_i f_i, g = \sum_{i=1}^{s} q_i f_i \Rightarrow f+g = \sum_{i=1}^{s}(q_i+h_i)f_i \in I$. $f \in I \Rightarrow f = \sum_{i=1}^{s} h_i f_i \Rightarrow hf = \sum_{i=1}^{s}(hh_i)f_i \in I$. □

Заметим, что если задана система полиномиальных уравнений $f_1 = 0, \ldots, f_s = 0$, то всякий элемент идеала соответствует уравнению $h_1 f_1 + \ldots + h_s f_s = 0$. Можно заметить, что понятие идеала весьма близко к понятию линейного пространства. Однако идеал, обладающий конечным базисом, как векторное пространство обладает бесконечным базисом.

Лемма 4.2. *Пусть* $V \in \mathbb{F}^n$ *— аффинное многообразие. Положим* $I(V) = \{f \in \mathbb{F}[x_1, \ldots, x_n] | \forall (a_1, \ldots, a_n) \in V : f(a_1, \ldots, a_n) = 0\}$. $I(V)$ *является идеалом.*

Доказательство. Ясно, что $0 \in I(V)$. Пусть $f, g \in I(V), h \in \mathbb{F}[x_1, \ldots, x_n]$. По условию леммы, $\forall (a_1, \ldots, a_n) \in V : f(a_1, \ldots, a_n) = 0, g(a_1, \ldots, a_n) = 0 \Rightarrow f(a_1, \ldots, a_n) + g(a_1, \ldots, a_n) = 0, h(a_1, \ldots, a_n) f(a_1, \ldots, a_n) = 0$. □

Пример 4.2. Докажем, что $I(\{(0,0)\}) =< x, y >$. Ясно, что $\forall A(x,y), B(x,y) : A(x,y)x + B(x,y)y|_{x=0,y=0} = 0$. Пусть $f(x,y) = \sum_{i,j} a_{ij}x^i y^j : f(0,0) = 0 \Rightarrow f = a_{00} + \sum_{i+j>0} a_{ij}x^i y^j =$

$$0 + \sum_{i+j>0} a_{ij}x^i y^j = x\left(\sum_{i>0,j\geq 0} a_{ij}x^{i-1}y^j\right) + y\left(\sum_{j>0,i\geq 0} a_{ij}x^{i-1}y^j\right) \in < x, y >$$

Лемма 4.3. *Пусть* $f_1, \ldots, f_s \in \mathbb{F}[x_1, \ldots, x_n]$. *Тогда* $< f_1, \ldots, f_s > \subset I(V(f_1, \ldots, f_s))$, *но эти идеалы не всегда совпадают.*

Доказательство. Пусть $f \in < f_1, \ldots, f_s > \Rightarrow f = \sum_{i=1}^{s} h_i f_i$. Т.к. $\forall (a_1, \ldots, a_n) \in V(f_1, \ldots, f_s) : f_i(a_1, \ldots, a_n) = 0$, $f(a_1, \ldots, a_n) = 0 \Rightarrow f \in I(V(f_1, \ldots, f_s))$. В качестве примера несовпадения этих идеалов докажем строгое включение $< x^2, y^2 > \subset I(V(x^2, y^2))$. Ясно, что идеалом многообразия $\{(0,0)\}$ является $< x, y >$. Т.к. $\forall A(x,y), B(x,y)$ все члены полинома $A(x,y)x^2 + B(x,y)y^2$ имеют степень не менее двух, $x \notin < x^2, y^2 >$. □

Лемма 4.4. *Пусть* V, W *— аффинные многообразия. Тогда:*

1. $V \subset W \Leftrightarrow I(V) \supset I(W)$
2. $V = W \Leftrightarrow I(V) = I(W)$

Доказательство. Пусть $V \subset W$. Тогда всякий полином, равный нулю на V, равен нулю и на W, т.е. $I(W) \subset I(V)$. Если $I(W) \subset I(V)$, то многочлены $g_1, \ldots, g_s$, определяющие W, принадлежат $I(W)$, а значит, и $I(V)$. Т.к. W есть множество общих нулей многочленов, $V \subset W$. Второе утверждение очевидно. □

Можно сформулировать следующие задачи:

- Описание идеала: каждый ли идеал $I \subset \mathbb{F}[x_1, \ldots, x_n]$ является конечно порожденным?
- Принадлежность идеалу: существует ли алгоритм, позволяющий решить вопрос о принадлежности полинома f идеалу $< f_1, \ldots, f_s >$?
- Теорема о нулях: какова связь между $< f_1, \ldots, f_s >$ и $I(V(f_1, \ldots, f_s))$?
- Совместность полиномиальных уравнений: можно ли выяснить, будет ли $V(f_1, \ldots, f_s)$ непустым? Если оно непусто, можно ли описать аффинное многообразие V?
- Конечность: является ли аффинное многообразие конечным множеством и можно ли выписать все его элементы в явном виде?
- Размерность: какова размерность заданного аффинного многообразия?

UNIVARIATEDIVISION(f, g)
1 $q \leftarrow 0; r \leftarrow f;$
2 **while** $(r \neq 0) \wedge (\mathrm{LT}(g) | \mathrm{LT}(r))$
3 **do** $q \leftarrow q + \mathrm{LT}(r)/\mathrm{LT}(g)$
4 $r \leftarrow r - (\mathrm{LT}(r)/\mathrm{LT}(g))g$
5 **return** (q, r)

Рис. 4.2: Алгоритм деления многочленов от одной переменной

4.1.2 Полиномы от одной переменной

Известно, что в случае многочленов от одной переменной возможно их деление с остатком. Более конкретно, каждый многочлен $f \in \mathbb{F}[x]$ может быть представлен как $f = qg + r, \deg r < \deg g, q, g, r \in \mathbb{F}[x]$. Частное q и остаток r могут быть найдены с помощью алгоритма, приведенного на рисунке 4.2. Необходимо отметить, что данный алгоритм опирается на понятие старшего члена полинома $\mathrm{LT}(f)$. Далее будут также использоваться коэффициент при старшем члене $\mathrm{LC}(f)$ и старший моном $\mathrm{LM}(f)$.

Теорема 4.2. *Пусть $\mathbb{F}$ — поле. Каждый идеал $I \subset \mathbb{F}[x]$ может быть представлен в виде $< f >$ для некоторого f, определяемого однозначно с точностью до умножения на ненулевую константу из $\mathbb{F}$.*

Доказательство. Если $I = \{0\}$, $f = 0$ и утверждение доказано. Если $I \neq \{0\}$, выберем в качестве f ненулевой многочлен наименьшей степени, принадлежащий I. Тогда всякий другой многочлен из I делится на f без остатка, т.к. в противном случае возникает противоречие с условием минимальности степени f. □

Идеал, порожденный единственным элементом, называется *главным идеалом*. Кольцо $\mathbb{F}[x]$ является областью главных идеалов. Можно показать, что $\forall f, g \in= F[x] :< f, g >=< GCD(f, g) >$. Это утверждение может быть обобщено на произвольное число многочленов (но не переменных!!!). Таким образом, в кольце многочленов от одной переменной для произвольного идеала может быть построен базис, состоящий из единственного элемента h. Тогда критерием принадлежности произвольного многочлена $\phi \in \mathbb{F}[x]$ идеалу является равенство нулю остатка от деления ϕ на h.

Таким образом, в случае кольца многочленов от одной переменной решение задач, приведенных в начале главы, не представляет сложности[1]. В случае многочленов от нескольких переменных ситуация значительно сложнее.

[1]Необходимо отметить, что уравнения от одной переменной степени $t > 4$ неразрешимы в радикалах (теорема Абеля).

4.2 Базисы Гребнера

4.2.1 Упорядочение мономов в $\mathbb{F}[x_1, \ldots, x_n]$ и алгоритм деления

Анализ алгоритма деления многочленов от одной переменной и алгоритма Гаусса диагонализации матрицы приводит к выводу о том, что их основой является упорядочение одночленов или столбцов матрицы, которые последовательно обрабатываются в соответствии с этим упорядочением. В частности, алгоритм деления предполагает $\ldots > x^{m+1} > x^m > \ldots > x^2 > x > 1$. Алгоритм Гаусса предполагает $x_1 > x_2 > \ldots > x_n$. Можно предположить, что аналогичное упорядочение должно быть использовано и при обобщении алгоритма деления на случай многочленов от нескольких переменных.

Заметим, что существует взаимно однозначное соответствие между мономами $x^\alpha = (x_1^{t_1}, \ldots, x_n^{t_n}$ и векторами показателей степеней $(t_1, \ldots, t_n)$. Поэтому будем рассматривать упорядочение векторов из $\mathbb{Z}^n$. Не всякое упорядочение является подходящим, т.к. оно должно быть совместимо с алгебраической структурой полиномиального кольца. Кроме того, необходимо иметь возможность сравнивать любую пару мономов, причем должно выполняться ровно одно из следующих соотношений:

$$x^\alpha > x^\beta, x^\alpha = x^\beta, x^\alpha < x^\beta.$$

Это приводит к следующему определению:

Определение 4.6. *Мономиальным упорядочением* на $\mathbb{F}[x_1, \ldots, x_n]$ называется любое бинарное отношение $>$ на $\mathbb{Z}_{0+}^n$, обладающее следующими свойствами:

1. $>$ является отношением линейного (полного) порядка на $\mathbb{Z}_{0+}^n$.
2. Если $\alpha > \beta$ и $\gamma \in \mathbb{Z}_{0+}^n$, то $\alpha + \gamma > \beta + \gamma$.
3. $>$ вполне упорядочивает Z_{0+}^n, т.е. любое непустое подмножество Z_{0+}^n имеет минимальный (наименьший) элемент.

Лемма 4.5. *Отношение порядка $>$ на множестве A вполне упорядочивает это множество тогда и только тогда, когда каждая строго убывающая последовательность элементов $a_i \in A : a_1 > a_2 > \ldots$ обрывается.*

Доказательство. Доказательство в обе стороны от противного. Если $>$ не есть вполне упорядочение, то существует непустое подмножество $S \subset A$, не имеющее минимального элемента. Пусть a_1 — произвольный элемент из S. Т.к. он не минимален, в нем найдется $a_2 < a_1$ и т.д. В результате будет получена бесконечная строго убывающая последовательность.

Обратно, если существует бесконечная строго убывающая последовательность, то множество $\{a_1, a_2, \ldots\}$ непусто и не имеет минимального элемента, т.е. $>$ не является вполне упорядочением. □

Рассмотрим несколько примеров мономиальных упорядочений.

Определение 4.7. Пусть $\alpha = (\alpha_1, \ldots, \alpha_n), \beta = (\beta_1, \ldots, \beta_n) \in \mathbb{Z}_{0+}^n$. *Лексикографическим упорядочением* над $\mathbb{Z}_{0+}^n$ называется бинарное отношение $>_{lex}$, такое что $\alpha >_{lex} \beta$, если самая левая ненулевая координата вектора $\alpha - \beta$ положительна. В этом случае будем также писать $x^\alpha >_{lex} x^\beta$

Пример 4.3. $(1,2,0) >_{lex} (0,3,4)$; $(3,2,4) >_{lex} (3,2,1)$.

Теорема 4.3. *Лексикографическое упорядочение на $\mathbb{Z}_{0+}^n$ является мономиальным упорядочением.*

Доказательство. Иррефлексивность, антисимметричность и транзитивность отношения следуют непосредственно из определения. Полнота следует из полноты числового отношения порядка. Пусть $\alpha >_{lex} \beta$. Тогда самая левая ненулевая координата j_0 вектора $\alpha - \beta$ положительна. Но тогда $\forall \gamma \in \mathbb{Z}^n (\alpha + \gamma) - (\beta + \gamma) = \alpha - \beta$ имеет ту же самую крайнюю левую ненулевую координату j_0, т.е. $\alpha + \gamma >_{lex} \beta + \gamma$.

Предположим, что $>_{lex}$ не является вполне упорядочением. Тогда существует строго убывающая бесконечная последовательность $\alpha^{(1)} > \alpha^{(2)} > \ldots$. Рассмотрим первые координаты этих векторов. По определению лексикографического упорядочения, они образуют невозрастающую последовательность неотрицательных целых чисел. Т.к. множество неотрицательных целых чисел вполне упорядочено, эта последовательность начиная с некоторого момента перестает убывать. Начиная с этого элемента последовательности начнем рассматривать вторые компоненты и т.д. Перебрав все координаты получим, что последовательность $\alpha^{(1)} > \alpha^{(2)} > \ldots$ содержит одинаковые элементы. Из полученного противоречия следует, что $>_{lex}$ является вполне упорядочением. □

Определение 4.8. *Градуированным лексикографическим упорядочением* называется бинарное отношение, такое что $\alpha >_{grlex} \beta$, если $|\alpha| = \sum_{i=1}^n \alpha_i > |\beta| = \sum_{i=1}^n \beta_i \vee |a| = |\beta| \wedge \alpha >_{lex} \beta$

Определение 4.9. *Градуированным обратным лексикографическим упорядочением* называется бинарное отношение, такое что $\alpha >_{grevlex} \beta$, если $|\alpha| = \sum_{i=1}^n \alpha_i > |\beta| = \sum_{i=1}^n \beta_i$ или $|a| = |\beta|$ и самая правая ненулевая координата вектора $\alpha - \beta$ отрицательна.

Последнее упорядочение оказывается вычислительно наиболее эффективным. Оно НЕ является градуированным лексикографическим упорядочением с обратным порядком переменных.

Пример 4.4. $(4,7,4) >_{grevlex} (4,2,3); (1,5,2) >_{grevlex} (4,1,3)$

После того, как введено какое-либо мономиальное упорядочение, можно обобщить алгоритм деления многочленов на случай нескольких переменных. Описание этого алгоритма приведено на рисунке 4.3. Необходимо отметить, что работа алгоритма не завершается при обнаружении первого же члена, не делящегося на старший член делителей, как это было в случае многочленов от одной переменной. Такие члены переносятся в остаток, т.к. после них в делимом могут быть члены, делящиеся на один из f_i.

MULTIVARIATEDIVISION$(f, f_1, \ldots, f_s)$

```
 1  a_i ← 0, i = 1..s; r ← 0
 2  p ← f;
 3  while p ≠ 0
 4  do i ← 1; HaveDivision ← false;
 5      while !HaveDivision ∧ i ≤ s
 6      do if LT(f_i)| LT(p)
 7            then a_i ← a_i + LT(p)/ LT(f_i)
 8                 p ← p − (LT(p)/ LT(f_i))f_i
 9                 HaveDivision ← true;
10            else  i ← i + 1;
11      if !HaveDivision
12        then r ← r + LT(p)
13             p ← p − LT(p)
14             return (r, a_1, ..., a_s)
```

Рис. 4.3: Алгоритм деления f на $f_1, \ldots, f_s$

Пример 4.5. В качестве примера рассмотрим деление $f = xy^2+1$ на $f_1 = xy+1, f_2 = y+1$ с использованием лексикографического упорядочения при $x < y$. Путем несложных действий получим $xy^2 + 1 = y(xy + 1) + (-1)(y + 1) + 2$.

Для иллюстрайции "раннего образования остатка" рассмотрим пример деления $f = x^2y + xy^2 + y^2$ на $f_1 = xy - 1, f_2 = y^2 - 1$. Получим $x^2y + xy^2 + y^2 = (x + y)(xy - 1) + 1(y^2 - 1) + x + y + 1$.

Отметим, что ни один член остатка не делится на старший член никакого делителя.

Результат работы этого алгоритма зависит как от используемого мономиального упорядочения, так и от порядка f_i. В отдельных случаях этот алгоритм не всегда даже возвращает нулевой остаток от деления при $f = \sum_i a_i(x_1, \ldots, x_n)f_i(x_1, \ldots, x_n)$. Например, если мы разделим $xy^2 - x$ на $xy+1, y^2-1$ с lex-упорядочением, то получим $xy^2 - x = y(xy + 1) + 0(y^2 - 1) + (-x - y)$. Но если поменять порядок делителей, то окажется, что $xy^2-x = x(y^2-1)+0(xy+1)+0$. Таким образом, данный алгоритм крайне несовершенен и не может дать ответ на вопрос о принадлежности произвольного полинома идеалу с заданным базисом. Чтобы исправить ситуацию, можно заметить, что идеал может быть задан различными базисами. Возникает вопрос: существует ли такой базис идеала $< f_1, \ldots, f_s >$, остаток от деления полинома f на который был бы равен нулю тогда и только тогда, когда $f \in < f_1, \ldots, f_s >$? Вначале необходимо выяснить, для всякого ли идеала можно построить конечный базис.

4.2.2 Мономиальные идеалы

Решение выошеописанной задачи начнем с рассмотрения важного частного случая.

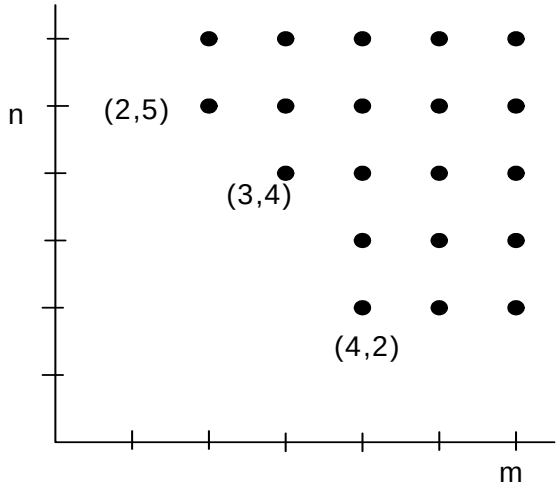


Рис. 4.4: Одночлены $x^m y^n$, входящие в мономиальный идеал $< x^4y^2, x^3y^4, x^2y^5 >$

Определение 4.10. Идеал $I \subset \mathbb{F}[x_1, \ldots, x_n]$ называется *мономиальным*, если $\exists A \subset \mathbb{Z}^n_{0+}$, такое что $I = \{\sum_{\alpha \in A} h_\alpha x^\alpha | h_\alpha \in \mathbb{F}[x_1, \ldots, x_n]\}$.

Примером мономиального идеала является $< x^4y^2, x^3y^4, x^2y^5 >$.

Лемма 4.6. *Пусть $I =< x^\alpha, \alpha \in A >$. Тогда моном $x^\beta \in I$ тогда и только тогда, когда $\exists \alpha \in A : x^\alpha | x^\beta$.*

Доказательство. Если $\exists \alpha \in A : x^\alpha | x^\beta$, то $x^\beta = x^\gamma x^\alpha$, откуда вытекает $x^\beta \in I$. Обратно, если $x^\beta \in I \Rightarrow x^\beta = \sum_i h_i x^{\alpha^{(i)}}, \alpha^{(i)} \in A$. Рассматривая h_i как линейные комбинации мономов, получим, что каждый член в правой части этого выражения делится на какой-то $x^{\alpha^{(i)}}$. x^β должен встретиться среди этих членов, откуда вытекает утверждение леммы. □

Эта лемма позволяет графически отобразить множество мономов, входящих в мономиальный идеал. На рисунке 4.4 представлено множество мономов, входящих в идеал $< x^4y^2, x^3y^4, x^2y^5 >$.

Лемма 4.7. *Пусть I — некоторый мономиальный идеал, а $f \in \mathbb{F}[x_1, \ldots, x_n]$. Тогда следующие условия эквивалентны:*

1. *$f \in I$;*
2. *Каждый член f принадлежит I.*
3. *f является $\mathbb{F}$-линейной комбинацией мономов из I.*

Доказательство. Цепочка импликаций $3 \Rightarrow 2 \Rightarrow 1$ очевидна. Импликация $1 \Rightarrow 3$ доказывается аналогично Лемме 4.6. □

Из этой леммы вытекает следующее:

Следствие 4.1. *Два мономиальных идеала совпадают тогда и только тогда, когда совпадают множества мономов, содержащихся в них.*

Основным свойством мономиальных идеалов является существование для них конечного базиса.

Теорема 4.4 (лемма Диксона)**.** *Любой мономиальный идеал* $I =< x^\alpha | \alpha \in A >\subset \mathbb{F}[x_1, \ldots, x_n]$ *может быть представлен в виде* $I =< x^{\alpha^{(1)}}, \ldots, x^{\alpha^{(s)}} >$, *где* $\alpha^{(1)}, \ldots, \alpha^{(s)} \in A$. *В частности, идеал имеет конечный базис.*

Доказательство. Доказательство по индукции. При $n = 1$ $I =< x_1^\alpha >, \alpha \in A$. Пусть β — наименьший элемент из A. Тогда все $x_1^\alpha : \alpha \in A$ делятся на x_1^β. Следовательно, $I =< x_1^\beta >$.

Пусть теорема справедлива для $n - 1$ переменной. Рассмотрим кольцо $\mathbb{F}[x_1, \ldots, x_{n-1}, y]$, так что мономы будут записываться как $x^\alpha y^m, \alpha \in \mathbb{Z}_{0+}^{n-1}, m \in \mathbb{Z}_{0+}$. Пусть $I \subset \mathbb{F}[x_1, \ldots, x_{n-1}, y]$ — мономиальный идеал. Рассмотрим идеал $J \subset \mathbb{F}[x_1, \ldots, x_{n-1}]$, порожденный такими мономами x^α, что $\exists m : x^\alpha y^m \in I$. Т.к. J — мономиальный идеал, по индукционному предположению он конечно порожден, $J =< x^{\alpha^{(1)}}, \ldots, x^{\alpha^{(s)}} >$. Тогда $\forall i : 1 \le i \le s \exists m_i : x^{\alpha^{(i)}} y^{m_i} \in I$. Пусть $m = \max m_i$. Рассмотрим идеалы $J_l \subset \mathbb{F}[x_1, \ldots, x_{n-1}]$, порожденные такими мономами x^β, что $x^\beta y^l \in I, 0 \le l \le m - 1$. По индукционному предположению, эти идеалы конечно порождены: $J_l =< x^{\alpha^{(l,1)}}, \ldots, x^{\alpha^{(l,s_l)}} >$. Докажем, что I порожден мономами из следующего списка:

$$\begin{aligned}
\text{из } J &: x^{\alpha^{(1)}} y^m, \ldots, x^{\alpha^{(s)}} y^m, \\
\text{из } J_0 &: x^{\alpha^{(0,1)}} y^0, \ldots, x^{\alpha^{(0,s)}} y^0, \\
\text{из } J_1 &: x^{\alpha^{(1,1)}} y^1, \ldots, x^{\alpha^{(1,s)}} y^1, \\
&\vdots \\
\text{из } J_{m-1} &: x^{\alpha^{(m-1,1)}} y^{m-1}, \ldots, x^{\alpha^{(m-1,s)}} y^{m-1}
\end{aligned}$$

Докажем вначале, что каждый моном в I делится хотя бы на один моном из этого списка. Пусть $x^\alpha y^p \in I$. Если $p \ge m$, то по определению J моном $x^\alpha y^p$ делится на некоторый моном $x^{\alpha^{(i)}} y^m$. Если $p \le m - 1$, то по определению идеала J_p моном $x^\alpha y_p$ делится на некоторый моном $x^{\alpha^{(p,j)}} y^p$. Это означает (Лемма 4.6, что мономы из приведенного конечного списка порождают идеал, содержащий те же мономы, что и I, откуда следует, что они совпадают.

Осталось доказать, что конечное множество образующих идеалов можно выбрать из заданного множества его образующих. Пусть $I \subset \mathbb{F}[x_1, \ldots, x_n], I =< x^\alpha, \alpha \in A >$. Как было показано выше, $I =< x^{\beta^{(1)}}, \ldots, x^{\beta^{(s)}} >$. По лемме 4.6, все $x^{\beta^{(i)}}$ делятся на некоторые $x^{\alpha^{(i)}}, \alpha^{(i)} \in A$. Ясно, что в качестве базиса I можно взять $< x^{\alpha^{(i)}}, i = 1..s >$. □

Лемма Диксона решает задачу описания мономиальных идеалов, устанавливая факт существования для каждого из них конечного базиса. Тогда факт принадлежности произвольного полинома f мономиальному идеалу $I =< x^{\alpha^{(i)}}, i = 1..s >$ *эквивалентен* факту равенства нулю остатка от деления f на $x^{\alpha^{(i)}}$. Необходимо еще раз отметить что в случае полиномиальных идеалов это в общем случае не так.

4.2.3 Базисы Гребнера

В данном разделе будет приведено полное решение задачи описания идеала.

Заметим, что как только задано мономиальное упорядочение, для каждого многочлена может быть однозначно определен его старший член.

Определение 4.11. Пусть $I \subset \mathbb{F}[x_1, \ldots, x_n]$ — ненулевой идеал.Пусть $LT(I)$ — множество старших членов элементов из I:

$$\mathrm{LT}(I) = \{cx^\alpha | \exists f \in I : \mathrm{LT}(f) = cx^\alpha\}.$$

Обозначим через $< \mathrm{LT}(I) >$ идеал, порожденный элементами из $LT(I)$.

Необходимо отметить, что если задан некоторый конечно порожденный идеал $I =< f_1, \ldots, f_s >$, то $< \mathrm{LT}(f_1), \ldots, \mathrm{LT}(f_s) >$ и $< \mathrm{LT}(I) >$ могут быть разными идеалами. Т.к. $f_i \in I, LT(f_i) \in \mathrm{LT}(I) \subset< \mathrm{LT}(I) >$. Поэтому $< \mathrm{LT}(f_1), \ldots, LT(f_s) > \subset< \mathrm{LT}(I) >$, причем последний может быть строго больше.

Пример 4.6. Пусть $I =< f_1, f_2 >, f_1 = x^3 - 2xy, f_2 = x^2y - 2y^2 + x$ и задано grlex упорядочение. Тогда $x(x^2y - 2y^2 + x) - y(x^2 - 2xy) = x^2$, т.е. $x^2 \in I$ и $x^2 = \mathrm{LT}(x^2) \in< \mathrm{LT}(I) >$, но $x^2 \notin< \mathrm{LT}(f_1), \mathrm{LT}(f_2) >$.

Теорема 4.5. *Пусть $I \subset \mathbb{F}[x_1, \ldots, x_n]$ — идеал. Тогда $< LT(I) >$ — мономиальный идеал и $\exists g_1, \ldots, g_s \in I :< \mathrm{LT}(I) >=< \mathrm{LT}(g_1), \ldots, \mathrm{LT}(g_s) >$.*

Доказательство. Старшие мономы $\mathrm{LM}(g)$ ненулевых элементов идеала порождают некоторый мономиальный идеал $< \mathrm{LM}(g), g \in I \setminus \{0\} >$. Так как $\mathrm{LM}(g) = c\,\mathrm{LT}(g), c \in \mathbb{F}\setminus\{0\}$, этот идеал совпадает с $< LT(g) : g \in I \setminus \{0\} =< \mathrm{LT}\, I >$. Следовательно, $< \mathrm{LT}(I) >$ — мономиальный идеал. По лемме Диксона $< \mathrm{LT}(I) >=< \mathrm{LM}(g_1), \ldots, \mathrm{LM}(g_s) >$ для конечного набора $g_i \in I$. Образующие этого идеала совпадают с точностью до ненулевых констант с $\mathrm{LT}(g_1), \ldots, \mathrm{LT}(g_s)$, откуда и следует утверждение теоремы. □

Это утверждение позволяет доказать конечную порожденность любого полиномиального идеала из кольца многочленов от *конечного* числа переменных.

Теорема 4.6 (Гильберта о базисе)**.** *Каждый идеал $I \subset \mathbb{F}[x_1, \ldots, x_n]$ является конечно порожденным, т.е. $I =< g_1, \ldots, g_s >$.*

Доказательство. Если $I = \{0\}$, то утверждение очевидно. В противном случае, как было доказано выше, $\exists g_1, \ldots, g_s \in I :< \mathrm{LT}(I) >=< \mathrm{LT}(g_1), \ldots, \mathrm{LT}(g_s) >$. Докажем, что $I =< g_1, \ldots, g_s >$.

Т.к. $g_i \in I, < g_1, \ldots, g_s > \subset I$. Пусть $f \in I$. Воспользуемся алгоритмом деления. Это приведет к представлению f в виде

$$f = a_1 g_1 + \ldots + a_s g_s + r,$$

где ни один член r нельзя поделить ни на один старший член $\mathrm{LT}(g_1), \ldots, \mathrm{LT}(g_s)$. Т.к. $r = f - a_1 g_1 - \ldots - a_s g_s \in I$, если $r \neq 0$, то $\mathrm{LT}(r) \in < \mathrm{LT}(I) >=< \mathrm{LT}(g_1), \ldots, \mathrm{LT}(g_s)$. Тогда $\mathrm{LT}(r)$ должен делиться хотя бы на один $\mathrm{LT}(g_i)$, что противоречит определению остатка. Следовательно, $r = 0$ и $f \in < g_1, \ldots, g_s >$, откуда $I \subset < g_1, \ldots, g_s >$. □

Естественно, что конечный базис, построенный при доказательстве этой теоремы, не является единственным. Однако он обладает рядом замечательных свойств.

Определение 4.12. Пусть задано некоторое мономиальное упорядочение. Конечное подмножество $G = \{g_1, \ldots, g_s\}$ элементов идеала I называется его *базисом Грёбнера*, если

$$< \mathrm{LT}(g_1), \ldots, \mathrm{LT}(g_s) >=< \mathrm{LT}(I) >$$

Другими словами, множество многочленов из идеала является его базисом Грёбнера тогда и только тогда, когда старший член любого элемента идеала делится хотя бы на один старший член $\mathrm{LT}\, g_i$. Из доказательства теоремы Гильберта о базисе вытекает следующий результат:

Следствие 4.2. *Пусть задано некоторое мономиальное упорядочение. Тогда любой ненулевой идеал $I \subset \mathbb{F}[x_1, \ldots, x_n]$ обладает базисом Грёбнера. Более того, базис Гребнера идеала I является его базисом.*

Доказательство. Множество $\{g_1, \ldots, g_s\}$, построенное в доказательстве теоремы 4.6, является базисом Грёбнера по определению. Там же было доказано, что оно является базисом идеала. □

Пример 4.7. Рассмотрим идеал $< x^3 - 2xy, x^2y - 2y^2 + x >$. Этот базис не является базисом Грёбнера относительно $grlex$ упорядочения, т.к. $x^2 \in < \mathrm{LT}(I) >$, но $x^2 \notin < x^3, x^2y >$.

Теорема 4.7. *Пусть $G = \{g_1, \ldots, g_s\}$ — базис Грёбнера идеала $I \subset \mathbb{F}[x_1, \ldots, x_n]$ и пусть $f \in \mathbb{F}[x_1, \ldots, x_n]$. Тогда существует единственный полином $r \in \mathbb{F}[x_1, \ldots, x_n]$, такой что*

1. *Ни один член r не делится ни на один из старших членов $\mathrm{LT}\, g_1, \ldots, \mathrm{LT}\, g_s$.*
2. *Существует $g \in I : f = g + r$.*

Другими словами, r является остатком от деления f на G, не зависящим от порядка делителей в G.

Доказательство. Алгоритм деления (см. рис. 4.3) позволяет записать $f = a_1g_1 + \ldots + a_sg_s + r$, где r удовлетворяет первому условию. Ясно также, что $g = a_1g_1 + \ldots + a_sg_s \in I$. Осталось доказать единственность r. Пусть $f = g + r = g' + r', g, g' \in I$. Тогда $r - r' = g' - g \in I$. Если $r \neq r'$, то $\mathrm{LT}(r - r') \in < \mathrm{LT}(I) > = < \mathrm{LT}(g_1), \ldots, \mathrm{LT}(g_s) >$, т.е. $\mathrm{LT}(r - r')$ делится на какой-то из старших членов $\mathrm{LT}(g_i)$, т.е. в r или r' существуют члены, делящиеся на $\mathrm{LT}(g_i)$, что противоречит условию 1. Следовательно, $r = r'$. □

Остаток r называется нормальной формой полинома f. Несмотря на единственность остатка, "частные"$a_1, \ldots, a_s$ зависят от порядка делителей. Кроме того, очевидно, что набор полиномов, являющийся базисом Грёбнера для одного мономиального упорядочения, может не быть таковым для другого.

Следствие 4.3. *Пусть $G = \{g_1, \ldots, g_s\}$ — базис Грёбнера идеала $I \subset \mathbb{F}[x_1, \ldots, x_n]$ и пусть $f \in \mathbb{F}[x_1, \ldots, x_n]$. $f \in I$ тогда и только тогда, когда остаток от деления f на G равен нулю*

Доказательство. Если остаток равен нулю, то принадлежность f идеалу очевидна. Если $f \in I$, то равенство $f = f + 0$ удовлетворяет всем условиям теоремы 4.7. Из единственности остатка от деления следует его равенство нулю. □

Заметим, что этот критерий позволяет решить задачу о принадлежности полинома идеалу, используя базис Грёбнера, построенный для любого мономиального упорядочения.

Для фиксированного мономиального упорядочения остаток от деления полинома f на упорядоченный набор $F = (f_1, \ldots, f_s)$ будем обозначать как $\overline{f}^F$. Как было показано выше, если F является базисом Грёбнера, порядок записи элементов в нем не важен.

Как следует из определения, единственным препятствием к тому, чтобы произвольный набор $f_1, \ldots, f_s$ был базисом Грёбнера идеала $< f_1, \ldots, f_s >$, является существование такой полиномиальной комбинации $f_1, \ldots, f_s$, такой что ее старший член не принадлежит идеалу $< \mathrm{LT}(f_1), \ldots, \mathrm{LT}(f_s) >$, т.е. старшие члены сокращаются. Фолрмализуем процесс сокращения старших членов.

Определение 4.13. Пусть $f, g \in \mathbb{F}[x_1, \ldots, x_n]$ — ненулевые полиномы.

1. Пусть[2] $\mathrm{multideg}(f) = \alpha$, $\mathrm{multideg}(g) = \beta$. Пусть $\gamma = (\gamma_1, \ldots, \gamma_n) : \gamma_i = \max(\alpha_i, \beta_i)$. Тогда x^γ называется наименьшим общим кратным мономов $\mathrm{LM}(f), \mathrm{LM}(g)$ и обозначается $x^\gamma = \mathrm{LCM}(\mathrm{LM}(f), \mathrm{LM}(g))$

2. *S-полиномом от f и g* называется

$$S(f, g) = \frac{x^\gamma}{\mathrm{LT}(f)} f - \frac{x^\gamma}{\mathrm{LT}(g)} g$$

[2]Мультистепенью полинома называется вектор показателей степеней его старшего члена.

BUCHBERGER($F = (f_1, \ldots, f_s)$)
1 $G \leftarrow F$;
2 $G' \leftarrow \emptyset$;
3 **while** $G \neq G'$
4 **do** $G' \leftarrow G$;
5 **for** $\{p, q\}, p \neq q, p, q \in G'$
6 **do** $S \leftarrow \overline{S(p,q)}^{G'}$
7 **if** $S \neq 0$
8 **then** $G \leftarrow G \cup \{S\}$
9 **return** G

Рис. 4.5: Алгоритм Бухбергера

Например, для grlex упорядочения $S(x^3y^2 - x^2y^3 + x, 3x^4y + y^2) = \frac{x^4y^2}{x^3y^2}(x^3y^2 - x^2y^3 + x) - \frac{x^4y^2}{3x^4y}(3x^4y + y^2) = x(x^3y^2 - x^2y^3 + x) - (y/3)(3x^4y + y^2) = -x^3y^3 + x^2 - y^3/3$.

Теорема 4.8 (Критерий Бухбергера S-пар)**.** *Базис $G = (g_1, \ldots, g_s)$ идеала I является его базисом Грёбнера тогда и только тогда, когда $\forall i \neq j : \overline{S(g_i, g_j)}^{G} = 0$.*

Доказательство этого утверждения сравнительно несложно и может быть найдено в [10].

Критерий Бухбергера позволяет построить алгоритм построения базиса Грёбнера, приведенный на рис. 4.5 (*алгоритм Бухбергера*). Необходимо отметить, что в данной форме алгоритм Бухбергера крайне неэффективен. Ясно, что наиболее сложной его частью является вычисление остатков от деления. Известны различные приемы как снижения числа обращений к процедуре деления, так и одновременного вычисления нескольких остатков. Более того, сложность вычислений существенно зависит от выбранного упорядочения.

Можно также отметить определенное сходство между алгоритмом Бухбергера и алгоритмами Евклида и Гаусса.

Базисы Грёбнера, порождаемые этим алгоритмом, как правило, содержат очень большое число элементов.

Определение 4.14. *Минимальным базисом Грёбнера* полиномиального идеала I называется такой его базис Грёбнера G, что $\forall p \in G : \mathrm{LC}(p) = 1$ и $\forall p \in G : \mathrm{LT}(p) \notin < \mathrm{LT}(G - \{p\}) >$.

Определение 4.15. *Редуцированным базисом Грёбнера* полиномиального идеала I называется такой его базис Грёбнера G, что $\forall p \in G : \mathrm{LC}(p) = 1$ и никакой моном никакого $p \in G$ не принадлежит $< \mathrm{LT}(G - \{p\}) >$.

Можно показать, что редуцированный базис Грёбнера единственен. Это позволяет решить задачу равенства двух идеалов: они совпадают тогда и только тогда, когда их

редуцированные базисы Грёбнера, построенные для некоторого (любого) мономиального упорядочения, одинаковы.

4.2.4 Применение базисов Грёбнера

Задача о принадлежности идеалу

Для того, чтобы определить факт принадлежности f идеалу $I =< f_1, \ldots, f_s >$, достаточно построить его базис Грёбнера G относительно проивзольного мономиального упорядочения и найти остаток $\overline{f}^G$. Если он равен нулю, то $f \in I$.

Решение полиномиальных уравнений

Пример 4.8. Рассмотрим систему уравнений

$$\begin{cases} 3x^2 + 2yz - 2\lambda x &= 0 \\ xz - \lambda y &= 0 \\ xy - z - \lambda z &= 0 \\ x^2 + y^2 + z^2 - 1 &= 0 \end{cases}$$

Построим базис Грёбнера относительно лексикографического упорядочения с $\lambda > x > y > z$. Он содержит следующие элементы (нормализация старших коэффициентов не проводилась):

$$1152 * z^7 - 1763 * z^5 + 655 * z^3 - 44 * z$$
$$118 * y * z^3 - 118 * y * z - 1152 * z^6 + 1605 * z^4 - 453 * z^2$$
$$3835 * z * y^2 - 6912 * z^5 + 10751 * z^3 - 3839 * z$$
$$3835 * y^3 + 3835 * y * z^2 - 3835 * y - 9216 * z^5 + 11778 * z^3 - 2562 * z$$
$$3835 * x * z + 3835 * y * z^2 - 1152 * z^5 - 1404 * z^3 + 2556 * z$$
$$3835 * x * y - 19584 * z^5 + 25987 * z^3 - 6403 * z, x^2 + y^2 + z^2 - 1$$
$$7670 * \lambda - 11505 * x - 11505 * y * z - 335232 * z^6 + 477321 * z^4 - 134419 * z^2$$

Можно заметить, что первый элемент этого базиса содержит многочлены, зависящие только от z. Т.к. $V(I)$ не зависит от базиса идеала, все значения z, соответствующие корням системы уравнений (т.е. точкам аффинного многообразия), могут быть найдены как корни этого многочлена с использованием специализированных методов. Далее они могут быть подставлены в оставшиеся уравнения, что дает возможность полностью описать все решения исходной системы уравнений.

4.2.5 Задача нахождения неявного представления

Пусть заданы параметрические уравнения

$$\begin{aligned} x_1 &= f_1(t_1, \ldots, t_m) \\ &\vdots \\ x_n &= f_n(t_1, \ldots, t_m), \end{aligned}$$

задающие подмножество некоторого аффинного многообразия. Как найти полиномиальные уравнения от x_i, зависящие только от x_i? Если f_i являются полиномами, то можно построить базис Грёбнера для лексикографического упорядочения с $t_1 > t_2 > \ldots > t_m > x_1 > \ldots > x_n$. Он будет содержать полиномы, с исключенными переменными, причем порядок их исключения определяется порядком переменных в лексикографическом упорядочении.

Упражнения

1. Доказать, что объединение и пересечение аффинных многообразий также являются аффинными многообразиями.
2. Доказать, что множества $\{(x, x) | x \in \mathbb{R}, x \neq 1\}$ и $\{(x, y) | x, y \in \mathbb{R}, y \geq 0\}$ не являются аффинными многообразиями.
3. Доказать, что $\forall f, g \in= F[x] :< f, g >=< GCD(f, g) >$.
4. Найти базис идеала $I(V(x^5 - 2x^4 + 2x^2 - x, x^5 - x^4 - 2x^3 + 2x^2 + x - 1))$.
5. Доказать, что градуированное лексикографическое упорядочение является мономиальным упорядочением.
6. Доказать, что градуированное обратное лексикографическое упорядочение является мономиальным упорядочением.

Вопросы к экзамену

1. Понятие множества. Способы задания.
2. Операции над множествами
3. Представление множеств в ЭВМ. Коды Грея.
4. Декартово произведение множеств.
5. Соответствия и отношения
6. Операции над бинарными отношениями
7. Классификация бинарных отношений
8. Отношения эквивалентности
9. Отношения порядка
10. Отображения на упорядоченных множествах
11. Замыкания отношений
12. Понятие алгебры
13. Морфизмы
14. Группоиды, моноиды, полугруппы, группы.
15. Конечные и циклические группы.
16. Подалгебры. Теорема Лагранжа
17. Кольца, полукольца, тела, поля
18. Умножение произвольных матриц
19. Умножение двоичных матриц
20. Вычисление значений многочленов

21. Билинейные формы. Линейная и циклическая свертки.
22. Алгоритмы Карацубы и Тоома-Кука вычисления свертки
23. Алгоритм Винограда
24. Перенос алгоритмов свертки на поля другой природы.
25. Гнездовые алгоритмы свертки
26. Алгоритм Агарвала-Кули
27. Итерированные алгоритмы свертки
28. Преобразование Фурье в дискретном и непрерывном случаях
29. Алгоритм Кули-Тьюки
30. Алгоритм Гуда-Томаса
31. Алгоритм Герцеля
32. Метод Блюстейна
33. Метод Рейдера
34. Линеаризованные и аффинные многочлены над конечными полями
35. Алгоритм Ванга-Жу
36. Циклотомический алгоритм
37. Представление целых чисел в ЭВМ
38. Сложение, умножение и возведение в степень целых чисел
39. Деление целых чисел
40. Идеалы и аффинные многообразия
41. Полиномы от одной переменной
42. Упорядочение мономов в $\mathbb{F}[x_1, \ldots, x_n]$ и алгоритм деления.
43. Мономиальные идеалы
44. Базисы Грёбнера
45. Применение базисов Грёбнера

Предметный указатель

Литература

[1] *Афанасьев В. Б., Грушко И. И.* Алгоритмы БПФ для полей $GF(2^m)$ // Помехоустойчивое кодирование и надежность ЭВМ. — М.: Наука, 1987. — С. 33–55.

[2] *Ахо А., Хопкрофт Д., Ульман Д.* Построение и анализ вычислительных алгоритмов. — М.: Мир, 1979. — 536 с.

[3] *Белоусов А. И., Ткачев С. Б.* Дискретная математика. — М.: Издательство МГТУ им. Баумана, 2001. — 744 с.

[4] *Берлекэмп Э. Р.* Алгебраическая теория кодирования. — М.: Мир, 1971. — 477 с.

[5] *Блейхут Р.* Теория и практика кодов, контролирующих ошибки. — М.: Мир, 1986. — 576 с.

[6] *Блейхут Р.* Быстрые алгоритмы цифровой обработки сигналов. — М.: Мир, 1989. — 448 с.

[7] *Габидулин Э. М., Афанасьев В. Б.* Кодирование в радиоэлектронике. — М.: Радио и связь, 1986. — 176 с.

[8] *Захарова Т. Г.* Вычисление преобразования Фурье в полях характеристики 2 // *Проблемы передачи информации*. — 1992. — Т. 28, № 2. — С. 62–76.

[9] *Кнут Д.* Искусство программирования. — М.: Вильямс, 2000. — Т. 2.

[10] *Кокс Д., Литтл Д., О'Ши Д.* Идеалы, многообразия и алгоритмы. — М.: Мир, 2000. — 687 с.

[11] *Мак-Вильямс Ф. Д., Слоэн Н. Д. А.* Теория кодов, исправляющих ошибки. — М.: Связь, 1979. — 744 с.

[12] *Новиков Д. Ф.* Дискретная математика для программистов. — СПб.: Питер, 2000.

[13] *Рейнгольд Э., Нивергельт Ю., Део Н.* Комбинаторные алгоритмы: теория и практика. — М.: Мир, 1980. — 478 с.

[14] *Трифонов П. В., Федоренко С. В.* Метод быстрого вычисления преобразования Фурье над конечным полем // *Проблемы передачи информации*. — 2003. — Т. 39, № 3. — С. 3–10.

[15] *Wang Y., Zhu X.* A fast algorithm for the Fourier transform over finite fields and its VLSI implementation // *IEEE Journal on Selected Areas in Communications*. — 1988. — Vol. 6, no. 3. — Pp. 572–577.