МОСКОВСКИЙ ОРДЕНА ЛЕНИНА, ОРДЕНА ОКТЯБРЬСКОЙ РЕВОЛЮЦИИ И ОРДЕНА ТРУДОВОГО КРАСНОГО ЗНАМЕНИ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ им. М.В. ЛОМОНОСОВА

Механико-математический факультет

Е.В.Панкратьев

КОМПЬЮТЕРНАЯ АЛГЕБРА

Факторизация многочленов

Издательство
Московского университета
1988

УДК 512.622

Панкратьев Е.В. Компьютерная алгебра. Факторизация многочленов. – М.: Изд-во Моск. ун-та, 1988. – 85 С. ISBN 5-221-00591-0.

Книга является первой в серии учебных пособий по курсу "Компьютерная алгебра". Рассматривается одна из актуальных задач компьютерной алгебры – разложение многочленов на неприводимые множители. В последние 20 лет получены значительные результаты, позволяющие эффективно использовать для решения этой задачи вычислительную технику. В пособии нашли отражение современные алгоритмы факторизации и работы, проводимые на механико-математическом факультете по их реализации.

Для студентов механико-математического факультета.

Р е ц е н з е н т ы :

докт. физ.-мат. наук В.Н. Латышев
канд. физ. мат. наук А.В. Михалев

Печатается по постановлению
Редакционно-издательского совета
Московского университета

077(02)-88-заказное
ISBN 5-211-00591-0

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Что такое компьютерная алгебра?

Термин "компьютерная алгебра" появился в конце 70-х годов и длительное время употреблялся в качестве синонима терминов "аналитические и символьные вычисления на ЭВМ". Даже в настоящее время этот термин на французском языке дословно означает "формальные вычисления".

В чем основные отличия символьных вычислений от численных и почему возник термин "компьютерная алгебра"?

Когда говорим о вычислительных методах, то считаем, что все вычисления выполняются в поле вещественных или комплексных чисел. В действительности же всякая программа для ЭВМ имеет дело только с конечным набором рациональных чисел, поскольку только такие числа представляются в компьютере. Для записи целого числа отводится обычно 16 или 32 двоичных символа (бита), для вещественного - 32 или 64 бита. Это множество не замкнуто относительно арифметических операций, что может выражаться в различных переполнениях, например, при умножении достаточно больших чисел или при делении на маленькое число. Еще более существенной особенностью вычислительной математики является то, что арифметические операции над этими числами, выполняемые компьютером, отличаются от арифметических операций в поле рациональных чисел, более того, для компьютерных операций не выполняются основные аксиомы поля (ассоциативности, дистрибутивности). Эти особенности компьютерных вычислений выражаются в терминах погрешности или точности вычислений. Оценка погрешности представляет одну из основных проблем вычислительной математики. Каждую задачу требуется решить с использованием имеющихся ресурсов ЭВМ, за обозримое время, с заданной точностью.

Набор объектов, применяемых в символьных вычислениях, весьма разнообразен, в частности, в них используется значительно большее множество рациональных чисел. Это множество все равно остается конечным, но ограничения на допустимые размеры числа (количество знаков в его записи) связаны обычно с размерами оперативной памяти ЭВМ, что позволяет пользоваться практически любыми рациональными числами, операции над которы-

ми выполняются за приемлимое время. При этом компьютерные операции над рациональными числами совпадают с соответствующими операциями в поле рациональных чисел. Таким образом, снимается одна из основных проблем вычислительных методов – оценка погрешности вычислений.

В компьютерной алгебре практически не применяются вещественные и комплексные числа, зато широко используется алгебраические числа. Алгебраическое число задается своим минимальным многочленом, а иногда для его задания требуется указать интервал на прямой или область в комплексной плоскости, где содержится единственный корень данного многочлена. Многочлены играют в символьных вычислениях исключительно важную роль. На использовании полиномиальной арифметики основаны теоретические методы аналитической механики, они используются во многих областях математики, физики и других наук. Кроме того, в компьютерной алгебре рассматриваются такие объекты, как дифференциальные поля (функциональные поля), допускающие показательные, логарифмические, тригонометрические функции, матричные кольца (элементы матрицы принадлежат кольцам достаточно общего вида) и другие. Даже при арифметических операциях над такими объектами происходит разбухание информации, для записи промежуточных результатов вычислений требуется значительный объем памяти ЭВМ.

Ограничения на алгоритмы решаемых компьютерной алгеброй задач накладываются имеющимися ресурсами ЭВМ и обозримостью времени счета. Однако ограничения по времени счета и по используемой памяти в символьных вычислениях существенно более обременительны, чем в вычислительных методах.

Предмет компьютерной алгебры определяют иногда как область математики, которая является слишком вычислительной, чтобы быть изложенной в курсах общей алгебры, и слишком алгебраической, чтобы попасть в литературу по вычислительной математике.

Большинство задач, рассматриваемых компьютерной алгеброй, представляет собой классические математические задачи, для многих из них известны алгоритмы решения "за конечное число шагов". Однако на практике эти алгоритмы часто не применяются из-за их высокой сложности (большое

количество операций), а используются различные эвристические методы.

Наряду с задачами алгебраического упрощения и интегрирования в конечном виде, задача разложения многочленов на неприводимые множители относится к бурно развивающимся направлениям компьютерной алгебры. Рассматриваем эту задачу в следующей постановке: дан многочлен $f(x) \in Z[x]$ с целыми коэффициентами от одной переменной, требуется разложить его на неприводимые множители. С точки зрения "чистого" математика эта задача давно решена полностью и окончательно: получен алгоритм, позволяющий находить требуемое разложение "за конечное число шагов". Один из таких алгоритмов получен в 1882 году Кронекером, чьим именем он и называется в настоящее время, хотя за 100 лет до Кронекера этот алгоритм был известен австрийскому астроному Шуверту. Следующий шаг в исследовании алгоритмов факторизации был сделан только в 60-х годах текущего столетия, когда был найден достаточно эффективный алгоритм для разложения на множители многочленов с коэффициентами из конечного поля. Использование этого алгоритма в сочетании с леммой Гензеля позволил получить алгоритмы факторизации многочленов с целыми коэффициентами, пригодные для практической реализации. С конца 60-х годов появляется большое количество работ по факторизации. Предлагаются усовершенствования алгоритмов, направленные на увеличение их быстродействия, на расширение области их применения, в частности, рассматривается задача факторизации многочленов от одной и многих переменных с коэффициентами из конечных полей, из полей алгебраических чисел и т.д. Крупным вкладом в теорию факторизации многочленов явилась работа [9], которая позволила получить алгоритм факторизации, сложность которого оценивается полиномом от его степени. Из советских математиков наиболее существенный вклад в эту теорию внесли ленинградские математики Д.Ю.Григорьев [4] и А.Л.Чистов [7].

Следует отметить, что литература по компьютерной алгебре, особенно на русском языке, представлена более, чем скромно. В 1986 году вышла книга [6], которая является, пожалуй, единственной монографией на русском языке, в которой достаточно широко и достаточно подробно освещаются различные вопросы компьютерной алгеб-

ры. Не успев появиться в продаже, эта книга стала библиографической редкостью. Оригинал этой книги на английском языке вышел в свет в конце 1982 года, а уже в 1983 вышло второе ее издание. К настоящему времени за рубежом вышло еще несколько книг по компьютерной алгебре, но они, к сожалению, пока не переведены на русский язык. Некоторые вопросы компьютерной алгебры затрагиваются в монографиях [1] и [5]. В основном же компьютерную алгебру приходится изучать по многочисленным статьям, ориентироваться в которых весьма нелегко.

Еще более скупо представлена литература по отдельным проблемам компьютерной алгебры. Перевод монографии Дэвенпорта об интегрировании алгебраических функций представляет собой одну из очень немногочисленных книг, полностью посвященных каким-либо проблемам компьютерной алгебры. Развитие такого направления, как факторизация многочленов, находит свое отражение почти исключительно в статьях и препринтах. В более солидной литературе можно найти только результаты, которые можно причислить к классическим, причем ни в одном месте они не собраны воедино: алгоритм Кронекера можно прочитать в монографии Ван дер Вардена [3], алгоритм Берлекампа – у Кнута [5], в сборнике [6] изложен алгоритм факторизации, опирающийся на разложении многочлена над полем p-адических чисел и переборе всех возможных произведений. Алгоритмы факторизации с полиномиальной сложностью содержатся в [7 – 9].

Настоящее пособие посвящено задаче факторизации многочленов. Основное внимание уделяется факторизации многочленов от одной переменной с целыми коэффициентами. Излагаются современные алгоритмы факторизации, описываются работы, выполненные на механико-математическом факультете в последние годы по реализации этих алгоритмов. В основу изложения положены конспекты спецкурса, прочитанного автором на механико-математическом факультете в 1983/84 и в 1986/87 учебных годах.

Автор пользуется случаем выразить свою благодарность А.В.Красинцу, конспекты которого были использованы при подготовке данного учебного пособия, А.В.Астрелину и И.И.Тютееву, являющимся одними из первых читателей данной рукописи, и, особенно, профессору В.Н.Латышеву и доценту А.В.Михалеву, взявшим на себя труд прочитать рукопись и сделавшим ряд полезных замечаний.

1. АЛГОРИТМЫ КРОНЕКЕРА

Прежде, чем переходить к изложению алгоритмов, сделаем несколько замечаний, касающихся используемых обозначений и формы записи алгоритмов. Запись алгоритмов осуществляем в форме, по возможности близкой к тем, которые используются в курсе информатики для средней школы и на механико-математическом факультете МГУ в курсе программирования [2]. Алгоритм снабжаем именем, за которым в скобках следует список параметров с указанием их типа. К стандартным используемым типам относятся:

Z – целое число;
Z+ – неотрицательное целое число;
Q – рациональное число;
Zp – целое p-адическое число;
Qp – рациональное p-адическое число;
R – вещественное число;
C – комплексное число;
CZ – гауссово число (вида a+bi, a,b ∈ Z, $i^2=-1$);
CQ – рационально-комплексное число;
L – логическая переменная (типа "да/нет").

Описания этих типов могут сокращаться до трех первых букв.

Могут использоваться структуры данных: стек, дек, вектор, множество и т.д.

При необходимости вводим новые типы данных, в частности, широко пользуемся данными типа "многочлен" и "разложение", определяемыми следующим образом:

```
Z[x] – многочлен :
      запись(степень : Z+
             коэффициенты : вектор элементов типа
                            Z с индексом Ø..n);
разложение :
      запись(количество_множителей : Z+
             множители : вектор элементов типа
                         многочлен с индексом
                         1..количество_множите-
                         лей).
```

Иногда используем многочлены с коэффициентами из другой области коэффициентов (поле рациональных чисел Q, конечное поле Fp и т.д.), соответствующие типы обозначаем Q[x], Fp[x] и т.д.

В записи алгоритмов косая черта \ означает,

что в строке за ней следуют комментарии.

Алгоритм Кронекера основан на следующих соображениях: если степень многочлена f равна n, то степень хотя бы одного множителя f_1 многочлена f не превосходит [n/2]; значения как f, так и f_1 в целых точках – целые числа, причем $f_1(i)$ делит f(i) для любого целого i; при фиксированном i значение $f_1(i)$ может принимать только конечное множество значений, состоящее из делителей числа f(i); коэффициенты многочлена f_1 однозначно восстанавливаются по его значениям в [n/2]+1 точке. Таким образом, для f_1 получается конечное число возможностей, непосредственным делением проверяем, получили ли делитель многочлена f.

Перепишем алгоритм Кронекера в соответствии со сделанными выше замечаниями.

1.1. АЛГОРИТМ Кронекера (многоч f,g, лог успех)

Арг: f
Рез: g, успех \= "да", если множитель найден.
Обозначения:
n == f.степень
m == g.степень
f(i), i ∈ Z == значение многочлена f в точке i.
Переменные:
M – множество элементов типа целое
U – множество 'динамических' векторов элементов типа целое

```
начало
.  успех := нет
.  U := множество делителей числа f(Ø)
.  цикл для i от 1 до [n/2] пока не успех
                          \поиск множителя степени i
.  .  M := множество делителей числа f(i)
.  .  U := U x M       \прямое произведение
.  .  цикл для каждого u из U пока не успех
.  .  .  построить многочлен g степени i, такой,
                    что g(j) = u(j)  для j=Ø..i
.  .  .  если f делится на g, то
.  .  .  .  успех := да
```

```
. . . . m := i
. . . конец если
. . конец цикла
. конец цикла
конец программы
```

Сформулируем несколько нетрудных упражнений, результаты которых могут понадобиться нам в дальнейшем.

Упражнение 1.2. Показать, что задача разложения многочлена $f \in Z[x]$ на неприводимые множители в кольце $Z[x]$ равносильна задаче разложения этого многочлена на неприводимые множители в $Q[x]$. А именно, показать, что если $f=g \cdot h$, где $g,h \in Q[x]$ и НОД числителей коэффициентов каждого из многочленов g и h равен 1, то $g,h \in Z[x]$.

Упражнение 1.3. Написать алгоритм деления с остатком многочленов из кольца $Q[x]$.

Упражнение 1.4. Составить программу для вычисления логической функции, аргументами которой являются многочлены $f,g \in Z[x]$, и которая принимает значение "да", если f делится на g.

Упражнение 1.5. Написать алгоритм построения многочлена степени m по его значениям в $m+1$ точке.

Замечание 1.6. Достаточно научиться разлагать на множители многочлены со старшим коэффициентом, равным 1. Действительно, если старший коэффициент равен a, то домножив на a^{n-1} и сделав замену $x=y/a$, сводим задачу к этому случаю. После ее решения остается сделать обратную замену и сократить на общий множитель a^{n-1}. Однако этот метод обычно оказывается неэффективным, за счет увеличения коэффициентов ухудшаются различные оценки и скорость работы алгоритмов. Поэтому в большинстве работающих алгоритмов таких преобразований не производится.

Другое решение задачи факторизации "за конечное число шагов" следует из того, что коэффициенты делителя – целые числа и их абсолютная величина ограничена сверху некоторой функцией от коэффициентов многочлена f. Эти границы понадобятся нам в дальнейшем для других целей, поэтому, не откладывая в долгий ящик, вычислим их в следующем параграфе.

Задача разложения на неприводимые множители "за конечное число шагов" многочленов от нескольких переменых с "классической" точки зрения решена также примерно сто лет назад. Соответствующий алгоритм также носит имя Кронекера и для некоторых областей коэффициентов (например, для поля комплексных чисел C) остается единственным известным алгоритмом решения этой задачи. Для многочленов с коэффициентами из кольца целых чисел, или из кольца алгебраических чисел, или из конечного поля и некоторых других получены в последнее время новые, более быстрые алгоритмы. Общая схема этих алгоритмов достаточно близка к соответствующим алгоритмам факторизации одномерных многочленов, хотя некоторые отличия весьма существенны. Изложение современных алгоритмов факторизации многомерных многочленов не входит в число вопросов, освещаемых в данном пособии. Читателю, интересующемуся этой задачей, следует обратиться к специальной литературе.

Ниже излагаем многомерный алгоритм Кронекера для задачи, поставленной следующим образом.

Пусть D - область целостности с однозначным разложением на множители,
$f(x_1, \ldots, x_n) \in D[x_1, \ldots, x_n]$.
Требуется разложить f на неприводимые множители.

1.7. Многомерный алгоритм Кронекера

АЛГОРИТМ Кронекера_многомерный
(многочлен f, разложение G)
Арг: $f \in Z[x_1, \ldots, x_n]$
Рез: G
Переменные:

многочлен $\overline{f} \in Z[y]$, разложение $\overline{G}$ многочлена $\overline{f}$
множество M элементов типа целое

Идея реализации:

Редуцировать задачу к одномерному случаю, путем введения новой неизвестной и заменой всех переменных достаточно высокими степенями этой неизвестной. Факторизовать получившийся многочлен от новой неизвестной. Выполнить обратную подстановку, пробным делением убедиться, получили ли желаемое разложение.

начало

. выбрать целое d большее, чем степени отдельных переменных в f

. заменить все переменные степенями новой неизвестной y:

$$\bar{f}(y) := S_d(f) = f(y, y^d, \ldots, y^{d^{n-1}}).$$

. разложить $\bar{f}(y)$ на неприводимые множители, т.е. $\bar{f}(y) = \bar{g}_1(y) \ldots \bar{g}_s(y)$, $\bar{g}_i(y) \in D[y]$, $1 \le i \le s$.

G.количество_множителей := 1

. m := 1

. M := {1, ..., s}

. цикл пока m ≤ [s/2]

. . цикл для всех подмножеств $\{i_1, \ldots, i_m\}$ множества M пока m ≤ [s/2]

. . . $g_{i_1,\ldots,i_m}(x_1,\ldots,x_n) :=$

$$:= S_d^{-1}(\bar{g}_{i_1}(y)\bar{g}_{i_2}(y) \ldots \bar{g}_{i_m}(y))$$

\где обратное преобразование S_d^{-1} определяется на одночленах

$$S_d^{-1}(\mu y^{b_1+db_2+\ldots+d^{v-1}b_v}) = \mu x_1^{b_1} \ldots x_v^{b_v},$$

\($0 \le b_i < d$ для $1 \le i \le v$, $\mu \in Z$), далее S_d^{-1} \распространяется по линейности.

. . . если f делится на g

. . . . то

. . . . G.множитель[G.колич_множит] := g

. . . . G.количество_множителей :=
:= G.количество_множителей +1

. . . . f := f/g

. . . . s := s - m

. . . . M.удалить $i_1, i_2, \ldots, i_m$

. . . конец если

. . конец цикла

. . m := m + 1

. конец цикла

. G.множитель[G.количество_множителей] := f

конец

2. КОМБИНАТОРНЫЕ ОЦЕНКИ

2.1. ГРАНИЦЫ ДЛЯ КОЭФФИЦИЕНТОВ ДЕЛИТЕЛЯ МНОГОЧЛЕНА

2.1.1. НЕРАВЕНСТВО КОШИ

Пусть $d \geq 1$,

$$P(x) = a_0 x^d + a_1 x^{d-1} + \ldots + a_d, \quad a_0 \neq 0 \qquad (2.1.2)$$

– многочлен с комплексными коэффициентами. Тогда любой корень z многочлена $P(x)$ удовлетворяет неравенству

$$|z| < 1 + \frac{\max\{|a_1|, \ldots, |a_d|\}}{|a_0|}.$$

Доказательство. Пусть $P(z)=0$. Если $|z| \leq 1$, то утверждение теоремы тривиально. Предположим, что $|z| > 1$ и положим $H = \max\{|a_1|, \ldots, |a_d|\}$. По предположению

$$a_0 z^d = -a_1 z^{d-1} - \ldots - a_d, \text{ следовательно,}$$

$$|a_0|\,|z|^d \leq H\,(|z|^{d-1} + \ldots + 1) < \frac{H\,|z|^d}{|z| - 1},$$

то есть $|a| \cdot (|z| - 1) < H$. □

2.1.3. НЕРАВЕНСТВО ЛАНДАУ

Пусть $F = \sum_{k=0}^{m} c_k x^k$. Положим

$$\|F\| = \left(\sum_{k=0}^{m} |c_k|^2\right)^{1/2}. \qquad (2.1.4)$$

Рассматриваем формулу 2.1.4 как некоторое удобное обозначение. Можно доказать, что эта формула задает на пространстве многочленов метрику, но мы не пользуемся этим фактом, необходимые нам свойства этой метрики будут доказаны.

Теорема. Предположим, что многочлен $P(x)$ задан формулой 2.1.2. Пусть $z_1, \ldots, z_d$ – корни многочлена $P(x)$. Положим

$M(P) = |a_0| \prod_{j=1}^{d} \max\{1, |z_j|\}$. Тогда $M(P) \leq \|P\|$.

Для доказательства теоремы нам понадобится

Лемма. Если Q – многочлен и z – комплексное число, то

$$\|(x + z)\,Q(x)\| = \|(\bar{z}x + 1)\,Q(x)\|. \qquad (2.1.5)$$

Доказательство. Пусть $Q(x) = \sum_{k=0}^{m} c_k x^k$. Тогда квадрат выражения в левой части равенства 2.1.5 равен

$$\sum_{k=0}^{m+1} (c_{k-1} + zc_k)\,(\bar{c}_{k-1} + \bar{z}\bar{c}_k) =$$

$$= (1 + |z|^2)\,\|Q\|^2 + \sum_{k=0}^{m+1} (zc_k\bar{c}_{k-1} + \bar{z}\bar{c}_k c_{k-1})$$

(полагаем $c_{-1} = c_{m+1} = 0$). Этому же выражению равен и квадрат правой части. □

Доказательство теоремы. Пусть $z_1, \ldots, z_k$ – корни многочлена $P(x)$, лежащие вне единичного круга. Тогда $M(P) = |a_0| \cdot |z_1 \ldots z_k|$. Положим

$$R(x) = a_0 \prod_{j=1}^{k} (\bar{z}_j x - 1) \prod_{j=k+1}^{d} (x - z_j) = b_0 x^d + \ldots + b_d.$$

k-кратное применение леммы дает $\|P\| = \|R\|$. Однако $\|R\|^2 \geq \|b_0\|^2 = [M(P)]^2$. □

Теорема 2.1.6. Пусть

$$Q = b_0 x^q + b_1 x^{q-1} + \ldots + b_q, \quad b_0 \neq 0$$

– делитель многочлена $P(x)$, задаваемого формулой 2.1.2. Тогда

$$|b_0| + |b_1| + \ldots + |b_q| \leq |b_0/a_0| \cdot 2^q \|P\|.$$

Доказательство. Легко проверяется, что

$$|b_0| + |b_1| + \ldots + |b_q| \leq 2^q M(Q),$$

но $M(Q) \leq |b_0/a_0| \cdot M(P)$ и из неравенства Ландау следует, что $M(P) \leq \|P\|$. □

Задача 2.1.7. Пусть $f(x) \in Z[x]$ и $h(x)$ – делитель многочлена $f(x)$. Предположим, что $\deg(h) \leq m$. Тогда

$$\|h\| \leq \binom{2m}{m}^{1/2} \cdot \|f\|.$$

(Символ $\binom{n}{k}$ используется здесь и неодно-

кратно в дальнейшем для обозначения биномиальных коэффициентов, т.е. числа сочетаний из n по k)

Другие полезные границы можно найти, например, в работе [8].

2.2. РЕДУЦИРОВАННЫЕ БАЗИСЫ РЕШЕТКИ

Определение 2.2.1. Решеткой в n-мерном векторном пространстве над полем вещественных чисел R или над полем рациональных чисел Q называется свободный Z-модуль L ранга n, т.е. существует базис $b_1, \ldots, b_n$ пространства R^n (соответственно Q^n), такой, что

$$L = \sum_{i=1}^{n} Zb_i = \{ \sum_{i=1}^{n} r_i b_i \mid r_i \in Z, \ 1 \le i \le n \}$$

В этом случае n называется рангом решетки, а множество векторов $b_1, \ldots, b_n$ – ее базисом.

Определение 2.2.2. Детерминантом d(L) решетки L называется положительное число, определяемое формулой

$$d(L) = |\det (b_1, b_2, \ldots, b_n)|,$$

для некоторого базиса $b_1, \ldots, b_n$ решетки L.

Упражнение 2.2.3. Показать, что определение 2.2.2 является корректным, т.е. d(L) не зависит от выбора базиса решетки L.

Для формулировки определения редуцированного базиса решетки нам нужно напомнить процесс ортогонализации Грама–Шмидта. Векторы b_i^* $(1 \le i \le n)$ и вещественные числа μ_{ij} $(1 \le j < i \le n)$ определяются по индукции формулами

$$b_i^* = b_i - \sum_{j=1}^{i-1} \mu_{ij} b_j^*, \qquad (2.2.4)$$

$$\mu_{ij} = (b_i, b_j^*)/(b_j^*, b_j^*) \qquad (2.2.5)$$

Отметим, что b_i^* – проекция вектора b_i на ортогональное дополнение к пространству $\sum_{j=1}^{i-1} R\, b_j$ в пространстве $\sum_{j=1}^{i} R\, b_j$ и что $\sum_{j=1}^{i} R\, b_j = \sum_{j=1}^{i} R\, b_j^*$

для $1 \le i \le n$. Таким образом векторы $b_1^*, \ldots, b_n^*$ образуют ортогональный базис пространства R^n.

В дальнейшем символ $|\ |$ используется как для обозначения абсолютной величины вещественных или комплексных чисел, так и для обозначения евклидовой длины вектора в вещественном векторном пространстве.

Упражнение 2.2.6. Показать, что

$$|\det(b_1^*, \ldots, b_n^*)| = d(L) = \prod_{i=1}^{n} |b_i^*|.$$

Упражнение 2.2.7. Показать, что для любого базиса $b_1, \ldots, b_n$ решетки L выполняется неравенство Адамара

$$d(L) \le \prod_{i=1}^{n} |b_i|.$$

Определение 2.2.8. Базис $b_1, \ldots, b_n$ решетки L называется редуцированным, если выполняются неравенства

$$|\mu_{ij}| \le 1/2 \quad \text{для} \quad 1 \le j \le i \le n \qquad (2.2.9)$$

и для $1 < i \le n$

$$|b_i^* + \mu_{ii-1} b_{i-1}^*|^2 \ge \frac{3}{4} |b_{i-1}^*|^2 . \qquad (2.2.10)$$

Векторы $b_i^* + \mu_{ii-1} b_{i-1}^*$ и b_{i-1}^* имеют простой геометрический смысл – это проекции векторов b_i и b_{i-1} на ортогональное дополнение к пространству $\sum_{j=1}^{i-2} R\, b_j$ в $\sum_{j=1}^{i} R\, b_j$. Константа 3/4 выбирается в значительной мере произвольно: вместо нее можно взять любое фиксированное вещественное число y, удовлетворяющее условию $1/4 < y < 1$.

Грубо говоря, редуцированный базис состоит из "почти ортогональных" векторов, расположенных в порядке "почти неубывания длин".

Использование редуцированных базисов решеток для целей факторизации многочленов основано

на следующем свойстве таких базисов: если $b_1, \ldots, b_n$ – редуцированный базис решетки L, то $|b_1|^2 \le 2^{n-1}\,|x|^2$ для любого вектора $x \in L$. К доказательству этого свойства и его обобщений сейчас и переходим.

Предложение 2.2.11. Пусть $b_1, \ldots, b_n$ – редуцированный базис решетки L в R^n и векторы $b_1^*, \ldots, b_n^*$ получены из этого базиса процессом ортогонализации Грама–Шмидта. Тогда

$$|b_j|^2 \le 2^{i-1}\cdot|b_i^*|^2 \text{ для } 1 \le j \le i \le n \qquad (2.2.12)$$

$$d(L) \le \prod_{i=1}^{n} |b_i| \le 2^{n(n-1)/4}\cdot d(L)\,. \qquad (2.2.13)$$

$$|b_1| \le 2^{(n-1)/4}\cdot d(L)^{1/n} \qquad (2.2.14)$$

Доказательство. Сначала докажем формулу 2.2.12. Из формул 2.2.9 и 2.2.10 получаем

$$|b_i^*|^2 \ge (3/4 - \mu_{i\,i-1}^2)\cdot|b_{i-1}^*|^2 \ge 1/2\cdot|b_{i-1}^*|^2$$

для $1 < i \le n$, откуда по индукции выводится неравенство

$$|b_i|^2 = |b_i^*|^2 + \sum_{j=1}^{i-1}\mu_{ij}^2\,|b_j^*|^2 \le$$

$$\le |b_i^*|^2 + \sum_{j=1}^{i-1}\frac{1}{4}\cdot 2^{i-j}|b_j^*|^2 =$$

$$= (1 + \tfrac{1}{4}(2^i - 2))\cdot|b_i^*|^2 \le 2^{i-1}\cdot|b_i^*|^2.$$

Из этих формул следует, что

$$|b_i|^2 \le 2^{j-1}\cdot|b_j^*|^2 \le 2^{i-1}\cdot|b_i^*|^2$$

для $1 \le j \le i \le n$. Таким образом формула 2.2.12 доказана.

Для доказательства формулы 2.2.13 достаточно воспользоваться упражнением 2.2.6 и неравенствами $|b_i^*| \le |b_i| \le 2^{(i-1)/2}\cdot|b_i^*|$.

Полагая в формуле 2.2.12 $j=1$, и взяв произведение по i от 1 до n, получим неравенство 2.2.14. Этим заканчивается доказательство предложения 2.2.11. □

Упражнение 2.2.15. Показать, что если в формуле 2.2.10 заменить 3/4 на некоторое вещественное число y, $1/4<y<1$, то появляющиеся в формулах 2.2.12, 2.2.13 и 2.2.14 степени числа 2 заменятся на такие же степени числа $4/(4y-1)$.

Предложение 2.2.16. Пусть $b_1, \ldots, b_n$ – редуцированный базис решетки L. Тогда для любого ненулевого вектора $x \in L$ выполняется неравенство

$$|b_1|^2 \le 2^{n-1} \cdot |x|^2.$$

Доказательство. Любой вектор $x \in L$ может быть выражен через векторы базиса $b_1^*, \ldots, b_n^*$ с целыми коэффициентами r_i, а через векторы $b_1, \ldots, b_n$ – в виде линейной комбинации с вещественными коэффициентами r_i', т.е. $x = \sum_{i=1}^{n} r_i b_i = \sum_{i=1}^{n} r_i' b_i^*$. Если i – наибольший индекс, для которого $r_i \neq 0$, то $r_i' = r_i \ge 1$. Таким образом,

$$2^{n-1}|x|^2 \ge 2^{n-1} r_i'^2 \cdot |b_i^*|^2 \ge 2^{n-1} |b_i^*|^2 \ge$$

$$\ge 2^{i-1} |b_i^*|^2 \ge |b_1|^2$$

Последние два неравенства вытекают из формулы 2.2.12. □

Обобщением полученного результата является следующее

Предложение 2.2.17 Пусть $b_1, \ldots, b_n$ – редуцированный базис решетки L, $x_1, \ldots, x_t$ – линейно независимые векторы решетки L. Тогда для любого j от 1 до t выполняется неравенство

$$|b_j|^2 \le 2^{n-1} \cdot \max\{|x_1|^2, \ldots, |x_t|^2\}.$$

Доказательство. Выразим векторы x_j через элементы базиса b_i: $x_j = \sum_{i=1}^{n} r_{ij} b_i$, где $r_{ij} \in Z$ $(1 \le i \le n)$ для $1 \le j \le t$. Для каждого фиксированного j через i(j) обозначим наибольшее значение i, для которого $r_{ij} \neq 0$. Перенумеруем векторы x_j так, чтобы числа i(j) не убывали, т.е. $i(1) \le i(2) \le \ldots \le i(t)$. Из доказательства предыдущего предложения можно получить неравенство

$|x_j|^2 \geq |b^*_{i(j)}|^2$ для всех j от 1 до t. (2.2.18)

Покажем, что $j \leq i(j)$ для всех j от 1 до t. Если это неравенство для некоторого j не выполняется, то все векторы $x_1, \ldots, x_j$ принадлежат подпространству $Rb_1 + Rb_2 + \ldots + Rb_{j-1}$, что противоречит линейной независимости векторов $x_1, \ldots, x_t$.

Воспользовавшись неравенством $j \leq i(j)$ и формулами 2.2.12 и 2.2.18, получаем для всех j от 1 до t неравенство

$$|b_j|^2 \leq 2^{i(j)-1} \cdot |b^*_{i(j)}|^2 \leq$$

$$\leq 2^{n-1} \cdot |b^*_{i(j)}|^2 \leq 2^{n-1} \cdot |x_j|^2.$$

Этим доказательство предложения 2.2.17 заканчивается. □

3. РАЗЛОЖЕНИЕ НА МНОЖИТЕЛИ, СВОБОДНЫЕ ОТ КВАДРАТОВ

Разложение многочлена на множители начнем с приведения его к некоторому каноническому виду. Прежде всего найдем НОД его коэффициентов, эта величина называется содержанием многочлена f и обозначается cont(f). Далее, разложим многочлен f на "свободные от квадратов" множители, т.е. на такие множители, которые являются произведениями взаимно простых неприводимых многочленов в первой степени. Это можно сделать путем дифференцирования исходного многочлена и нахождения общих делителей многочлена и его производной. Свободный от квадратов многочлен, содержание которого равно 1, назовем примитивным.

В принятых нами обозначениях программа факторизации многочлена f принимает следующий вид:

3.1. АЛГОРИТМ факторизация_многочлена (многочлен f, цел cont(f), разложение U)

```
Арг:    f
Рез:    U, cont(f) \ содержание многочлена f
Переменные:
        G - разложение
          \ на свободные от квадратов множители
        V - разложение
          \ текущего многочлена из разложения g.
Обозначения:
        s == G.количество_множителей
        g == G.множители
        u == U.множители
        v == V.множители
```

```
начало
.  u.начать работу
.  вычислить cont(f)
.  f(x) := f(x)/cont(f)
.  разложить_на_свободные_от_квадратов_множители (f,g)
       \  f(x) = g1(x) · g2^2(x) ... gs^s(x) ,
.  цикл для i от 1 до s
.  .  разложить примитивный, свободный от квадратов на неприводимые множители (g[i], v)
   .  u.добавить v
.  конец цикла
конец
```

Вычисление содержания многочлена сводится к нахождению НОД целых чисел. Поскольку мы предполагаем коэффициенты не слишком большими, вполне достаточно ограничиться алгоритмом Евклида нахождения НОД.

Разложение примитивного (без нетривиальных общих делителей коэффициентов) многочлена на свободные от квадратов множители осуществляется следующим образом.

3.2. АЛГОРИТМ разложить_на_свободные_от_квадратов_множители (многочлен f, разложение G)

```
Арг:    f(x) ∈ Z[x], cont(f)=1
Рез:    G
Переменные:
        h, c, d - многочлены
        k - целое
Обозначения:
        s == G.количество_множителей
        g == G.множители
```

```
Начало
.  h(x) := НОД(f(x), f'(x)), где f'(x)=df(x)/dx
.  c(x) := f(x)/h(x)
.  d(x) := (df(x)/dx)/h(x) - dc(x)/dx
.  k    := 1
.  цикл пока c(x) ≠ 1
.  .  g[k](x) := НОД (c(x), d(x))
.  .  c(x)    := c(x)/g[k](x)
.  .  d(x)    := d(x)/g[k](x) - dc(x)/dx
.  .  k       := k+1
.  конец цикла
.  s  := k-1
конец
```

Для доказательства корректности этого алгоритма предположим, что $f = \prod_{i=1}^{s} g_i^i$, где g_i свободны от квадратов и взаимно простые. Тогда $h = \prod_{i=2}^{s} g_i^{i-1}$, до начала цикла

$c(x) = \prod_{i=1}^{s} g_i$, $f'/h = \sum_{i=1}^{s} (i g_i' \cdot \prod_{j=1, j\neq i}^{s} g_j)$

и $d = \sum_{i=1}^{s} ((i-1)\, g_i' \prod_{j=1, j\neq i}^{s} g_j)$.

В теле цикла выполняется присваивание многочленам c(x) и d(x) значений $c(x) = \prod_{i=k+1}^{s} g_i$ и

$$d = \sum_{i=k+2}^{s} ((i-k-1)\, f'_i \cdot \prod_{j=k+1, j\neq i}^{s} f_j).$$

Задача 3.3. Построить аналог алгоритма 3.2 для многочленов с коэффициентами из поля Fp.

Алгоритмы разложения на неприводимые множители примитивного свободного от квадратов многочлена с целыми коэффициентами составляют основу данного курса.

Современные алгоритмы разложения примитивного свободного от квадратов многочлена $f(x) \in Z(x)$ на неприводимые множители основаны на следующих соображениях. Кольцо целых чисел Z вкладывается в полное нормированное поле K. Предполагается, что мы умеем раскладывать на множители многочлены из кольца K[x], т.е. для любого наперед заданного числа $\delta > 0$ можем вычислить с абсолютной точностью δ коэффициенты всех неприводимых делителей данного многочлена (предполагается некоторая нормировка делителей, например, равенство единице старшего коэффициента). Каждому неприводимому в K[x] делителю h(x) многочлена f(x) соответствует однозначно определенный неприводимый в Z[x] делитель g(x) многочлена f(x), который делится на h(x) (более точно, g(x) представляет собой произведение нескольких неприводимых в K[x] делителей многочлена f(x), если, конечно, сам h(x) не принадлежит Q[x]). Для нахождения неприводимого в Z[x] делителя многочлена f(x) либо используют перебор произведений различных подмножеств неприводимых в K[x] делителей многочлена f(x), либо для восстановления g(x) по h(x) пользуются следующим методом. Ограничивают возможную степень многочлена g(x) положительным числом m; выделяют свободный Z-модуль ранга m+1 в модуле многочленов с целыми коэффициентами степени не выше m, в котором должен находиться искомый многочлен g(x), в частности, выделенный модуль может совпадать со всем множеством многочленов степени не выше m, вкладыва-

ют этот модуль в евклидово пространство над полем Q так, чтобы многочлену g(x) соответствовал кратчайший вектор в выделенном модуле, называемом обычно решеткой, и находят этот кратчайший вектор.

4. ВЫДЕЛЕНИЕ ЛИНЕЙНЫХ МНОЖИТЕЛЕЙ

Прежде чем переходить к общим алгоритмам разложения многочленов на неприводимые множители, рассмотрим случай, когда у многочлена имеются линейные множители. Нахождение линейных множителей осуществляется значительно проще, чем в общем случае нахождение неприводимых множителей. В большинстве систем компьютерной алгебры прежде, чем применять общие методы факторизации, у многочлена выделяются линейные множители.

Нахождение линейных множителей основано на теореме Безу, которая утверждает, что если рациональное число m/n, где m – целое, n – натуральное, НОД(m,n)=1, является корнем многочлена с целыми коэффициентами, то n делит старший коэффициент этого многочлена, а m делит его свободный член. Кроме того, между рациональными корнями многочлена и его линейными множителями существует взаимно однозначное соответствие: m/n является корнем многочлена $f(x) \in Z[x]$ тогда и только тогда, когда f(x) делится на nx – m (предполагается, что m и n взаимно простые числа).

4.1. АЛГОРИТМ рациональные_корни (многочлены f,g, стек M)

Арг: f(x) – исходный многочлен

Рез: M – стек элементов типа Q
\ множество рациональных корней многочлена f(x).
g(x) – делитель максимальной степени исходного многочлена, не имеющий рациональных корней.

```
Начало
.  М.начать работу
.  a := f.старший коэффициент
.  b := f.свободный член
.  g(x) := f(x)
.  цикл для всех пар (p,q), где p - натуральное
      число, q - целое, p делит a, q делит b,
      НОД(p,q)=1
.  .  разделить g(x) на px-q с остатком, т.е.
      g(x) = (px-q)*h(x) + r
.  .  если r=0, то
.  .  .  М.добавить p/q
.  .  .  g(x) := h(x)
.  .  конец если
.  конец цикла
конец
```

4.2. Организация перебора

Простейший случай:

```
.  цикл для p от 1 до a
.  .  если p делит a, то
.  .  .  цикл для q от 1 до b
.  .  .  .  если (q делит b и НОД(p,q)=1), то
.  .  .  .  .  цикл для j=-1;1
.  .  .  .  .  .  разделить g(x) на px-jq с ос-
                  татком, т.е. g(x) = (px-jq)*h(x) + r
.  .  .  .  .  .  если r=0, то
.  .  .  .  .  .  .  М.добавить (p,q)
.  .  .  .  .  .  .  g(x) := h(x)
.  .  .  .  .  .  конец если
.  .  .  .  .  конец цикла
.  .  .  .  конец если
.  .  .  конец цикла
.  .  конец если
.  конец цикла
```

4.3. Перебор с предварительным разложением старшего коэффициента и свободного члена на простые множители

Предполагается, что количество простых чисел, на которые делятся a или b, невелико (не превосходит NDEL). Эти числа располагаются в массиве del, соответствующие показатели степеней - в pow1 и pow2. Числа p и q задаются векторами

curr1 и curr2, которые содержат показатели степеней простых делителей чисел p и q соответственно.

```
.  curr1 := Ø
.  p := Ø
.  M.начать работу
.  конец_p_перебора := "нет"
.  цикл пока не конец_p_перебора
.  .  q :=1
.  .  curr2 := Ø
.  .  конец_q_перебора := "нет"
.  .  цикл пока не конец_q_перебора
.  .  .  ДЕЛЕНИЕ (g,p,q,успех)
.  .  .  если успех, то
.  .  .  .  M.добавить (p,q)
.  .  .  .  цикл для i от 1 до NDEL
.  .  .  .  .  pow1[i] := pow1[i] - curr1[i]
.  .  .  .  .  pow2[i] := pow2[i] - curr2[i]
.  .  .  .  конец цикла
.  .  .  конец если
.  .  .  если q > Ø
.  .  .  .  то q := -q
.  .  .  .  иначе  NEXTQ
.  .  .  конец если
.  .  конец цикла
.  .  NEXTP
.  конец цикла
```

4.4. АЛГОРИТМ NEXTP

```
начало
.  конец_p_перебора := "да"
.  цикл для i от 1 до NDEL пока конец_p_перебора
.  .  если curr1[i] < pow1[i]
.  .  .  то curr1[i] := curr1[i] + 1
.  .  .     конец_p_перебора := "нет"
.  .  .  иначе curr1[i] := Ø
.  .  конец если
.  конец цикла
.  p := 1
.  если не конец_p_перебора, то
.  .  цикл для i от 1 до NDEL
.  .  .  p := p * del[i]^curr1[i]
.  .  конец цикла
.  конец если
конец
```

4.5. АЛГОРИТМ NEXTQ

```
начало
.  конец_q_перевора := "да"
.  цикл для i от 1 до NDEL пока конец_q_перевора
.  .  если (curr2[i] < pow2[i]  и pow1[i]=0)
.  .  .  то curr2[i] := curr2[i] + 1
.  .  .     конец_q_перевора := "нет"
.  .  .  иначе curr2[i] := 0
.  .  конец если
.  конец цикла
.  q := 1
.  если не конец_q_перевора, то
.  .  цикл для i от 1 до NDEL
.  .  .  q := q * del[i]^curr2[i]
.  .  конец цикла
.  конец если
конец
```

5. ФАКТОРИЗАЦИЯ, ОСНОВАННАЯ НА ПЕРЕБОРЕ НЕПРИВОДИМЫХ СОМНОЖИТЕЛЕЙ В K[x]

5.1. ОБЩАЯ СХЕМА

Рассмотрим подробнее первый из упоминавших-ся выше методов, основанный на полном разложении многочлена в K[x] и переборе возможных комбина-ций неприводимых в K[x] сомножителей. Достаточно научиться находить один неприводимый множитель многочлена f(x).

5.1.1. АЛГОРИТМ выделить_неприводимый_множитель (многоч f,g)

Арг: f(x) $\in$ Z[x]
Рез: g(x) $\in$ Z[x], g(x) неприводим в Z[x]
Переменные: U - разложение
Обозначение: r == U.количество_множителей

Начало
. выбрать полное нормированное поле K, содержа-щее Z
. $f_1(x)$:= f(x)/ldcf(f)
  \ldcf(f) - старший коэффициент многочлена f
. разложить на неприводимые множители в K[x] ($f_1(x)$, U)
. найден множитель := "нет"
. цикл для t от 1 до [r/2] пока не найден мно-житель
. . цикл для всех подмножеств $\{i_1, \ldots, i_t\}$ множества {1, ..., r} пока не найден множитель
. . . если $g(x)=ldcf(f)\cdot u_{i_1}(x)\cdot u_{i_2}(x)\ldots u_{i_t}(x) \in$ $\in$ Z[x], то
. . . . g(x).разделить на НОД(коэффициентов)
. . . . найден множитель := "да"
. . . конец если
. . конец цикла
. конец цикла
конец

Для обоснования этого алгоритма достаточно воспользоваться результатами следующих упражне-ний.

Упражнение 5.1.2. Показать, что в кольце K[x] имеет место однозначность разложения на неприводимые множители.

Упражнение 5.1.3. Показать, что любой неприводимый в Q[x] делитель многочлена f(x) представляется в виде произведения некоторого количества многочленов $u_i(x)$ в кольце K[x], хотя бы один неприводимый в Q[x] множитель многочлена f может быть представлен в виде произведения не более [r/2] множителей $u_i(x)$.

Упражнение 5.1.4. Показать, что если $f(x) \in Z[x]$, $g(x) \in Q[x]$, $ldcf(g) = 1$ и g(x) делит f(x) в Q[x], то $ldcf(f).g(x) \in Z[x]$.

Упражнение 5.1.5. Показать, что если $f(x), g(x) \in Z[x]$, и g(x) делит f(x) в кольце Q[x], то g(x)/cont(g) делит f(x) в Z[x].

Поскольку на компьютере мы не можем реализовать элементы поля K (можем иметь дело только с конечным подмножеством рациональных чисел), то нам необходимо выбрать достаточную точность вычислений, т.е. решить следующую задачу.

Найти рациональные числа δ и μ, такие, что если коэффициенты неприводимых над K многочленов вычислены с точностью δ, то коэффициенты многочлена g(x), вычисляемого в предложении "если" описанного выше алгоритма, отличаются не более, чем на μ от целых чисел тогда и только тогда, когда многочлен g(x) – делитель f(x) в Q[x].

Более точно: предположим, что поле K является конечномерным пространством над полем $\bar{Q}$ – пополнением поля рациональных чисел по метрике, согласованной с метрикой поля K. Таким образом, множество многочленов из K[x] степени не выше m – конечномерное векторное пространство над $\bar{Q}$. Пусть $e_1, \ldots, e_s$ – базис этого пространства, $v = v_1.e_1 + \ldots + v_s.e_s$. Определим $|v| = \max_{i=1}^{s} |v_i|$ для любого вектора v из этого пространства. Требуется определить положительные числа δ и μ, исходя из следующих условий.

Пусть $w_1(x), \ldots, w_r(x) \in Qe_1 + \ldots + Qe_s$, $|w_i - u_i| < \delta$ для i от 1 до r, тогда

$$|ldcf(f).u_{i_1}(x).u_{i_2}(x) \ldots u_{i_t}(x) - ldcf(f).w_{i_1}(x).w_{i_2}(x) \ldots w_{i_t}(x)| < \mu$$

если и только если
$ldcf(f).u_{i_1}(x).u_{i_2}(x) \ldots u_{i_t}(x) \in Z[x]$.

Задача 5.1.6. Пусть $f(x) \in Z[x]$, deg f = m, абсолютная величина коэффициентов f(x) не превосходит F. Определить достаточные значения δ и μ, если в качестве поля K выбрано: a) поле вещественных чисел R; b) поле комплексных чисел C.

Значительно меньшая точность понадобится, когда потребуем только выполнения условия: если

$g(x) = ldcf(f).u_{i_1}(x).u_{i_2}(x) \ldots u_{i_t}(x) \in Z[x]$,

то

$|ldcf(f).w_{i_1}(x).w_{i_2}(x) \ldots w_{i_t}(x) - g(x)| < 1/2$.

В этом случае отброс вариантов производится не путем сравнения коэффициентов с целыми числами, а пробным делением f(x) на произведение многочленов из K[x], коэффициенты которого округлены до ближайшего целого числа.

Задача 5.1.7. Определить достаточное значение δ для этого метода, если в качестве поля K выбрано: a) поле вещественных чисел R; b) поле комплексных чисел C.

Детализируя предложенный алгоритм, рассмотрим задачу разложения на неприводимые множители многочлена $f(x) \in K[x]$. Пользуемся некоторым итерационным алгоритмом разложения на неприводимые множители:

Найти нулевое приближение разложения
Оценить необходимую точность вычислений
Цикл пока не достигнута требуемая точность
. выполнить шаг итерации
конец цикла

Известно, (см., например, [3]), что на поле рациональных чисел Q существуют метрики двух типов: архимедовы и p-адические. Пополнение поля Q по архимедовой метрике дает нам поле вещественных чисел R, которое можно использовать в предлагаемом алгоритме в качестве полного нормированного поля K. Наряду с этим при использовании архимедовой метрики можно в качестве поля K выб-

рать поле комплексных чисел C. При таком выборе любой неприводимый многочлен линеен. Если его старший коэффициент равен 1, то он полностью определяется своим корнем α. Для нахождения корня (вещественного или комплексного) можно воспользоваться какой-либо модификацией метода Ньютона. Трудности возникают при задании начального приближения (нам нужно найти все корни), а также при оценке достигнутой точности. Для нахождение вещественных корней можно использовать метод Штурма. Преимущество метода Штурма состоит в том, что он позволяет локализовать все вещественные корни и следить за достигнутой точностью. Обобщение метода Штурма для нахождения комплексных корней многочлена предложено Пинкертом [9].

Невысокие степени неприводимых многочленов над полем вещественных (не выше второй) и комплексных (первая) чисел приводят к большому количеству сомножителей в разложении f(x) над полем K. Таким образом, внутренний цикл может выполняться $O(2^{m-1})$ раз, кроме того, много времени и памяти потребует арифметика рациональных чисел.

Более удобным представляется использование p-адической метрики. Применение ее для решения задач факторизации стало возможным после того, как был получен достаточно эффективный метод разложения многочленов на множители над полем p-адических чисел. Этот метод состоит из двух ключевых алгоритмов: первый из них называется алгоритмом Берлекэмпа и позволяет достаточно быстро разлагать на множители многочлены с коэффициентами из конечного поля, что соответствует нахождению нулевого приближения разложения в описанном выше алгоритме. Второй алгоритм представляет собой p-адический аналог метода Ньютона и носит название леммы Гензеля. Метод факторизации, основанный на алгоритме Берлекэмпа и лемме Гензеля, принят во многих системах компьютерной алгебры. Прежде чем перейти к его изложению, отметим некоторые особенности алгоритма 5.1.1 при использовании p-адической метрики.

Выбор полного нормированного поля K сводится к выбору простого числа p, после чего в качестве поля K берется поле p-адических чисел - пополнение поля Q по p-адической метрике. Не вдаваясь в подробности строения поля p-адических

чисел, отметим только, что итерационный процесс нахождения разложения f(x) над полем p-адических чисел сводится к построению такого разложения по модулю возрастающих степеней числа p. На простое число p накладывается два условия - p не должно делить ldcf(f) и результант многочленов f(x) и df(x)/dx. Первое условие означает, что при переходе от Z к полю вычетов по модулю p степень многочлена не уменьшается, второе - что при переходе к полю вычетов многочлен остается свободным от квадратов. Отметим, что различный выбор простого числа p может дать различное количество множителей в разложении многочлена по модулю p.

Выбор необходимой точности вычислений сводится к нахождению верхней границы B для абсолютной величины коэффициентов многочлена $g(x) \in Z[x]$, который делит f(x). Эту величину можно посчитать по теореме 2.6. Далее натуральное число s определяется из условия $p^s > 2 \cdot B$.

Нулевое приближение разложения f(x) в поле p-адических чисел получается из разложения многочлена f(x) в поле вычетов по модулю p. Это разложение выполняется с помощью алгоритма Берлекэмпа.

Итерационный шаг уточнения разложения заключается в переходе от сравнения по модулю p^k к сравнению по модулю $q=p^t$, где $t > k$. Наиболее часто используется случай $t=2k$ (квадратичный подъем) или $t=k+1$ (линейный подъем). Этот переход выполняется с помощью леммы Гензеля. Итерационный процесс заканчивается, когда показатель степени t будет не меньше значения s, определенного выше. В качестве представителей системы вычетов по модулю p^t берется сбалансированная система, т.е. целые числа не превосходящие по абсолютной величине числа $(p^t - 1)/2$.

Проверка испытуемой комбинации на получение делителя многочлена f(x) осуществляется пробным делением.

Небольшая модификация предложенного алгоритма позволяет находить все неприводимые делители многочлена f(x), если при нахождении некоторого неприводимого делителя g(x) не прекращать работу алгоритма, а запомнить g(x), заменить

f(x) на f(x)/g(x) и удалить неприводимые над K многочлены, делящие g(x). При этом также уменьшится значение r.

Таким образом алгоритм 5.1.1 принимает вид

5.1.8. АЛГОРИТМ разложить_на_неприводимые (многочлен f, разложение G)

Арг: f(x) ∈ Z[x]

Рез: G \разложение на неприводимые множители многочлена f(x)

Переменные:

разложение U \ разложение f(x) над K[x]

множество M элементов типа натуральное число

Начало

- выбрать простое число p
  \ p не должно делить ldcf(f) и
  \ результант(f,f')
- B := $2^{m-1} \cdot \|f\|$
  \ оценивается необходимая
  \ точность вычислений
- s := $[\log_p B] + 1$
- q := p^s
- $f_1(x)$:= f(x)/ldcf(f) (mod q)
- найти нулевое приближение разложения ($f_1(x)$,U)
- поднять разложение U до разложения по модулю q
  \ достигается кратным применением
  \ леммы Гензеля
- r := U.количество_множителей
- G.колич_множит := G.колич_множит + 1
- t := 1
- M := {1, ..., r}
- цикл пока t ≤ [r/2]
  - цикл для всех подмножеств $\{i_1, \ldots, i_t\}$ множества M пока t ≤ [r/2]
    - g(x) := ldcf(f)·$u_{i_1}(x)$·$u_{i_2}(x)$ ... $u_{i_t}(x)$
    - g(x) := g(x)/cont(g)
    - если f(x) делится на g(x)
      - то
      - G.множитель[G.колич_множит] := g(x)
      - G.колич_множит:=G.колич_множит +1
      - f(x) := f(x)/g(x)
      - r := r - t
      - M.удалить $i_1, i_2, \ldots, i_t$
    - конец если

```
.  .  конец цикла
.  .  t := t + 1
.  конец цикла
.  G.множитель[G.колич_множит] := f(x)
конец
```

Переходим теперь к детализации предложенного алгоритма.

5.2. РАЗЛОЖЕНИЕ МНОГОЧЛЕНОВ НА НЕПРИВОДИМЫЕ МНОЖИТЕЛИ ПО МОДУЛЮ P

Этот раздел посвящен детализации предписания "Найти нулевое приближение разложения". Разделим его на два этапа:

- разложить многочлен на неприводимые множители по модулю простого p
  \результатом будет целое r и вектор u
  \элементов типа многочлен с индексом 1..r
- найти добавочные множители v[i]
  \результатом должен явиться вектор v элементов типа Z[x] с индексом 1..r, такой, что
  $$\sum_{i=1}^{r} v[i](x) \prod_{j=1, j\neq i} u[j](x) \equiv 1 \pmod{p}$$
  \при этом на элементы вектора v накладываются условия
  \ deg (v[i](x)) < deg (u[i](x))
  \вектор v понадобится нам в предписании
  \"выполнить шаг итерации"

Второй этап не представляет принципиальной трудности. Для его выполнения достаточно разложить рациональную функцию на элементарные дроби, что можно сделать методом неопределенных коэффициентов

$$1/\prod_{i=1}^{r} u_i(x) = \sum_{i=1}^{r} v_i(x)/u_i(x).$$

Можно также несколько раз воспользоваться расширенным алгоритмом Евклида.

Как и в случае простых чисел, задача разложения многочлена на простые множители безусловно сложнее, чем нахождение НОД, но если выполнять разложение по модулю некоторого простого числа,

то оно осуществляется не так сложно, как можно было бы ожидать. Значительно проще найти простые множители произвольного многочлена степени n по модулю 2, чем с помощью любого из известных методов определить сомножители произвольного n-разрядного числа в двоичной системе счисления. Этот удивительный факт - следствие алгоритма разложения, открытого в 1967 году Элвином Р. Берлекэмпом.

Пусть p - простое число. Все рассматриваемые ниже операции с многочленами будут выполняться по модулю p.

Предположим, что задан многочлен u(x), коэффициенты которого выбраны из множества {0, 1,..., p-1}. Считаем, что многочлен u(x) нормирован, т.е. его старший коэффициент равен 1 и свободен от квадратов, если его рассматривать над полем Fp. Если это условие не выполнено, то можно воспользоваться результатом задачи 3.3.

Предложение 5.2.1. Для любого многочлена $v(x) \in Fp[x]$

$$v(x^p) = [v(x)]^p$$

Доказательство. Для любых многочленов $v_1(x)$ и $v_2(x) \in Z[x]$ по модулю p выполняются равенства

$$(v_1(x) \cdot v_2(x))^p = (v_1(x))^p \cdot (v_2(x))^p,$$

$$[v_1(x)+v_2(x)]^p = v_1^p + C_p^1 v_1^{p-1} v_2 + \ldots + v_2^p =$$

$$= v_1^p(x) + v_2^p(x)$$

поскольку все биномиальные коэффициенты кратны p (т.е. в Fp обращаются в нуль). Далее, для всякого целого числа a по малой теореме Ферма имеем $a^p \equiv a \pmod p$. Поэтому, если $v(x) = v_m x^m + \ldots + v_0$, то

$$[v(x)]^p = (v_m x^m)^p + \ldots + (v_0)^p =$$

$$= v_m x^{mp} + \ldots + v_1 x^p + v_0 = v(x^p),$$

что и доказывает (5.2.1).

Идея Берлекэмпа состоит в том, чтобы воспользоваться китайской теоремой об остатках, которая справедлива для многочленов точно так же, как и для натуральных чисел. Из этой теоремы вытекает, что для любого набора $(s_1, \ldots, s_r)$ це-

лых чисел по модулю p существует единственный многочлен v(x) такой, что

$$v(x) \equiv s_i \pmod{p_i(x)}, \quad 1 \le i \le r$$
$$\deg v < \deg p_1 + \ldots + \deg p_r = \deg u. \qquad 5.2.3$$

(Отметим, что сравнение выполняется в кольце многочленов с коэффициентами из конечного поля, т.е. утверждение $g(x) \equiv h(x) \pmod{f(x)}$ означает, что разность $g(x)-h(x)$ в кольце $Z[x]$ принадлежит идеалу, порожденному элементами $f(x)$ и p.)

Если нам известен многочлен v(x), удовлетворяющий выписанной системе сравнений, то можно получить разложение u(x) на множители, используя тот факт, что если $r \ge 2$ и $s_1 \ne s_2$, то НОД$(u(x), v(x)-s_1)$ делится на $p_1(x)$ и не делится на $p_2(x)$.

Поскольку решение системы 5.2.3 может оказаться полезным для решения интересующей нас задачи разложения многочлена на множители, рассмотрим систему 5.2.3 более подробно. Прежде всего заметим, что решение v(x) этой системы удовлетворяет условию

$$(v(x))^p \equiv s_j^p \equiv s_j \equiv v(x) \pmod{p_j(x)} \quad \text{при } 1 \le j \le r,$$

поэтому

$$(v(x))^p \equiv v(x) \pmod{u(x)}, \ \deg v < \deg u \qquad 5.2.4$$

В поле Z/pZ выполняется разложение

$$x^p - x = (x-\emptyset).(x-1) \ldots (x-(p-1)),$$

доказательство которого восходит еще к Лагранжу (1771 год). Следовательно, любой многочлен v(x) удовлетворяет соотношению

$$(v(x))^p - v(x) =$$
$$= (v(x)-\emptyset)\ (v(x)-1) \ldots (v(x)-(p-1)), \qquad 5.2.5$$

в котором все операции выполняются по модулю p. Отсюда следует, что если многочлен v(x) удовлетворяет соотношению 5.2.4, то u(x) делит левую часть равенства 5.2.5, а следовательно, любой неприводимый множитель многочлена u(x) должен делить один из p взаимно простых множителей в правой части равенства 5.2.5. Значит, все решения сравнения 5.2.4 должны представляться в виде 5.2.3 при некотором выборе значений $s_1, \ldots, s_r$, т.е. у этого сравнения имеется ровно p^r решений. Таким образом, решения сравнения 5.2.4 дают нам

ключ к отысканию разложения многочлена $u(x)$ на неприводимые множители. Может показаться, что найти все решения сравнения 5.2.4 еще труднее, чем разложить $u(x)$ на неприводимые множители, однако в действительности это не так, поскольку множество решений сравнений 5.2.4 замкнуто относительно сложения, следовательно, оно является векторным пространством над полем Z/pZ.

Пусть $\deg u(x) = n$; рассмотрим матрицу размера $n.n$

$$Q = \begin{bmatrix} q_{0,0} & q_{0,1} & \cdots & q_{0,n-1} \\ \vdots & \vdots & & \vdots \\ q_{n-1,0} & q_{n-1,1} & \cdots & q_{n-1,n-1} \end{bmatrix},$$

элементы которой определяются соотношениями

$$x^{kp} \equiv q_{k,n-1}x^{n-1} + \ldots + q_{k,1}x + q_{k,0} \pmod{u(x)}.$$

Многочлен $v(x) = v_{n-1}x^{n-1} + \ldots + v_0$ является решением сравнения 5.2.4 тогда и только тогда, когда выполняется векторное равенство $(v_0, \ldots, v_{n-1}).Q = (v_0, \ldots, v_{n-1})$.

В самом деле, последнее равенство выполняется тогда и только тогда, когда

$$v(x) = \sum_j v_j x^j = \sum_j \sum_k v_k q_{k,j} x^j \equiv \sum_k v_k x^{kp} = v(x^p) =$$

$$= [v(x)]^p \pmod{u(x)}.$$

Построение матрицы Q легко можно осуществить следующим образом. Для сравнительно малых p можно воспользоваться таким методом вычисления $x^k \pmod{u(x)}$. Пусть

$u(x) = x^n + \ldots + u_1x + u_0$; и

$x^k \equiv a_{k,n-1}x^{n-1} + \ldots + a_{k,1}x + a_{k,0} \pmod{u(x)}$.

Тогда

$$x^{k+1} \equiv a_{k,n-1}x^n + \ldots + a_{k,1}x^2 + a_{k,0}x \equiv$$

$$\equiv a_{k,n-1}(-u_{n-1}x^{n-1} - \ldots - u_1x - u_0) +$$

$$+ a_{k,n-2}x^{n-1} + \ldots + a_{k,0}x =$$

$$= a_{k+1,n-1}x^{n-1} + \ldots + a_{k+1,1}x + a_{k+1,0},$$

где $a_{k+1,j} = a_{k,j-1} - a_{k,n-1}u_j$. По определению полагаем $a_{k,-1} = 0$, так что $a_{k+1,0} = -a_{k,n-1}u_0$.

Таким образом, алгоритм Берлекэмпа разложения многочлена на неприводимые множители состоит в следующем.

5.2.6. АЛГОРИТМ БЕРЛЕКЭМПА

```
   АЛГОРИТМ найти_нулевое_приближение_разложения
            (цел p, многочлен f(x), разложение U)
Арг:     p, f(x)
            \ свободный от квадратов по модулю p
Рез:     U  \ разложение f на неприводимые над
            \ полем Fp
Обозначения:
        t == U.количество_множителей
        n == f.степень
        u == U.множители
Переменные:
         Q - матрица (n x n)  элементов поля Fp
         m - целое
         v - вектор элементов типа  многочлен  с
             индексом 1..r
```

```
начало
.  сформировать матрицу Q
.  базис нуль-пространства матрицы (n,Q-I,r,v)
        \ I - единичная матрица
.  t := 1
.  m := 1
.  u[1](x) := f(x)
.  цикл пока t < r
.  .  цикл для j от 2 до r пока t < r
.  .  .  цикл для всех s от Ø до p-1 пока t < r
.  .  .  .  h(x):=НОД(u[m](x),v[j](x)-s)
.  .  .  .  если h(x)≠1, то
.  .  .  .  .  если h(x) = u[m](x),
.  .  .  .  .  .  то    выход из цикла
.  .  .  .  .  .  иначе
.  .  .  .  .  .          t := t + 1
.  .  .  .  .  .          u[t](x) := h(x)
.  .  .  .  .  .          u[m](x) := u[m](x)/h(x)
.  .  .  .  .  конец если
.  .  .  .  конец если
.  .  .  конец цикла
.  .  конец цикла
.  конец цикла
конец
```

Заметим, что для сравнительно небольших p (когда мы можем хранить таблицу обратных элементов для всех элементов поля Fp) сложность алгоритма Берлекэмпа оценивается величиной $O(pn^3)$.

Детализацию этого алгоритма начнем с рассмотрения предписания "базис нуль пространства матрицы Q-I".

5.2.7. АЛГОРИТМ базис_нуль_пространства_матрицы_Q (цел n,r, матр Q[Ø:n-1,Ø:n-1], v[1:r,Ø:n-1])

Арг: n - размерность матрицы Q
Q

Рез: r - размерность нуль-пространства матрицы Q
v - вектор элементов типа (вектор элементов из поля Fp с индексом Ø..n-1) с индексом 1..r, т.е линейно независимые векторы v[1], ..., v[r],
\ записываем v в виде матрицы
\ с индексами 1..r, Ø..n-1
такие что $v[j].Q = \vec{Ø}$

Идеи реализации:
приведение к треугольному виду операциями со столбцами, т.е. переход от Q к QxB, где B - невырожденная матрица

Переменные:
c - вектор элементов типа Z с индексом 1..n
\ $c[j] \geq Ø$ <==>
\ $q_{c[j],j} = -1$, все остальные элементы
\ этой строки = Ø

```
.  r := Ø
.  цикл для j от Ø до n-1 выполнять
.  .  c[j] := -1
.  конец цикла
.  цикл для k от Ø до n-1 выполнять
                          \поиск зависимости строк
.  .  если существует j: Ø≤j<n такое, что
      Q[k,j] ≠ Ø и c[j] <Ø,
.  .  то
.  .  .  умножить j-ый столбец матрицы Q на
      -1/Q[k,j]
```

```
. . . цикл для i от Ø до j-1 и от j+1 до n-1
. . . . цикл для всех l от Ø д n-1 выполнять
. . . . . Q[l,i] := Q[l,i] + Q[k,i]·Q[l,j]
. . . . конец цикла
. . . . c[j] := k
        \эти операции не меняют строк матрицы с
        \номерами Ø,1,...,k-1, т.к. Q[s,j] = Ø
        \для всех s < k
. . . конец цикла
. . иначе
        \матрица Q приведена к ступенчатому виду
. . . r := r+1
       \начинаем нахождение собственных векторов
. . . цикл для j от Ø до n-1
. . . . v[r,j] := Ø
. . . конец цикла
. . . цикл для  s от Ø до n-1
. . . . j:= c[s]
. . . . если j ≥ Ø, то
. . . . .       v[r,j] := Q[k,s]
. . . . конец если
. . . конец цикла
. . . v[r,k] := 1
. . конец если
. конец цикла
конец
```

5.2.8 ПРИМЕР ВЫЧИСЛЕНИЯ МАТРИЦЫ Q И НАХОЖДЕНИЯ ЕЕ НУЛЬ-ПРОСТРАНСТВА

Данный пример взят из монографии Кнута [4].

Пусть $u(x) = x^8 + x^6 + 10x^4 + 10x^3 + 8x^2 + 2x + 8$, $p = 13$. Непосредственными вычислениями проверяется, что НОД$(u(x), u'(x)) = 1$. Следовательно, $u(x)$ свободен от квадратов. $x^{\emptyset} \equiv 1 \pmod{u(x)}$, значит 1-я строка матрицы Q равна (1, Ø,..., Ø). Вычислим вторую строку, т.е. $x^{13} \equiv ? \pmod{u(x)}$. Ниже приводятся вычисления.

k	$a_{k,7}$	$a_{k,6}$	$a_{k,5}$	$a_{k,4}$	$a_{k,3}$	$a_{k,2}$	$a_{k,1}$	$a_{k,0}$
0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	1	0
2	0	0	0	0	0	1	0	0
3	0	0	0	0	1	0	0	0
4	0	0	0	1	0	0	0	0
5	0	0	1	0	0	0	0	0
6	0	1	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0
8	0	12	0	3	3	5	11	5
9	12	0	3	3	5	11	5	0
10	0	4	3	2	8	0	2	8
11	4	3	2	8	0	2	8	0
12	3	11	8	12	1	2	5	7
13	11	5	12	10	11	7	1	2

Получили вторую строку матрицы Q, записанную в обратном порядке. Продолжая подобным образом, получим остальные строки матрицы Q:

$$Q = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 1 & 7 & 11 & 10 & 12 & 5 & 11 \\ 3 & 6 & 4 & 3 & 0 & 4 & 7 & 2 \\ 4 & 3 & 6 & 5 & 1 & 6 & 2 & 3 \\ 2 & 11 & 8 & 8 & 3 & 1 & 3 & 11 \\ 6 & 11 & 8 & 6 & 2 & 7 & 10 & 9 \\ 5 & 11 & 7 & 10 & 0 & 11 & 7 & 12 \\ 3 & 3 & 12 & 5 & 0 & 11 & 9 & 12 \end{pmatrix}$$

Вычитая единичную матрицу, получим

$$Q-I = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 7 & 11 & 10 & 12 & 5 & 11 \\ 3 & 6 & 3 & 3 & 0 & 4 & 7 & 2 \\ 4 & 3 & 6 & 4 & 1 & 6 & 2 & 3 \\ 2 & 11 & 8 & 8 & 2 & 1 & 3 & 11 \\ 6 & 11 & 8 & 6 & 2 & 6 & 10 & 9 \\ 5 & 11 & 7 & 10 & 0 & 11 & 6 & 12 \\ 3 & 3 & 12 & 5 & 0 & 11 & 9 & 11 \end{pmatrix}$$

Переходим к нахождению нуль-пространства.

k=0. Первая строка нулевая, таким образом, получаем собственный вектор v[1] = (1, 0,..., 0)

k=1. В качестве допустимого значения j можно взять любое $j \neq 1$ (напомним, что нумерация столбцов начинается с 0). Удобно взять $j=5$, т.к. $a_{15} = 12 \equiv -1 \pmod{13}$. Прибавляя к j-му столбцу 5-ый столбец, умноженный на a_{1j}, $j=0, 2, 3, 4, 6, 7$, получим

0	0	0	0	0	0	0	0
0	0	0	0	0	12	0	0
11	6	5	8	1	4	1	7
3	3	9	5	9	6	0	4
4	11	2	6	12	1	8	9
5	11	11	7	10	6	1	10
1	11	6	1	6	11	9	3
12	3	11	9	6	11	12	2

Продолжая таким же образом, получим

k=2, $j=4$

0	0	0	0	0	0	0	0
0	0	0	0	0	12	0	0
0	0	0	0	12	0	0	0
8	1	3	11	4	9	10	6
2	4	7	1	1	5	9	3
12	3	0	5	3	5	4	5
0	1	2	5	7	0	3	0
11	6	7	0	7	0	6	12

k=3, $j=1$

0	0	0	0	0	0	0	0
0	0	0	0	0	12	0	0
0	0	0	0	12	0	0	0
0	12	0	0	0	0	0	0
9	9	8	9	11	8	8	5
1	10	4	11	4	4	0	0
5	12	12	7	3	4	6	7
2	7	2	12	9	11	11	2

k=4, j=7

0	0	0	0	0	0	0	0
0	0	0	0	0	12	0	0
0	0	0	0	12	0	0	0
0	12	0	0	0	0	0	0
0	0	0	0	0	0	0	12
1	10	4	11	4	4	0	0
8	2	6	10	11	11	0	9
1	6	4	11	2	0	0	10

k=5, j=0

0	0	0	0	0	0	0	0
0	0	0	0	0	12	0	0
0	0	0	0	12	0	0	0
0	12	0	0	0	0	0	0
0	0	0	0	0	0	0	12
12	0	0	0	0	0	0	0
5	0	0	0	5	5	0	9
12	9	0	0	11	9	0	10

Таким образом, матрица приведена к ступенчатому виду. Для нахождения собственных векторов в качестве свободных параметров выбираем последние две координаты. При этом получаются векторы
v[2] = (0, 5, 5, 0, 9, 5, 1, 0) и
v[3] = (0, 9, 11, 9, 10, 12, 0, 1).

Им соответствуют многочлены

$v[2](x) = x^6 + 5x^5 + 9x^4 + 5x^2 + 5x,$

$v[3](x) = x^7 + 12x^5 + 10x^4 + 9x^3 + 11x^2 + 9x.$

Находим НОД(u(x), v[2](x) − s). Получаем

$НОД(u(x), v[2](x) - 0) = x^5 + 5x^4 + 9x^3 + 5x + 5$

$НОД(u(x), v[2](x) - 2) = x^3 + 8x^2 + 4x + 12.$

При всех s отличных от 0 и 2 получаем НОД(u(x), v[2](x)−s) = 1. Поскольку при приведении матрицы Q к ступенчатому виду мы получили r = 3, продолжаем поиск неприводимых множителей. Находим, что при s = 6

$НОД\ (v[3](x) - s,\ x^5 + 5x^4 + 9x^3 + 5x + 5) = x^4 + 2x^3 + 3x^2 + 4x + 6$

при s =8,

$НОД(v[3](x) - s,\ x^5 + 5x^4 + 9x^3 + 5x + 5) = x + 3,$

при остальных значениях s этот НОД равен 1.

Таким образом, мы нашли все три неприводимых сомножителя, на которые исходный многочлен u(x) разлагается в поле вычетов по модулю 13.

5.3. ЛЕММА ГЕНЗЕЛЯ

Лемма Гензеля в своей классической формулировке, как она принята в алгебре и теории чисел, утверждает, что разложение многочлена на взаимно простые сомножители, выполненное по модулю простого числа p можно продолжить до разложения в кольце p-адических чисел. Доказательство ее можно найти, например, в монографии Ван дер Вардена [3, с. 549]. Основу доказательства составляет итерационный процесс перехода от сравнения по модулю некоторой степени p к сравнению по модулю большей степени p. Показывается, что этот переход можно выполнить за конечное число шагов, вопросам сложности в алгебраическом доказательстве уделяется мало внимания. Нас же в первую очередь интересует алгоритм этого перехода с учетом возможности его практической реализации и оценкой его времени работы. В этом разделе изложен вариант леммы Гензеля, основанный на квадратичном подъеме, т.е. на переходе от сравнения по модулю q к сравнению по модулю q^2.

Рассмотрим задачу в следующей постановке:

Дано: многочлен $f(x) \in Z[x]$, число q взаимно простое с ldcf(f), разложение многочлена f(x) на взаимно простые множители над кольцом вычетов по модулю q:

$$f(x) \equiv u_1(x) \ldots u_r(x) \pmod q \qquad (5.3.1)$$

Предполагается, что $ldcf(u_1) = ldcf(f)$, а старшие коэффициенты всех остальных множителей равны 1. Кроме того, заданы значения переменных $v_1, \ldots, v_r$ типа многочлен, удовлетворяющие условиям:

$$\deg(v_i) < \deg(u_i) \text{ для } 1 \le i \le r, \qquad (5.3.2)$$

и

$$\sum_{i=1}^{r} v_i(x)\,\tilde{u}_i(x) \equiv 1 \pmod q, \qquad (5.3.3)$$

где

$$\tilde{u}_i(x) = \prod_{j=1, j\neq i}^{r} u_i(x).$$

В п. 5.2. сказано, как найти многочлены $v_i(x)$ для первого шага алгоритма (когда $q = p$ является простым числом).

Надо: поднять это разложение до сравнения по модулю q^2, т.е. найти такие $\hat{u}_1(x), \ldots, \hat{u}_r(x)$, что $\mathrm{ldcf}(\hat{u}_1) = \mathrm{ldcf}(f)$, старшие коэффициенты всех остальных множителей равны 1, и $f_1(x) \equiv \hat{u}_r(x) \ldots \hat{u}_r(x) \pmod{q^2}$. Требуется также найти новые значения переменных $v_1, \ldots, v_r$, удовлетворяющие соотношениям 5.3.2 и 5.3.3, в которых многочлены $u(x)$ заменены на $\hat{u}(x)$, а q заменено на q^2.

В рассматриваемом алгоритме факторизации q является степенью p, а множители $u_1, \ldots, u_r$ получаются подъемом неприводимых делителей $f(x)$ по модулю p.

АЛГОРИТМ квадратичный_подъем(цел q,
многочлен f(x), разложения U,V)

Арг: f(x) ∈ Z[x]
q \ основание сравнения
U,V

Рез: q \ новое значение основания сравнения
U,V

Обозначения:
r == U.колич_множит = V.колич_множит
u == U.множители
v == V.множители

Переменные:
t(x) - многочлен
w - вектор элементов типа многочлен с индексом 1..r

Начало

$$t(x) := \left(f(x) - \prod_{i=1}^{r} u[i]\ (x)\right) \pmod{q^2}$$

\ из условий на старшие коэффициенты
\ следует, что deg t < deg f.

```
t(x) := t(x)/q
     \ деление выполняется нацело, т.к. из
     \ 5.3.1 следует, что t(x) ≡ Ø (mod q)
цикл для i от 1 до r
     \многочлены û_i(x) ищем в виде
     \ u_i(x) + q·w_i(x), для чего находим
     \ w_i(x)  такие, что deg w_i < deg u_i    и
     \ Σ_i w_i(x) ũ_i(x) ≡ t(x) (mod q)    (5.3.4)
     \ условию 5.3.4 удовлетворяют многочлены
     \ t(x)·v_i(x), но для них не выполняется
     \ ограничение по степеням при переходе
     \ от t(x)·v_i(x) к его остатку от деления
     \ на u_i(x), не изменится значение соот-
     \ ветствующего слагаемого по модулю
     \ u_i(x)·ũ_i(x) = Π_i u_i(x)
     \выполнение равенства 5.3.4 после замены
     \всех многочленов t·v_i соответствующими
     \остатками следует из того, что степени
     \и левой, и правой его части меньше сте-
     \пени многочлена f(x).
.  w[i](x) := остаток от деления t(x).v[i](x)
                 на u[i](x) по mod q
.  u[i](x) := u[i](x) + q.w[i](x)
конец цикла
     \ многочлены û_i(x) найдены, переходим к
     \ модификации многочленов v_i
t(x) := (1 - Σ_{i=1}^{r} v[i](x)·ũ[i](x))    (mod q²)
     \из условий deg v < deg u следует, что
     \ deg t < deg f.
t(x) := t(x)/q
     \деление выполняется нацело, т.к. из
     \5.3.1 следует, что t(x) ≡ Ø (mod q)
цикл для i от 1 до r
     \многочлены v̂_i(x) ищем в виде v_i(x) +
     \+ q·w_i(x), для чего находим w_i(x)
     \такие, что  deg w_i < deg u_i    и
     \ Σ_i w_i(x)·ũ_i(x) ≡ t(x) (mod q)    (5.3.5)
     \условию 5.3.5 удовлетворяют  многочлены
     \t(x)·v_i(x), но для них  не  выполняется
     \ограничение по степеням при переходе от
```

```
\ t(x).v_i(x) к его остатку от деления
\ на u_i(x), не изменится значение соот-
\ ветствующего слагаемого по модулю
\ u_i(x)·ũ_i(x) = ∏_i u_i(x)
\выполнение равенства 5.3.5 после замены
\всех многочленов t·v_i соответствующими
\остатками следует из того, что степени
\и левой, и правой его части меньше сте-
\пени многочлена f(x).
. . w[i](x) := остаток от деления t(x)·v[i](x)
                на u[i](x) по mod q
. . v[i](x) := v[i](x) + q·w[i](x)
. конец цикла
конец
```

5.4. ОБСУЖДЕНИЕ АЛГОРИТМА

Выше изложена общая схема алгоритма факторизации, основанного на разложении многочлена над полем p-адических чисел и на рассмотрении произведений неприводимых над этим полем множителей. Различные этапы алгоритма допускают некоторые вариации, часть из которых мы и рассмотрим.

Замечание 5.4.1. Наибольшее количество операций в рассмотренном алгоритме факторизации требуется при выполнении перебора множителей. Версия алгоритма, излагаемая Калтофеном в [5], предполагает, что старший коэффициент первого множителя над полем Z/pZ совпадает со старшим коэффициентом исходного многочлена $f(x) \in Z[x]$. При этом из перебора исключался первый сомножитель, перебор осуществлялся по всем подмножествам множества $\{2, \ldots, r\}$, максимальное возможное их количество равно 2^{r-1}. Если первый сомножитель разделить на ldcf(f), что можно выполнить в кольце Zp, то достаточно организовать перебор только по тем подмножествам множества $\{1, \ldots, r\}$, которые содержат не более $[r/2]$ элементов. В действительности деление на ldcf(f) происходит не в кольце Zp, а в некотором кольце вычетов Z/qZ, где q является степенью p. Целесообразно, по-видимому, это деление выполнять после подъема разложения на неприводимые множители до сравнения по модулю q. При этом коэффициенты

многочлена f/ldcf(f) увеличатся, что потребует большего времени для выполнения пробных делений, однако более существенным представляется сокращение времени работы за счет меньшего количества рассматриваемых вариантов перебора.

Замечание 5.4.2. Ограничение вариантов перебора можно организовать не по максимальному количеству сомножителей, а по максимальной степени делителя. Достаточно ограничиться степенью [n/2], где n = deg (f). При этом получается лучшее ограничение на необходимую точность разложения q. Количество рассматриваемых вариантов может быть в этом случае существенно большим.

Упражнение 5.4.3. Организовать перебор вариантов сомножителей с ограничением по суммарной степени.

Замечание 5.4.4. Как отмечалось выше, проверку, представляет ли произведение g(x) неприводимых над Qp многочленов делитель f(x) в кольце Z[x], целесообразнее производить не путем сравнивания его коэффициентов с целыми числами, что потребует значительного увеличения точности вычислений, а путем пробного деления f(x) на g(x). Напомним, что коэффициенты многочлена g(x) вычислены с какой-то точностью, т.е. по модулю некоторого числа q, и представлены целыми числами. Прежде чем выполнять деление многочлена f(x) на g(x), целесообразно проверить выполнение некоторых необходимых признаков делимости: например, свободный член многочлена g(x) должен делить свободный член многочлена f(x), можно оценить допустимую величину второго по старшинству коэффициента в g(x) и сравнить ее с фактической и т.д.

Задача 5.4.5. Пусть $f(x), g(x) \in Z[x]$, g(x) делит f(x) в кольце Q[x]. Получить оценку для абсолютной величины коэффициента, следующего за старшим в g(x), через коэффициенты многочлена f(x).

Замечание 5.4.6. В процессе деления многочленов коэффициенты частного могут получиться по абсолютной величине больше допустимых значений для коэффициентов делителя многочлена f(x). В таком случае деление нужно немедленно прекращать и переходить к следующей комбинации делителей.

Замечание 5.4.7. При изложении алгоритма уточнения решения мы пользовались квадратичным

подъемом, который позволяет переходить от сравнения по модулю q к сравнению по модулю q^2. Чтобы избежать многократного превышения достигнутой точности над требуемой, на последнем шаге можно ограничиться меньшим значением q, либо применить линейный подъем.

Замечание 5.4.8. Как отмечалось выше, одно из преимуществ использования p-адической метрики состоит в том, что неприводимые по модулю p многочлены могут иметь как угодно высокие степени. Может возникнуть предположение, что для любого многочлена $f(x) \in Z[x]$ можно выбрать простое число p так, что разложение f(x) по модулю p на неприводимые множители будет совпадать с разложением f(x) в кольце Z[x]. Эта гипотеза неверна, можно привести пример неприводимого в Z[x] многочлена сколь угодно большой степени, который по модулю любого простого p разлагается на линейные или квадратичные множители. Берлекэмпом следующая теорема приписывается Х.П.Ф.Свиннертону-Дайеру.

Теорема 5.4.9. Пусть n - целое число, а $p_1, \ldots, p_n$ - различные положительные простые числа. Тогда полином $f_{p_1 \ldots p_n}(x)$ со старшим коэффициентом, равным единице и степени 2^n, корни которого равны $e_1\sqrt{p_1}+ \ldots + e_n\sqrt{p_n}$, причем $e_i = \pm 1$ для всех $1 \le i \le n$, имеет целые коэффициенты и неприводим в Z[x]. Более того, для любого простого числа q полином $f_{p_1 \ldots p_n}(x)$ по модулю q раскладывается на неприводимые в Zq[x] многочлены степени не выше второй.

Для доказательства теоремы нам потребуется знакомство с основными фактами теории Галуа. Читателю, не знакомому с теорией Галуа, рекомендуется либо ознакомиться с нею, например, просмотрев соответствующую главу в монографии ван дер Вардена, либо пропустить доказательство данной теоремы, что можно сделать без существенного ущерба для понимания дальнейшего материала.

Пользуемся следующими обозначениями

$f_k(x) = f_{p_1 \ldots p_k}(x)$ и

$$K_k = Q(\sqrt{p_1}, \ldots, \sqrt{p_k}), \quad 1 \le k \le n.$$

Методом математической индукции докажем, что $f_n(x) \in Z[x]$, $[K_n: Q] = 2^n$ и $\theta = \sqrt{p_1} + \ldots + \sqrt{p_n}$ – примитивный элемент расширения K_n над Q.

Для n=1 эти факты очевидны.

Пусть $n > 1$. Из предположения индукции $f_{n-1}(x) \in Z[x]$ и из соотношения

$f_n(x) = f_{n-1}(x + \sqrt{p_n}) f_{n-1}(x - \sqrt{p_n})$ следует, что $f_n(x) \in Z[\sqrt{p_n}, x]$, причем коэффициенты симметричны относительно $\sqrt{p_n}$ и $-\sqrt{p_n}$. Из фундаментальной теоремы о симметрических функциях следует, что эти коэффициенты должны быть целыми. Из предположения индукции $[K_{n-1}: Q] = 2^{n-1}$ заключаем, что множество

$$B_k = \{1\} \cup \{\sqrt{p_{i_1} \cdots p_{i_j}} \mid j = 1, \ldots, k, \quad 1 \le i_1 < i_2 < \ldots < i_j \le k\}$$

образует базис линейного пространства K_k над Q при $1 \le k \le n-1$.

Покажем, что $\sqrt{p_n}$ не принадлежит линейному пространству, порожденному множеством B_{n-1}. Предположим противное. Тогда существуют рациональные числа $r_\emptyset, \ldots, r_{i_1}, \ldots, i_j$, такие, что

$$r_\emptyset + \sum_{1 \le i_1 < \ldots < i_j < n} r_{i_1, \ldots, i_j} \sqrt{p_{i_1} \cdots p_{i_j}} = \sqrt{p_n} \qquad (5.4.1\emptyset)$$

Поскольку все p_i, $1 \le i \le n$, – различные простые числа, в левой части равенства 5.4.1∅ содержится не менее двух ненулевых коэффициентов. Тогда существует p_k такое, что в одном из ненулевых слагаемых содержится $\sqrt{p_k}$, а в другом – нет. Без потери общности полагаем $p_k = p_{n-1}$. Тогда равенство 5.4.1∅ можно переписать в виде

$s_\emptyset + s_1\sqrt{p_{n-1}} = \sqrt{p_n}$, где $s_\emptyset, s_1 \in K_{n-2}$ (5.4.11)

Из линейной независимости над Q элементов множества B следует, что $s_\emptyset \neq \emptyset$ и $s_1 \neq \emptyset$. Возведя обе части равенства 5.4.11 в квадрат и выполнив несложные преобразования, получим

$\sqrt{p_{n-1}} = (p_n - s_0^2 - s_1^2 p_{n-1})/2s_0s_1 \in K_{n-2}$, что противоречит предположению индукции. Следовательно, $[K_n: K_{n-1}] = 2$ и $[K_n: Q] = 2^n$.

Покажем, что $K_n = Q(\theta)$.

Пусть $\alpha_1, \alpha_2, \ldots, \alpha_{2^{n-1}}$ – корни многочлена $f_{n-1}(x)$, причем $\alpha_1 = \sqrt{p_1} + \ldots + \sqrt{p_{n-1}}$. Рассмотрим многочлены $g_1(x) = f_{n-1}(\alpha_1 + \sqrt{p_n} - x)$ и $g_2(x) = x^2 - p_n$. Очевидно, что $g_1, g_2 \in Q(\theta)[x]$ и имеют общий корень $\sqrt{p_n}$. Однако $g_1(-\sqrt{p_n}) \neq 0$, так как все корни многочлена $f_1(x)$ лежат в поле K_{n-1}, а $\sqrt{p_n} \notin K_{n-1}$. Следовательно, НОД $(g_1, g_2) = x - \sqrt{p_n} \in Q(\theta)[x]$, значит $\theta \in$

$$\in Q(\alpha_1, \sqrt{p_n})$$ и

$Q(\theta) = Q(\alpha_1, \sqrt{p_n})$. По предположению индукции $K_{n-1} = Q(\alpha_1)$, следовательно, $Q(\theta) = K_n$. Неприводимость многочлена $f_n(x)$ следует теперь из равенства степени этого многочлена степени расширения над Q, порождаемого его корнем.

Свойство разложимости по модулю любого простого числа q выводится из следующих фактов. Все квадратные корни из элементов p_i лежат в некотором квадратичном расширении поля Z/qZ, а поскольку все квадратичные расширения поля Z/qZ изоморфны, то можно считать, что все корни многочлена f_n по модулю q лежат в поле Галуа $GF(q^2)$. Если у многочлена f_n имеется неприводимый по модулю q множитель степени $m > 2$, то его корни порождают поле $GF(q^m)$ и не могут быть элементами поля $GF(q^2)$. □

6. АЛГОРИТМЫ ФАКТОРИЗАЦИИ, ОСНОВАННЫЕ НА ВЫБОРЕ МАЛОГО ВЕКТОРА В РЕШЕТКЕ

Теперь рассмотрим второй подход к решению задачи факторизации, предложенный в п. 5.1, а именно, выделяем неприводимый в Z[x] делитель многочлена f(x) путем построения некоторой решетки и отысканием в ней "малого" вектора.

6.1. ОБЩАЯ СХЕМА ФАКТОРИЗАЦИИ

В самых общих чертах алгоритм выделения неприводимого множителя с использованием редуцированного базиса решетки имеет следующий вид:

6.1.1. АЛГОРИТМ выделить_неприводимый_множитель (многочлены f(x),g(x))

Арг: f(x) ∈ Z[x], deg f(x) = m
Рез: g(x) ∈ Z[x], g(x) неприводим в Z[x]

```
Начало
.  выбрать полное нормированное поле K,
       содержащее Z
.  ограничить сверху степень неприводимого мно-
       жителя  натуральным числом n,
       например, n:=m-1
.  определить достаточную точность вычислений
.  найти с требуемой точностью неприводимый
       h(x) ∈ K[x],  делящий f(x)
          \ В результате дальнейших вычислений бу-
          \ дет найден неприводимый над  Z  много-
          \ член, делящийся на h(x).
.  сформировать решетку L, ввести на ней
       норму !! !!
          \ Искомый многочлен g(x) должен принадле-
          \ жать  L и быть в ней вектором минималь-
          \ ной длины
.  оценить  норму  искомого  g(x),
          \ найти  B  такое,  что !!g(x)!! < B
.  определить существует ли в L вектор v такой,
       что !!v!! < B
.  если да
.  .  то найти такой вектор v; g(x) := v
.  иначе
.  .  g(x) := f(x)
.  конец если
конец
```

Некоторые дополнительные комментарии к сформулированному алгоритму.

Наиболее трудный этап в этом алгоритме заключается в нахождении минимального вектора решетки. Для решения этой задачи воспользуемся алгоритмом построения редуцированного базиса решетки. В общем случае этот алгоритм не позволяет находить минимальный вектор в решетке, но находит вектор, длина которого отличается от минимального вектора решетки не более, чем в 2^n раз, где n - размерность решетки. Таким образом, на вводимую норму $\|\ \|$ накладывается более сильное условие: $\|g(x)\|$ должна отличаться от нормы любого взаимно простого с g(x) многочлена $w(x) \in Z[x]$ не менее, чем в 2^n раз. Константа B должна быть выбрана таким образом, чтобы выполнялись неравенства $\|w(x)\| > B$ и $\|g(x)\| < B$ и алгоритм построения редуцированного базиса решетки давал положительный ответ на вопрос о существовании вектора, длина которого меньше B, в том и только в том случае, когда $\|g(x)\| < B$. Искомая норма $\|\ \|$ зависит от неприводимого над K[x] многочлена h(x), при этом нужно помнить, что многочлен h(x) мы вычисляем с некоторой точностью (не абсолютной), эта точность должна быть достаточно хорошей для того, чтобы сформулированные выше условия на $\|\ \|$ остались справедливыми.

Нахождение с требуемой точностью неприводимого множителя $h(x) \in K[x]$ можно разделить на два этапа: нахождение нулевого приближения и уточнение множителя.

Перепишем алгоритм 6.1.1 с учетом сделанных замечаний.

6.1.2. АЛГОРИТМ выделить_неприводимый_множитель (многочлены f(x),g(x))

Арг: $f(x) \in Z[x]$, $\deg f(x) = m$

Рез: $g(x) \in Z[x]$, g(x) неприводим в Z[x]

Начало

- выбрать полное нормированное поле K, содержащее Z
- ограничить сверху степень неприводимого множителя натуральным числом n, например, $n := m-1$
- определить достаточную точность вычислений

```
.  найти нулевое привлижение неприводимого
       h(x) ∈ K[x]
.  цикл пока не достигнута требуемая точность
.  .  уточнить h(x) ∈ K[x]
.  конец цикла
.  сформировать решетку L, ввести на ней
       норму ‖ ‖
.  оценить норму искомого g(x): ‖g(x)‖ < B
.  редуцировать базис решетки L
.  если L.базис[Ø] < B
.  .  то g(x) := L.базис[Ø]
.  конец если
конец
```

Рассмотрим основные особенности алгоритма факторизации при использовании архимедовой и p-адической метрики на поле Q. В первом случае в качестве поля K выберем поле комплексных чисел C, во втором - поле p-адических чисел Qp.

6.2. АРХИМЕДОВА МЕТРИКА

При использовании архимедовой метрики на поле рациональных чисел Q в качестве полного нормированного расширения поля Q возьмем поле комплексных чисел K. В комплексном случае неприводимый многочлен $h(x) \in K[x]$ является линейным, мы можем считать его нормированным, т.е. $h(x) = x-\alpha$ для некоторого $\alpha \in C$. Вычисление α сводится к нахождению комплексного корня многочлена f, что можно сделать с произвольной наперед заданной точностью. Решетка L совпадает в этом случае с Z-модулем всех многочленов с целыми коэффициентами степени $\le n$. Норму многочлена $g \in L$ определим следующим образом: $\|g\|^2 = \|g\|^2 + c \cdot |g(\alpha)|^2$, где $|\ |$ - норма комплексного числа, $\|\ \|$ - обычная евклидова норма на пространстве многочленов, а $c=c(f)$ - некоторая константа, зависящая от исходного многочлена f. При вычислениях на ЭВМ α задается в виде $s+t \cdot i$, где s и t - рациональные числа. Пусть S - точность вычисления α, т.е. $|\alpha - (\hat{s}+\hat{t} \cdot i)| < 2^{-S}$. Введенная выше норма будет зависеть от точности, с которой вычислено значение корня α. Таким образом, для обоснования алгоритма достаточно показать, что для произвольно заданного многочлена $f(x) \in Z[x]$

можно явно вычислить положительные числа S, c и B такие, что норма ‖ ‖, определенная значением корня α многочлена f, вычисленного с точностью 2^{-S}, и константой c, удовлетворяет условиям:

$$2^n \cdot \|g\|^2 < B \qquad (6.2.1)$$

если g – минимальный многочлен для α над Z,
n – ранг решетки L,

$$\|p\|^2 > B \qquad (6.2.2)$$

для любого многочлена $p \in Z[x]$, не делящегося на g.

После этого задача сводится к построению редуцированного базиса решетки. Цель данного раздела – показать, что числа B, c и S можно выбрать следующим образом:

$$B = \binom{2n}{n} \cdot \|f\|^2 + 1 ; \qquad (6.2.3)$$

$$c \geq 2^{\frac{n^2}{2}+\frac{n}{2}+4} \cdot B^{n+1/2} \cdot \|f\|^{n-1} ; \qquad (6.2.4)$$

$$S \geq \log_2(c \cdot 4n \cdot \|f\| \cdot (2 + \|f\|)^{n-1}). \qquad (6.2.5)$$

Обоснование такого выбора произведено позднее, сейчас же перепишем получившийся алгоритм с учетом выбора значений B,c,S. Отметим, что для использования приведенных выше формул нам удобно поменять местами некоторые команды алгоритма.

6.2.6. АЛГОРИТМ выделить_неприводимый_множитель (многочлены f(x),g(x))
Арг: f(x) ∈ Z[x], deg f(x) = m
Рез: g(x) ∈ Z[x], g(x) неприводим в Z[x]
Переменные:
решетка L
цел S, вещ B,c

Начало
. L.ранг := m
. n:=m-1
. B:= $\binom{2n}{n} \cdot \|f\|^2 + 1$;
. c:= $2^{\frac{n^2}{2}+\frac{n}{2}+4} \cdot B^{n+1/2} \cdot \|f\|^{n-1}$;
. S:=[$\log_2(c \cdot 4n \cdot \|f\| \cdot (2 + \|f\|)^{n-1})$] + 1.
. найти комплексный корень (f(x), α, 2^{-S})
. цикл для i от Ø до n
. . L.базис[i] := x^i
. конец цикла
. редуцировать базис (L)
. если ||L.базис[Ø]|| < B
. . то g(x) := L.базис[Ø]
. иначе
. . g(x) := f(x)
. конец если
конец

В получившейся версии алгоритма осталось детализировать два предписания :
- найти комплексный корень;
- редуцировать базис.

Нахождение комплексных корней многочлена представляет собой одну из классических задач вычислительной математики, мы не останавливаемся на ней. Алгоритм редуцирования базиса решетки изложен в следующем параграфе.

6.2.7. Обоснование выбора значений B,c,S

Пусть $f \in Z[x]$ - примитивный многочлен степени n и $\alpha \in C$ - корень многочлена f. Предположим, что $h \in Z[x]$ - минимальный многочлен для α. Очевидно, что h - неприводимый множитель многочлена f. Цель этого параграфа состоит в том, чтобы показать, как вычисление α с достаточной точностью дает нам возможность определить многочлен h.

В данном параграфе символ δ используется для обозначения степени многочлена (δf = степени

f, δh = степени h и т.д.).

Предложение 6.2.8. Пусть $s \in Z$, $s \geq 0$ и предположим, что $\tilde{\alpha}$ удовлетворяет неравенству

$$|\alpha - \tilde{\alpha}| < 2^{-s}. \qquad (6.2.9)$$

Тогда

$$|h(\tilde{\alpha})| < 2^{-s}\delta h \, \|f\| \, (2+\|f\|)^{\delta h - 1}. \qquad (6.2.10)$$

Доказательство.

Для доказательства этого предложения потребуются оценки $|\alpha| \leq \|f\|$ (так как α – корень многочлена f) и оценки коэффициентов многочлена h:

$$|h_j| \leq \binom{\delta h}{j} \cdot \|f\| \qquad (6.2.11)$$

Собственно доказательство проводится путем разложения h в ряд Тейлора в окрестности точки α и несложных вычислений с использованием приведенных оценок.

Следующий этап вычислений заключается в приближении с точностью 2^{-s} значений $\tilde{\alpha}^i$ рациональными числами (чтобы не проводить вычислений со слишком большими знаменателями). После этого, выбрав произвольным образом рациональное число c, можем вложить многочлены с целыми коэффициентами степени не выше m (рассматриваем случай $m=\delta f$) в $(m+3)$-мерное пространство над полем Q, при этом образ модуля многочленов образует в этом пространстве решетку размерности $m+1$. Эту решетку обозначаем L. Наша задача заключается в выборе констант s и c таким образом, чтобы кратчайший вектор этой решетки давал нам неприводимый множитель многочлена f. (Квадрат нормы элемента решетки равен сумме квадрата нормы многочлена и приближенного значения квадрата модуля значения многочлена в точке α, умноженного на c^2.)

Предложение 6.2.12. Пусть $g \in Z[x]$ – многочлен степени не выше m такой, что НОД(h,g)=1. Предположим, что $\delta h \leq m$ и что

$$2^{\frac{m^2}{2}+\frac{m}{2}+4} B^{1/2+m} \|f\|^{m-1} \leq c \leq$$

$$\leq \frac{2^s}{4m \, \|f\| \, (2+\|f\|)^{m-1}}, \qquad (6.2.13)$$

где $B = \binom{2m}{m} \|f\|^2 + 1$. Тогда $\|\hat{h}\|^2 < B$ и $\|\hat{g}\|^2 \geq 2^m B$.

Доказательство. Первым делом покажем, что $\|\hat{h}\|^2 < B$. Так как $\|\hat{h}\|^2 = \|h\|^2 + c^2 |\tilde{h}(\tilde{\alpha})|^2$ и $|\tilde{h}(\tilde{\alpha})| \leq |h(\tilde{\alpha})| + |h(\tilde{\alpha}) - \tilde{h}(\tilde{\alpha})|$, получаем $\|\hat{h}\|^2 \leq \|h\|^2 + c^2(|h(\tilde{\alpha})|^2 + 2|h(\tilde{\alpha})||h(\tilde{\alpha}) - \tilde{h}(\tilde{\alpha})| + |h(\tilde{\alpha}) - \tilde{h}(\tilde{\alpha})|^2)$.

Оценки на слагаемые:

$$|h(\tilde{\alpha})| < \frac{1}{2c}, \quad |h(\tilde{\alpha}) - \tilde{h}(\tilde{\alpha})| < \frac{1}{2c},$$

$$\|h\|^2 \leq \binom{2\delta h}{\delta h} \|f\|^2,$$

оставляются читателю в качестве упражнения.

Доказательство неравенства $\|\hat{g}\|^2 \geq 2^m B$. Предполагаем, что $\|g\|^2 < 2^m B$ (в противном случае неравенство очевидно). Должны доказать, что $c^2 |\tilde{g}(\tilde{\alpha})|^2 \geq 2^m B$. Поскольку

$$|g(\tilde{\alpha}) - \tilde{g}(\tilde{\alpha})| \leq 2^{-s+\frac{m}{2}} \cdot (m+1) \cdot B^{1/2}$$

и $2^{-s}(m+1) \leq \frac{1}{c}$,

то достаточно доказать неравенство $c|g(\tilde{\alpha})| \geq 2(2^m B)^{1/2}$.

Найдем $a, b \in Z[x]$, такие, что $\delta a < \delta g$ и $\delta b < \delta h$, и $ah + bg = r$, где $R \in Z$ обозначает результант многочленов h и g. Из неравенства Адамара (2.2.7) следует, что абсолютные значения коэффициентов a и b ограничены величиной

$\|h\|^{m-1} \|g\|^m$, а значит и $2^{\frac{m^2}{2}} B^m$.

Отсюда, пользуясь тем, что $\|f\| - 1 \geq \|f\|/4$, можно получить неравенство

$$\max(|a(\tilde{\alpha})|, |b(\tilde{\alpha})|) < 2^{2+\frac{m^2}{2}} B^m \|f\|^{m-1}.$$

Отсюда $|a(\tilde{\alpha})h(\tilde{\alpha})| < 1/2$ и $|g(\tilde{\alpha})| \geq 1/(2|b(\tilde{\alpha})|)$.

6.2.14. Обсуждение алгоритма

Отметим, что введенные выше константы B, c и S можно вычислять не для максимально возможной степени делителя многочлена f, т.е. m-1, а для текущей степени n. При этом точность вычислений на промежуточных этапах понизится, соответственно скорость счета увеличится, кроме того с большой вероятностью нам не придется считать до максимального значения n. Так будет, если неприводимый множитель, соответствующий корню α, имеет степень меньше, чем m-1. С учетом этого замечания и использованием алгоритма редуцирования базиса решетки вышеприведенный алгоритм принимает вид:

6.2.15. АЛГОРИТМ выделить_неприводимый_множитель (многочлены f(x),g(x))

```
Арг:    f(x) ∈ Z[x], deg f(x) = m
Рез:    g(x) ∈ Z[x], g(x) неприводим в Z[x]
Переменные:
        решетка L
-------------------------------------------------
Начало
.  вычислить начальное приближение α° корня α  и
     значение  S° такое, что в шаре с центром α°
     радиуса S° содержится ровно один корень
     многочлена f
.  L.базис[Ø] := 1
.  успех := "нет"
.  цикл для n := 1..m-1 пока не успех
.  .  L.базис[n] := x^n
.  .  минимальный многочлен (L, успех)
.  конец цикла
.  если успех
.  то
.  .  g(x) := L.базис[Ø]
.  иначе
.  .  g(x) := f(x)
.  конец если
Конец программы
```

6.2.16 АЛГОРИТМ минимальный_многочлен
(решетка L, логич успех)

Арг: L \решетка, заданная диагональным \базисом
Рез: УСПЕХ - переменная типа да/нет
L \решетка, в которой построен \редуцированный базис
Глобальные переменные: f - многочлен
α - комплексно-рациональное число

Начало
. вычислить B
. вычислить c
. вычислить S
. уточнить корень α
. редуцировать базис (L)
. успех := (||L.базис[Ø]||² < B)
Конец

6.3 P-АДИЧЕСКАЯ МЕТРИКА

Переходим к подробному изложению алгоритма факторизации многочленов от одной переменной, основанному на использовании p-адической метрики и построении редуцированного базиса решетки. Предполагаем, что мы нашли неприводимый по модулю некоторого простого числа p множитель многочлена $f(x) \in Z[x]$, и что мы подняли этот неприводимый множитель до некоторого множителя $h(x)$, делящего многочлен $f(x)$ по модулю некоторой степени числа p. Предположим также, что старший коэффициент многочлена $h(x)$ равен 1 и что многочлен $f(x)$ не делится на $h^2(x)$ по модулю p. Таким образом, предполагаем, что

$$ldcf(h) = 1 \qquad (6.3.1)$$

$(h \bmod p^k)$ делит $(f \bmod p^k)$ в $(Z/p^kZ)[x]$ (6.3.2)

$(h \bmod p)$ неприводим в $Fp[x]$ (6.3.3)

$(h \bmod p)^2$ не делит $(f \bmod p)$ в кольце $Fp[x]$ (6.3.4)

Положим $l = \deg(h)$, тогда $Ø < l \le n = \deg(f)$.

Покажем, что множество многочленов $g(x) \in Z[x]$, которые делятся по модулю p на многочлен $h(x)$, образуют в $Z[x]$ главный идеал, порожденный некоторым неприводимым множителем $h_Ø(x)$ многоч-

лена f(x).

Другими словами, пусть φ_k обозначает естественный гомоморфизм кольца Z[x] на фактор-кольцо $(Z/p^kZ)[x]$, ядро гомоморфизма φ_k совпадает с главным идеалом (p^k) кольца Z[x], φ_1 обозначим просто φ. Пусть H – главный идеал кольца Z[x], порожденный многочленом h, удовлетворяющим условиям 6.3.1 – 6.3.4. Тогда существует $h_\emptyset \in Z[x]$ такой, что при любом k в Z[x] совпадают идеалы $\varphi_k^{-1}(\varphi_k(H)) = \varphi^{-1}(\varphi(H)) = (h_\emptyset)$. Многочлен $h_\emptyset$ является неприводимым в Z[x] и делит f(x).

Предложение 6.3.5. Существует неприводимый в кольце Z[x] множитель $h_\emptyset(x)$ многочлена f(x), для которого (h mod p) делит $(h_\emptyset \bmod p)$, и этот множитель определен однозначно с точностью до знака. Кроме того, если $g(x) \in Z[x]$ и g(x) делит f(x), то следующие условия эквивалентны.

(h mod p) делит (g mod p) в кольце Fp[x]; (6.3.6)

(h mod p^k) делит (g mod p^k) в кольце $(Z/p^kZ)[x]$; (6.3.7)

$h_\emptyset$ делит g в кольце Z[x]. (6.3.8)

В частности, (h mod p^k) делит $(h_\emptyset \bmod p^k)$ в кольце $(Z/p^k)[x]$. В теоретико-кольцевых терминах эти условия переписываются следующим образом:

$\varphi(g) \in (\varphi(h)) \subset Fp[x]$; (6.3.6)

$\varphi_k(g) \in (\varphi_k(h)) \subset (Z/p^kZ)[x]$; (6.3.7)

$g \in (h_\emptyset) \subset Z[x]$. (6.3.8)

В частности,

$\varphi_k(h_\emptyset) \in (\varphi_k(h)) \subset (Z/p^kZ)[x]$.

Доказательство. Существование многочлена $h_\emptyset$ следует из того, что $\varphi(f)$ делится на $\varphi(h)$. Поскольку многочлен $\varphi(h)$ неприводим, на него делится $\varphi(h_i)$ хотя бы для одного из неприводимых делителей h_i многочлена f, а так как эти делители взаимно просты, то делится в точности один из них.

Поскольку φ является кольцевым гомоморфизмом, и разлагается в композицию гомоморфизмов $\psi_k \cdot \varphi_k$, где ψ_k – естественный гомоморфизмом коль-

ца $Z/p^kZ[x] \dashrightarrow Fp[x]$, как из 6.3.8, так и из 6.3.7 следует 6.3.6.

Покажем, что из 6.3.6 следует 6.3.7 и 6.3.8.

Пусть выполнено условие 6.3.6. Тогда $\varphi(f/g) \notin (\varphi(h))$ в силу 6.3.4 и однозначности разложения на множители в $Fp[x]$. Значит, $f/g \notin (h_\emptyset) \subset \varphi^{-1}(\varphi(h))$. Из однозначности разложения на множители в $Z[x]$ следует, что $g \in (h_\emptyset)$, т.е. выполнено 6.3.8.

Пусть снова выполнено условие 6.3.6. Поскольку $Fp[x]$ является областью главных идеалов, из 6.3.6 следует, что существуют $u(x), v(x) \in Fp[x]$ такие, что $u\cdot\varphi(h) + v\cdot\varphi(f/g) = 1$ в $Fp[x]$. Поскольку φ – эпиморфизм, ядро которого порождено числом p, выписанное соотношение можно поднять до равенства в кольце $Z[x]$

$$u'\cdot h + v'\cdot f/g = 1 - p\cdot w.$$

Обозначим $w' = 1 + pw + p^2w^2 + \ldots + p^{k-1}w^{k-1}$, применяя φ_k к предыдущему соотношению, умноженному на $g\cdot w'$, получим

$$\varphi_k(g\cdot w'\cdot u')\cdot\varphi_k(h) + \varphi_k(w'\cdot v')\cdot\varphi_k(f) = \varphi_k(g).$$

Из этого соотношения и 6.3.2 следует 6.3.7.□

Рассмотрим на кольце многочленов $R[x]$ фильтрацию по степеням многочленов, т.е. для любого неотрицательного целого s векторное пространство, состоящее из многочленов степени не выше s, обозначается $R_s[x]$. Введенная фильтрация индуцирует фильтрацию на кольце $Z[x]$: $Z_s[x] = Z[x] \cap R_s[x]$. $R_s[x]$ образует вещественное линейное пространство размерности s+1, а $Z_s[x]$ является в нем решеткой (свободным Z-модулем максимального ранга).

Рассмотрим целое число $m \geq 1$. Через $L = L(m,k)$ обозначим множество всех многочленов в кольце $Z[x]$, которые делятся на $h(x)$ по модулю p^k и степень которых не превосходит m, т.е. $L(m,k) = Z_m[x] \cap \varphi_k^{-1}((\varphi_k(h)))$. Другими словами, L состоит из тех многочленов, коэффициенты остатков от деления которых на $h(x)$ в p-адической метрике не превосходят p_{-k-1}, т.е. являются малыми величинами. Легко видеть, что базис решетки L образует следующее множество многочленов

$$\{p^k x^i \mid \emptyset \leq i < l\} \cup \{h\cdot x^j \mid \emptyset \leq j \leq m-l\}. \qquad (6.3.9)$$

В качестве базиса всего вещественного пространства многочленов степени не выше m удобно выбрать одночлены $\{x^i \mid 0 \le i \le m\}$, длиной многочлена назовем евклидову длину этого многочлена в выбранном базисе, который мы предполагаем ортонормированным. Отождествляем многочлен с вектором его коэффициентов в выделенном базисе. Матрица коэффициентов базиса (6.3.9) имеет в этом базисе треугольную форму и легко видеть, что $d(L) = p^{kl}$.

Покажем, что элементы решетки L с малой длиной лежат в главном идеале, порожденном многочленом $h_0(x)$ в $Z[x]$.

Предложение 6.3.10. Предположим, что многочлен $b \in L$ удовлетворяет неравенству

$$p^{kl} > |f|^m \cdot |b|^n. \qquad (6.3.11)$$

Тогда b делится на $h_0(x)$ в кольце $Z[x]$, в частности, НОД$(f,b) \neq 1$.

Доказательство. Можно считать, что $b \neq 0$. Положим $g = $ НОД(f, b). Достаточно показать, как следует из предыдущего предложения, что $\varphi(g) \in (\varphi(h))$. Предположим противное. Пользуясь неприводимостью $\varphi(h)$ и эпиморфностью гомоморфизма φ, получаем существование многочленов $u,v,w \in Z[x]$ таких, что

$$u \cdot h + v \cdot g = 1 - p \cdot w. \qquad (6.3.12)$$

Напомним, что $l = \deg(h)$, $n = \deg(f)$, $m+1$ – размерность решетки L. Положим $m' = \deg(b)$ и $e = \deg(g)$.

Очевидно, что $0 \le e \le m' \le m$. Положим $s = n+m'-l-1$.
Пусть $M_f = Z_s[x] \cap (f)$, $M_b = Z_s[x] \cap (b)$, и $M = M_f + M_b$, т.е. M является Z-модулем, состоящим из всех многочленов вида $u \cdot f + v \cdot b$, где $u \in Z_{m'-l}[x]$, $v \in Z_{n-l}[x]$.

Покажем, что множество элементов

$$\{x^i f: 0 \le i < m'-e\} \cup \{x^j b: 0 \le j < n-e\} \qquad (6.3.13)$$

образует базис Z-модуля M. Очевидно, что они порождают M, остается только показать, что выписанная система многочленов линейно независима над Z. Предположим, что $u \cdot f + v \cdot b = 0$, где

deg(u) < m'-e, deg(v) < n-e. Разделим это соотношение на g. Получим, $u\cdot(f/g) + v\cdot(b/g) = 0$. Пользуясь взаимной простотой многочленов f/g и b/g и ограничениями на степени u и v, получаем, что u=v=0.

Рассмотрим проекцию

$\pi: R_s[x] \longrightarrow R_s[x]/R_{e-1}[x]$.

Пусть $M' = \pi(M)$. Покажем, что M' - решетка в $R_s[x]/R_{e-1}[x]$. Для этого достаточно показать, что $M \cap \ker(\pi) = 0$, т.е. $M \cap Z_{e-1}[x] = 0$. Пусть $w \in M \cap Z_{e-1}[x]$, тогда $w = u\cdot f + v\cdot b$ по определению M, следовательно, w делится на g (= НОД(f,b)). Поскольку deg (w) < deg (g), получаем w = 0. Учитывая линейную независимость элементов множества 6.3.13 над Z, получаем, что эти элементы образуют базис решетки M'. Неравенство Адамара (2.2.7) утверждает, что

$d(M') \le |f|^{m'-e}\cdot|b|^{n-e} \le |f|^{m}\cdot|b|^{n}$. Пользуясь предположением теоремы, получаем $d(M') < p^{kl}$.

Для получения желаемого противоречия, покажем, что из 6.3.12 следует обратное неравенство $d(M') \ge p^{kl}$.

Покажем, что для любого элемента $\mu \in M$, если deg μ < e+l, то $\vartheta_k(\mu) = 0$, т.е. $\mu \in p^k Z$. Домножим соотношение 6.3.12 на $\mu/g\cdot(1+pw+\ldots+p^{k-1}w^{k-1})$. Получим $u_1\cdot h+v_1\cdot\mu \equiv \mu/g \pmod{p^k Z[x]}$, где u_1, v_1 - некоторые многочлены из кольца Z[x]. Поскольку $\mu_k \in M$, $\vartheta(\mu)$ делится на $\vartheta_k(h)$, следовательно, $\vartheta_k(\mu/g)$ также делится на $\vartheta_k(h)$. Сравнивая степени, получаем, что $\vartheta_k(\mu) = 0$.

Для завершения доказательства достаточно теперь показать, что базис $b_l, b_{l+1}, \ldots, b_{n+m'-l-1}$ решетки M' можно выбрать таким образом, что $\deg_j(b_i) = j$. Это упражнение на приведение невырожденной целочисленной матрицы к треугольному виду оставляется читателю. При таком выборе базиса, старшие коэффициенты первых l многочленов делятся на p^k. Значит d(M'), который в полученном базисе равен произведению старших коэффициентов, удовлетворяет неравенству $d(M') \ge p^{kl}$, что завершает доказательство теоремы.

Следующий результат позволяет находить неприводимый делитель многочлена f.

Предложение 6.3.14. Пусть f,p,k,n,h,l выбраны так, как предполагалось в начале параграфа, L – решетка, заданная базисом 6.3.9. Предположим, что $b_1, \ldots, b_{m+1}$ – редуцированный базис решетки L и что выполняется неравенство

$$p^{kl} > 2^{mn/2} \binom{2m}{m}^{n/2} |f|^{m+n}. \qquad (6.3.15)$$

Если $h_\varnothing$ – неприводимый над Z многочлен, делящийся на h, то $\deg(h_\varnothing) \le m$ тогда и только тогда, когда

$$|b_1| < (p^{kl}/|f|^m)^{1/n}. \qquad (6.3.16)$$

Доказательство. Если условие 6.3.16 выполнено, то по предложению 6.3.12 многочлен b_1 делится на $h_\varnothing$. Решетка L выбрана так, что $\deg b \le m$ для любого $b \in L$, следовательно, $\deg(h_\varnothing) \le m$.

Предположим теперь, что $\deg(h_\varnothing) \le m$. Тогда $h_\varnothing \in L$ по предложению 6.3.10.

Полагая в 2.2.16 $x = h_\varnothing$, получим

$$\|b_1\| \le 2^{m/2} \cdot \|h_\varnothing\| .$$

Теперь из задачи 2.1.7 следует неравенство

$$b_1 \le 2^{m/2} \cdot \binom{2m}{m}^{1/2} \cdot \|f\| .$$

Подставляя сюда 6.3.15, получим 6.3.16.□

Теперь можно сформулировать следующий алгоритм нахождения неприводимого в Z[x] многочлена, делящегося по модулю p на неприводимый по модулю p многочлен h(x).

6.3.17. АЛГОРИТМ неприводимый_множитель
(многоч f,h,g, цел p)

```
Арг:    p,f,h   \h - неприводимый по модулю p
                \многочлен, делящий f по модулю p
Рез:    g       \неприводимый над Z многочлен,
                \делящий f и делящийся по модулю
                \p на h.
Обозначения:
        n == f.степень
        l == h.степень
Переменные:
        целые k
        решетка L
```

```
Начало
.  найден множитель := "нет"
.  цикл для m от 1 до n-1 пока не найден множитель
.  .  вычислить k такое, чтобы выполнялось неравенство 6.3.15
.  .  пользуясь леммой Гензеля поднять сравнение
      h·u ≡ f (mod p)
        до сравнения по модулю p^k
.  .  получить базис решетки L по формулам 6.3.9
.  .  редуцировать базис решетки L
.  .  если L.базис[1] удовлетворяет 6.3.16
.  .  то
.  .  .  g.степень := m
.  .  .  g.коэффициенты := L.базис[1]
.  .  .  найден множитель := "да"
.  .  конец если
.  конец цикла
конец программы
```

Недостаток этого алгоритма заключается в том, что для каждого значения l нужно применять алгоритм Гензеля, строить новую решетку и редуцировать ее базис. Следующее предложение позволяет применять алгоритм построения редуцированного базиса решетки только один раз для максимального возможного значения l=n-1.

Предложение 6.3.18. Предположим, что обозначения выбраны так же, как и в предложении 6.3.14, и что выполнены те же предположения.

Предположим кроме того, что существуют индексы j такие, что

$$\|b_j\| < (p^{kl}/\|f\|^m)^{1/n} . \qquad (6.3.19)$$

Пусть t – наибольшее значение j, для которого выполнено 6.3.19. Тогда

$$\deg(h_\emptyset) = m+1-t ,$$

$$h_\emptyset = НОД(b_1, \ldots, b_t) ,$$

и неравенство 6.3.19 выполнено для всех j таких, что $1 \le j \le t$.

Доказательство. Пусть J обозначает множество индексов $j \in \{1,\ldots,m\}$, для которых выполнено неравенство 6.3.19, $\#J$ – мощность множества J. Тогда $h_\emptyset$ делит b_j для любого $j \in J$, следовательно, $h_\emptyset$ делит многочлен $h_1 = НОД(\{b_j : j \in J\})$. Положим $s = m - \deg(h_\emptyset)$. Поскольку $\deg(b_j) \le m$ для всех $j \in J$, эти многочлены принадлежат Z-модулю $Z_s[x]\cdot h_1(x)$, ранг которого равен $s + 1 = m + 1 - \deg(h_1)$. В силу линейной независимости векторов b_j получаем

$$\#J \le m+1-\deg(h_1). \qquad (6.3.20)$$

Применяя результат задачи 2.1.7 к многочленам $h_\emptyset x^i$, получаем

$$\|h_\emptyset x^i\| = \|h_\emptyset\| \le \binom{2m}{m}^{1/2} \cdot \|f\| \text{ для всех } i \ge \emptyset.$$

Для $\emptyset \le i \le m-\deg(h_\emptyset)$ имеем $h_\emptyset x^i \in L$, так что из 2.2.16 получаем

$$\|b_j\| \le 2^{m/2} \cdot \binom{2m}{m}^{1/2} \cdot \|f\| \quad \text{для } 1 \le j \le m-\deg(h_\emptyset).$$

Из неравенства 6.3.15 следует, что индексы от 1 до $m + 1 - \deg(h_\emptyset)$ принадлежат множеству J. Поскольку h_1 делится на $h_\emptyset$ и выполняется неравенство 6.3.20, получаем $\{1, \ldots, m+1-\deg(h_\emptyset)\} \subset J$, откуда $\deg(h_\emptyset) = \deg(h_1) = m+1-t$.

Остается показать, что h_1 с точностью до знака совпадает с $h_\emptyset$. Для этого достаточно показать, что многочлен h_1 примитивный, т.е. наи-

больший общий делитель его коэффициентов равен 1, а примитивность многочлена h_1 легко вытекает из примитивности хотя бы одного b_j, $j \in J$.

Возьмем произвольный индекс $j \in J$ и пусть $d_j = cont(b_j)$. Тогда многочлен b_j/d_j делится на $h_\emptyset$, следовательно, $b_j/d_j \in L$, так как $h_\emptyset \in L$. Поскольку b_j принадлежит базису решетки L, получаем $d_j = 1$. □

Полученный результат используется в следующем алгоритме.

6.3.18. АЛГОРИТМ неприводимый_множитель
(многоч f,h,g, цел p)

```
Арг:    p,f,h      \h - неприводимый по модулю p
                   \многочлен, делящий f по мо-
                   \дулю p
Рез:    g          \неприводимый над Z многочлен,
                   \делящий f и делящийся по мо-
                   \дулю p на h.
Обозначения:
        n == f.степень
        l == h.степень
Переменные:
        цел k,t
        решетка L
```

```
Начало
.  найден множитель := "нет"
.  вычислить k такое, чтобы выполнялось неравен-
      ство 6.3.15 для l=m-1
.  пользуясь леммой Гензеля поднять сравнение
      h·u ≡ f (mod p) до сравнения по модулю p^k
.  построить базис решетки L по формулам 6.3.9
.  редуцировать базис решетки L
.  найти максимальное значение t, для которого
      L.базис[t] удовлетворяет 6.3.16
.  если t > ∅
.  то
.  .  g.степень := n-t
.  .  g.коэффициенты :=
         := НОД (L.базис[1], ..., L.базис[t])
.  .  найден множитель := "да"
.  конец если
конец
```

7. РЕДУЦИРОВАНИЕ БАЗИСА В РЕШЕТКЕ

В этом параграфе рассмотрим алгоритм построения редуцированого базиса решетки, полученный в работе [5]. Определение решетки и редуцированного базиса приведены в параграфе 2. Там же описаны основные свойства редуцированных базисов, которые понадобятся нам в алгоритмах факторизации многочленов.

Ниже сформулирован и обоснован алгоритм построения редуцированного базиса решетки. Построение редуцированного базиса ведем, последовательно присоединяя очередной (k-ый) элемент исходного базиса решетки и редуцируя базис подрешетки, натянутой на векторы с 1-го по k-ый. Алгоритм содержит два основных шага: на одном из них мы из присоединяемого вектора вычитаем целые кратные векторов, уже включенных в редуцированный базис, чтобы обеспечить выполнение условия 2.2.9. При этом длина редуцированной части базиса не меняется. Второй шаг, направленный на выполнение условия 2.2.10, сводится к перестановке добавляемого вектора с последним, уже включенным в редуцированный базис, при такой перестановке длина редуцированной части базиса уменьшается на 1. Переменная k указывает номер элемента, который пытаемся присоединить к редуцированной части базиса, т.е. редуцированная часть базиса содержит в каждый момент k-1 вектор. Начальное значение k равно 2, т.к. любой базис решетки, порождаемой одним вектором, является редуцированным (условия 2.2.9 и 2.2.10 выполняются автоматически, поскольку нет различных индексов).

В описании алгоритма редуцирования базиса пользуемся типом данных "решетка", который описан в параграфе 1. В отличие от принятых там обозначений, здесь значения индексов принадлежат отрезку 1..n (а не 0..n-1), т.е.

```
решетка : запись(ранг == n  : Z+
                 базис : вектор b элементов типа
                         (вектор элементов типа R
                         или Q с индексом 1..n) с
                         индексом 1..n)
```

7.1. АЛГОРИТМ редуцирование_базиса (решетка L)

```
Арг:    L     \ решетка, задаваемая исходным
              \ базисом
Рез:    L     \ решетка, задаваемая
              \ редуцированным базисом
```

Обозначения:

n == L.ранг

b == L.базис

Переменные:

μ – нижняя треугольная матрица коэффициентов, вычисляемых по формулам 2.2.4 и 2.2.5

B – вектор элементов типа R с индексом 1..n

Элементы вектора B представляют собой длины соответствующих векторов из ортогонального базиса b^*, вычисляемого по формулам 2.2.4 и 2.2.5.

```
начало
.  начальная установка (L, μ, B)
.  k := 2
.  цикл пока k ≤ n
.  .  обеспечить выполнение условия 2.2.9 для
.  .        i=k и j=k-1
.  .  если условие 2.2.10 не выполнено
.  .  .  то
.  .  .     поменять местами k-ый элемент базиса
.  .  .         b̃ с (k-1)-м
.  .  .     если k > 2
.  .  .     .  то  k := k-1
.  .  .     конец если
.  .  .  иначе
.  .  .     цикл для j от k-2 до 1 шаг -1
.  .  .     .  обеспечить выполнение условия
.  .  .                  2.2.9 для k,j
.  .  .     конец цикла
.  .  .     k := k+1
.  .  конец если
.  конец цикла
конец
```

Детализируем предложенный алгоритм.

7.2. АЛГОРИТМ начальная_установка
(решетка L, матрица μ, вектор B)

Арг: L

Рез: μ – нижняя треугольная матрица коэффициентов, вычисляемых по формулам 2.2.4 и 2.2.5.

B – вектор элементов типа вещественное число с индексом 1..n Элементы вектора B представляют собой длины соответствующих векторов из ортогонального базиса b1, вычисляемого по формулам 2.2.4 и 2.2.5.

Переменные:

b1– вектор элементов типа (вектор элементов типа R с индексом 1..n) с индексом 1..n

Обозначения:

n == L.ранг
b == L.базис \ исходный базис решетки
(a,b) обозначает скалярное произведение n-мерных векторов

```
начало
.  цикл для q от 1 до n
   .  b1[q] := b[q]
.  .  цикл для l от 1 до q-1
.  .  .  μ[q,l] := (b[q],b1[l])/B[l]
.  .  .  b1[q]  := b1[q] - μ[q,l]·b1[l]
.  .  конец цикла
.  B[q] := (b1[q],b1[q])
.  конец цикла
конец
```

Отметим, что в предлагаемой версии алгоритма переменная b1 (двумерный массив, соответствующий ортогональному базису b^*) локальна, в остальной части алгоритма этот базис в явном виде не используется. Используется вектор B, элементы которого представляют собой квадраты длин ортогонального базиса, что позволяет значительно экономить память, используемую основной программой (вместо двумерного массива хранится одномерный). При этом нужно проследить, как изменяются компоненты вектора B при различных выполняемых преобразованиях, что сделано при описании соответствующих предписаний.

7.3. АЛГОРИТМ

обеспечить_выполнение_условия_2.2.9
(решетка L, матрица μ, цел k,j)

Арг: L
k>j – индексы

Рез: μ – треугольная матрица коэффициентов, вычисляемых по формулам 2.2.4 и 2.2.5.

Переменные:
r – целое

```
начало
.  если |μ[k,j]| > 1/2, то
.  .  r := ближайшее целое к μ[k,j]
.  .  b[k] := b[k] – r·b[j]
.  .  цикл для i от 1 до j–1
.  .  .  μ[k,i] := μ[k,i] – r·μ[j,i]
.  .  конец цикла
.  .  μ[k,j] := μ[k,j] – r
.  конец если
конец
```

Элементы нижней треугольной матрицы μ вычисляются по формулам 2.2.5. В данном алгоритме меняем только вектор с индексом k. При этом меняется только k–ая строка матрицы μ, из нее вычитается j–ая строка матрицы (μ–E), умноженная на r. (E – единичная матрица.)

Прежде чем переходить к формулировке алгоритма перестановки k–го элемента базиса с (k–1)–м, выведем соответствующие формулы. Пусть $b_1, \ldots, b_n$ – текущий базис, ему соответствует ортогональный базис b^* и нижняя треугольная матрица μ. Элементы нового базиса обозначим буквой c с соответствующим индексом, соответствующий ортогональный базис – c^*, нижнюю треугольную матрицу – ν. Вычислить элементы базиса c не представляет труда:

$$c_{k-1} = b_k,\ c_k = b_{k-1},\ c_i = b_i \text{ для } i \neq k, k-1. \quad (7.4)$$

Переходим к вычислению ортогонального базиса. Как отмечалось выше, левая часть равенства 7.10 представляет собой квадрат длины ортогонального дополнения i–го вектора к подпространству, порожденному векторами с 1–го по (i–2)–ой.

Таким образом,

$$c^*_{k-1} = b^*_k + \mu_{kk-1} b^*_{k-1}. \qquad (7.5)$$

Для вычисления c^*_k спроектируем b^*_{k-1} на ортогональное дополнение к Rc^*_{k-1}. Получим

$$\nu_{kk-1} = (b^*_{k-1}, c^*_{k-1})/(c^*_{k-1}, c^*_{k-1}) =$$

$$= \mu_{kk-1} |b^*_{k-1}|^2 / |c^*_{k-1}|^2, \qquad (7.6)$$

$$c^*_k = b^*_{k-1} - \nu_{kk-1} c^*_{k-1}. \qquad (7.7)$$

Для $i \neq k-1, k$ имеем $c^*_i = b^*_i$. (7.8)

Для вычисления коэффициентов ν нам понадобится выразить старый ортогональный базис через новый. Из соотношений 7.5, 7.7 и 7.8 получаем

$$b^*_{k-1} = \nu_{kk-1} c^*_{k-1} + c^*_k \qquad (7.9)$$

$$b^*_k = (1 - \mu_{kk-1}\nu_{kk-1}) c^*_{k-1} - \mu_{kk-1} c^*_k =$$

$$(|b^*_k|^2 / |c^*_{k-1}|^2) \cdot c^*_{k-1} - \mu_{kk-1} c^*_k. \qquad (7.10)$$

Подставив соотношения 7.4 – 7.10 в формулу 2.2.4 и приведя подобные члены, получим для $i > k$

$$\nu_{ik-1} = \mu_{ik-1}\nu_{kk-1} + \mu_{ik} |b^*_k|^2 / |c^*_{k-1}|^2, \qquad (7.11)$$

$$\nu_{ik} = \mu_{ik-1} - \mu_{ik}\,\mu_{kk-1}. \qquad (7.12)$$

Наконец,

$$\nu_{k-1\ j} = \mu_{kj},\ \nu_{kj} = \mu_{k-1\ j} \text{ для } 1 \le j < k-1; \qquad (7.13)$$

$$\nu_{ij} = \mu_{ij} \text{ для } 1 \le j < i \le n,\ \{i,j\} \cap \{k-1,k\} = \emptyset. \qquad (7.14)$$

Реализация полученных формул описывается следующим алгоритмом:

7.15 АЛГОРИТМ

поменять_местами_k-ый_элемент_базиса_b_с_(k-1)-м
(цел k, решетка L, матрица μ, вектор B)

Арг: k, L, μ, B

\ μ - нижняя треугольная матрица коэффициентов, вычисляемых по формулам 2.2.4 и 2.2.5.

\ B - вектор элементов типа вещественное число с индексом 1..n Элементы вектора B представляют собой длины соответствующих векторов из ортогонального базиса b1, вычисляемого по формулам 2.2.4 и 2.2.5.

Рез: L, μ, B

\ В векторе L.базис поменялись местами два элемента. Соответствующие изменения произошли с элементами матрицы μ и вектора B.

Обозначения:

n == L.ранг

b == L.базис

Переменные:

BB, μμ - вещественные числа

начало

. μμ := μ[k,k-1]

. BB := B[k] + μ²·B[k-1]

. μ[k,k-1] := μ·B[k-1]/BB

. B[k] := B[k-1]·B[k]/BB

. B[k-1] := BB

. $\begin{bmatrix} b[k-1] \\ b[k] \end{bmatrix} := \begin{bmatrix} b[k] \\ b[k-1] \end{bmatrix}$

. цикл для i от 1 до k-2

. . $\begin{bmatrix} \mu[k-1,i] \\ \mu[k,i] \end{bmatrix} := \begin{bmatrix} \mu[k,i] \\ \mu[k-1,i] \end{bmatrix}$

. конец цикла

. цикл для i от k+1 до n

. . $\begin{bmatrix} \mu[i,k-1] \\ \mu[i,k] \end{bmatrix} :=$

$$:= \begin{bmatrix} 1 & \mu[k,k-1] \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 \\ 1 & -\mu \end{bmatrix} \begin{bmatrix} \mu[i,k-1] \\ \mu[i,k] \end{bmatrix}$$

. конец цикла

конец

Переходим к обоснованию алгоритма 7.1 построения редуцированного базиса решетки. Оно состоит из доказательства двух утверждений. Во-первых, докажем, что если алгоритм завершит свою работу, то в результате получим редуцированный базис данной решетки. Во-вторых, докажем, что для любого исходного базиса решетки алгоритм после конечного числа шагов закончит работу.

Для доказательства первого утверждения, заметим, что мы выполняем над элементами базиса только элементарные преобразования, которые переводят базис Z-модуля в другой базис того же самого Z-модуля, т.к. и перестановка элементов базиса и прибавление к одному из элементов целого кратного другого - обратимые операции. Таким образом, в любой момент времени b представляет собой базис исходной решетки. Элементы этого базиса с 1-го по (k-1)-ый представляют собой редуцированный базис подрешетки, порожденной этими элементами, т.к. для них выполнены условия 2.2.9 и 2.2.10. Когда мы производим преобразования вектора b[k], направленные на то, чтобы для него выполнялись условия 2.2.9, эти преобразования никак не отражаются на векторах с 1-го по (k-1)-ый, а если мы производим перестановку k-го элемента базиса с (k-1)-ым, то уменьшаем длину редуцированной части базиса на 1 (если она больше 1; меньше 1 длина редуцированной части базиса быть не может) и одновременно уменьшаем значение k. Для векторов с 1-го по (k-1)-ый снова выполнены условия 2.2.9 и 2.2.10. Таким образом, если алгоритм завершил свою работу, то получившийся базис - редуцированный базис исходной решетки L.

Прежде чем перейти к доказательству второго утверждения, сформулируем несколько определений и задач различной степени сложности.

Упражнение 7.16. Пусть $m(L) = \min \{|x|^2 : x \in L, x \neq 0\}$. Показать, что для любой решетки L выполняется неравенство $m(L) > 0$.

Введем обозначение

$$d_i = \det ((b_j, b_l))_{1 \le j,l \le i} \text{ для } 0 \le i \le n. \quad (7.17)$$

Упражнение 7.18. Показать, что для всех i от 1 до n выполняется равенство $d_i = \prod_{j=1}^{i} |b_j^*|^2$.

Для всех $i > 0$ числа d_i могут быть интерпретированы как квадраты детерминантов решеток ранга i, порожденных векторами $b_1, \ldots, b_i$ в векторных пространствах $\sum_{j=1}^{i} Rb_j$.

Задача 7.19. Показать, что каждая такая решетка содержит ненулевой вектор x, удовлетворяющий неравенству

$$|x|^2 \le (4/3)^{(i-1)/2} \cdot d_i^{1/i} .$$

Переходим теперь к доказательству второй части утверждения о корректности алгоритма построения редуцированного базиса. Пусть

$$D = \prod_{j=1}^{n-1} d_i .$$

Из упражнения 7.18 следует, что значение D меняется только тогда, когда изменяется хотя бы один из векторов b_i^*. Это может произойти только при перестановке двух векторов базиса. При этом новое значение $|b_{i-1}^*|$ равно $|b_i^* + \mu_{i\ i-1} b_{i-1}^*|$, что по условию меньше, чем 3/4 от прежнего значения этой величины. Таким образом, при каждой выполняемой перестановке элементов базиса значение величины D умножается на положительное число, меньшее 3/4. Из упражнений 7.16 – 7.19 следует, что D ограничено снизу некоторой положительной величиной, следовательно, перестановка элементов выполняется в алгоритме только конечное число раз, обозначим его m. При каждом выполнении перестановки значение i уменьшается на 1, при выполнении команд, следующих за ключевым словом "иначе" в алгоритме, значение i увеличивается на 1. Начальное значение i равно 2, алгоритм продолжает работу до тех пор, пока $i \le n$, следовательно ситуация "иначе" встречается m+n−1 раз, т.е. тело цикла "пока" выполняется конечное число раз. Таким образом, алгоритм завершает работу после выполнения конечного числа шагов. Ниже мы оценим это число для случая, когда все координаты исходного базиса решетки – целые числа.

Итак, переходим к оценке сложности алгоритма построения редуцированного базиса решетки в

предположении, что все координаты исходного базиса являются целыми числами. Именно такой случай нам понадобится для получения алгоритма факторизации полиномиальной сложности. Наша цель сейчас – доказательство следующего предложения.

Предложение 7.20. Пусть $L \subset Z^n$ – решетка с базисом $b_1, \ldots, b_n$ и предположим, что задано положительное число $B \geq 2$ такое, что для любого вектора b_i из исходного базиса решетки выполняется неравенство $|b_i|^2 \leq B$. Тогда алгоритм построения редуцированного базиса, описанный выше, требует для своего выполнения $O(n^4 \log B)$ арифметических операций над целыми числами, двоичная длина которых представляет $O(n \log B)$. Таким образом, для построения редуцированного базиса достаточно $O(n^6 \log^3 B)$ бинарных операций.

Доказательство. Перед началом работы алгоритма величины d_i, определенные формулами 7.17 удовлетворяют неравенству $d_i^i \leq B$, что легко следует из упражнения 7.18. Таким образом, перед началом работы алгоритма $D \leq B^{n(n-1)/2}$. Из определения D легко следует, что в случае, когда $L \subset Z^n$, D – неотрицательное целое число, которое не может обратиться в нуль, в силу линейной независимости векторов, составляющих базис решетки. Таким образом, во все время работы алгоритма $D \geq 1$. Выше отмечалось, что при перестановках элементов базиса, выполняемых алгоритмом построения редуцированного базиса, величина D убывает не медленнее геометрической прогрессии со знаменателем 3/4, следовательно, таких перестановок выполняется $O(\log B^{n(n-1)/2}) = O(n^2 \log B)$. Мы оценили, таким образом, сколько раз в головной программе, в цикле "пока" встречается ситуация "то" в условии "если". Из доказательства конечности времени работы алгоритма следует, что ситуация "иначе" встречается также $O(n^2 \log B)$ раз. Итак, тело цикла в основном алгоритме выполняется $O(n^2 \log B)$ раз. Внутренний цикл в ситуации "иначе" дает еще один множитель n для алгоритма 7.3, который выполняется таким образом $O(n^3 \log B)$ раз. Переходим теперь к оценке сложности отдельных предписаний алгоритма 7.1.

В предписании "начальная_установка" два вложенных цикла дают $O(n^2)$ повторений тела цикла, в котором встречаются векторные операции: скалярное произведение векторов, умножение вектора на число, вычитание векторов, что дает еще один множитель n. При этом арифметические операции выполняются над рациональными чиислами. Сложность выполнения операций над этими числами оценим несколько позже.

Количество арифметических операций в алгоритме 7.3, как легко видеть, равно $O(n)$, такое же количество операций в алгоритме 7.4. Сравнивая количество проходов через отдельные ветви алгоритма, получаем, что количество арифметических операций в алгоритме представляется величиной $O(n^4 \log B)$, что и утверждалось в первой части предложения.

Оценим теперь сложность выполнения арифметических операций, выполняемых над рациональными числами в процессе работы алгоритма 7.1. Прежде всего, опишем множество значений, которые могут принимать знаменатели всех встречающихся во время вычислений рациональных чисел. Прежде всего, покажем, что в качестве знаменателей всех встречающихся чисел могут быть использованы только числа d_i, вычисляемые по формулам 7.17, а именно:

$$|b_i^*|^2 = d_i/d_{i-1}, \quad (1 \le i \le n), \qquad (7.21)$$

$$d_{i-1} b_i^* \in L \subset Z^n, \quad (1 \le i \le n), \qquad (7.22)$$

$$d_j \mu_{ij} \in Z, \quad (1 \le j < i \le n). \qquad (7.23)$$

Первое из этих соотношений следует из упражнения 7.18.

Для доказательства второго соотношения выразим векторы b_i^* с неопределенными коэффициентами y_{ij} через исходный базис:

$$b_i^* = b_i - \sum_{j=1}^{i-1} y_{ij} b_j .$$

Неизвестные коэффициенты с фиксированным первым индексом i определяются из системы линейных уравнений

$$(b_i,b_l) = \sum_{j=1}^{i-1} y_{ij}(b_j,b_l) \quad (1 \le l \le i-1). \quad (7.24)$$

Решая систему 7.24 методом Крамера, воспользовавшись формулой 7.17 получаем, что $d_{i-1}\,y_{ij} \in Z$ для всех допустимых значений индексов. Отсюда уже вытекает соотношение 7.22.

Воспользовавшись соотношениями 2.2.5, 7.21 и 7.22 получаем цепочку равенств

$$d_j\mu_{ij} = d_j(b_i,b_j^*)/(b_j^*,b_j^*) = d_{j-1}(b_i,b_j^*) =$$

$$= (b_i,d_{j-1}b_j^*) \in Z,$$

чем заканчивается доказательство соотношения 7.23.

Посмотрим теперь, как меняются величины d_i в процессе работы алгоритма. В начале работы алгоритма они вычисляются по формулам 7.17. В процессе работы эти величины меняются только при перестановке элементов базиса (в алгоритме 7.4). В этом случае d_{k-1} заменяется (в обозначениях 7.4 – 7.8) на

$$d_{k-1}\cdot|c_{k-1}^*|^2/|b_{k-1}^*|^2 = d_{k-2}\cdot|c_{k-1}^*|^2,$$

значения d_i при $i \ne k-1$ не меняются. При этом все значения d_i остаются целыми и во все время работы алгоритма удовлетворяют неравенствам $d_i \le B$. Таким образом, мы оценили множество чисел, которые могут появляться в вычислениях в качестве знаменателей.

Для оценки числителей достаточно найти верхнюю грань для величин $|b_i^*|^2$, $|b_i|^2$ и $|\mu_{ij}|$. Первая из этих величин оценивается просто: в начале работы алгоритма выполняются неравенства $|b_i^*|^2 \le |b_i|^2 \le B$. В процессе работы величина $\max\{|b^*|_i^2 : 1 \le i \le n\}$ не возрастает, для доказательства чего достаточно заметить, что изменение векторов b_i^* происходит только при перестановке элементов базиса, при этом выполняются неравенства $|c_{k-1}^*|^2 < \frac{3}{4}|b_{k-1}^*|^2$ и $|c_k^*|^2 \le |b_{k-1}^*|^2$,

поскольку c_k^* – проекция вектора b_{k-1}^*. Таким образом, во все время работы алгоритма $|b_k^*|^2 \le B$. Оценка величин $|b_i|^2$ и μ_{ij} существенно сложнее. Чтобы получить ее, докажем, что всякий раз в точке проверки условия окончания цикла "пока" выполняются следующие неравенства:

$$|b_i|^2 \le nB \qquad \text{для } i \ne k; \tag{7.25}$$

$$|b_k|^2 \le n^2(4B)^n, \quad \text{если } k \ne n+1; \tag{7.26}$$

$$|\mu_{ij}| \le 1/2 \qquad \text{для } 1 \le j < i < k; \tag{7.27}$$

$$|\mu_{ij}| \le (nB^j)^{1/2} \quad \text{для } 1 \le j < i,\ i > k; \tag{7.28}$$

$$|\mu_{ij}| \le 2^{n-k}(nB^{n-1})^{1/2} \quad \text{для } 1 \le j < k, \quad \text{если } k \ne n+1. \tag{7.29}$$

При доказательстве этих неравенств пользуемся следующим соотношением:

$$\mu_{ij}^2 \le |b_i|^2 / |b_j^*|^2 = d_{j-1}|b_i|^2 / d_j \le \le B^{j-1}|b_i|^2. \tag{7.30}$$

Перед первым выполнением тела цикла неравенства 7.25 и 7.26 следуют из неравенства $|b_i^2| \le B$, которое выполняется по определению B. Вместе с 7.30 это неравенство дает соотношение $|\mu_{ij}| \le B^{j/2}$, откуда следует 7.27 и 7.28, а учитывая, что $B \ge 2$, и 7.29. Таким образом, перед первым выполнением тела цикла неравенства 7.25 – 7.29 справедливы.

Предположим теперь, что неравенства 7.25 – 7.29 выполнены перед началом выполнения тела цикла, и покажем, что эти неравенства будут выполнены и в конце тела цикла, т.е. перед выполнением следующего цикла. Неравенство 7.27 совпадает с 2.2.9, которое выполняется для текущего k все время работы алгоритма 7.1, в том числе и в начале цикла. Выполнение неравенства 7.25 для $i < k$ следует из соотношений 2.2.4, 7.27 и неравенства $|b_i^*|^2 < B$. Покажем, что из 7.25 для $i > k$ и из неравенства 7.29 следуют неравенства 7.26 и 7.28. Доказательство 7.26 сводится к разложению b_k в сумму по формуле 2.2.4, применению неравен-

ства 7.29 и вычислению суммы геометрической прогрессии. Неравенство 7.28 непосредственно вытекает из 7.30 и 7.25.

Итак, покажем, что если перед началом выполнения тела цикла справедливы неравенства 7.25–7.29, то и после его выполнения неравенства 7.25 и 7.29 также имеют место. Отдельно рассмотрим два случая: работа алгоритма осуществляется по ветви "то", т.е. осуществляется перестановка двух векторов; второй случай – алгоритм идет на ветвь "иначе", т.е. осуществляется выполнение условия 2.2.9 для всех $j < k$.

В первом случае множество векторов $\{b_i : i \neq k\}$ не изменилось (мы поменяли местами k–ый вектор с (k–1)–ым и уменьшили после этого k на 1). Во втором случае множество $\{b_i : i > k\}$ векторов, для которых нам нужно доказать неравенство 7.25, заменилось на собственное его подмножество. Этим заканчивается индуктивный шаг для неравенства 7.25.

Переходим к оценке μ_{kj} (если $k \neq n+1$). Снова в теле цикла может выполняться одна из двух серий команд, причем в конце каждой серии значение переменной k меняется, либо увеличиваясь, либо уменьшаясь на 1. Значение k увеличивается, когда алгоритм идет по второй ветви, т.е. достигается выполнение условия 2.2.9 для всех $j < k$ (отдельно, до цикла достигается это условие для $j=k-1$, в цикле – для $j < k-1$). Эти операции никак не влияют на μ_{ij}, если $i > k$, в частности, при $i=k+1$. На следующем шаге значение k увеличивается на 1, и неравенства 7.29 следуют из 7.28, которые выполняются на предыдущем этапе.

Значение k уменьшается на 1, если в теле цикла выполняетется перестановка векторов, при этом после ее выполнения новые значения коэффициентов μ_{kj} совпадают со старыми значениями (с учетом замены k на k–1). Остается только проследить, как изменилось значение коэффициентов μ_{kj}, когда до перестановки векторов мы добивались выполнения условия 2.2.9 для $\mu_{k\ k-1}$. При этом к k–ой строке матрицы μ прибавлялась (k–1)–я строка, умноженная на r, где $|r| < 2|\mu_{k\ k-1}|$. Учиты-

вая, что $|\mu_{k-1\ j}| < 1/2$, получаем

$$|\mu_{kj} - r\ \mu_{k-1\ j}| \le |\mu_{kj}| + |\mu_{k\ k-1}| \le$$

(по индуктивному предположению)

$$\le 2^{n-k+1}(nB^{n-1})^{1/2}. \qquad (7.31)$$

Поскольку новое значение k на 1 меньше старого, получаем требуемую формулу, доказательство неравенств 7.25 – 7.29 закончено.

Для завершения доказательства предложения 7.2Ø нам осталось только оценить значения, получающиеся на промежуточных этапах алгоритма. Заметим, что при выполнении алгоритма 7.3 значения коэффициентов μ не более, чем удваиваются. Поскольку в одном теле основного цикла алгоритм 7.3 выполняется не более, чем k–1 раз, то

$$|\mu_{i\ j}| \le 2^{n-1}(nB^{n-1})^{1/2} \quad \text{для} \quad j < k-1,$$

откуда, воспользовавшись формулами 2.2.4, ортогональностью векторов b_i^*, неравенствами $|b_i^*| < B$ получаем неравенства $|b_i|^2 \le n^2(4B)^n$ для $1 \le i \le n$. Остается перемножить границы для знаменателей и для абсолютных величин, прологарифмировать и выделить главную часть, чтобы получить оценки, фигурирующие в предложении 7.2Ø. □

8. ЗАМЕЧАНИЯ ПО РЕАЛИЗАЦИИ АЛГОРИТМОВ ФАКТОРИЗАЦИИ

Работы по реализации алгоритмов факторизации ведутся на механико-математическом факультете с 1983 года. Первый комплекс программ по факторизации разработан С.Г.Хлебутиным. В качестве языка программирования выбран алгоритмический язык РЕФАЛ. Программы реализованы на ЭВМ типа СМ-4 и эксплуатируются в операционной системе RSX-11M.

Выбор языка РЕФАЛ объясняется следующими соображениями. При реализации систем компьютерной алгебры языки программирования, используемые для численных расчетов (ФОРТРАН и ему подобные), оказываются малопригодными, поскольку символьные вычисления ориентируются на использование сложных структур данных, в первую очередь списков. Вычисления проводятся с целыми и рациональными числами, однако эти числа могут быть весьма велики, что предполагает наличие длинной арифметики. Кроме того, желательно иметь возможность просто реализовывать рекурсивные процедуры. Из языков программирования, имеющихся на ЭВМ СМ-4, этим требованиям ближе всего отвечает алгоритмический язык РЕФАЛ. Язык РЕФАЛ относится к языкам программирования функционального типа, программа на этом языке представляет собой не последовательность операторов, как на ФОРТРАНЕ, АЛГОЛЕ и других операторных языках, а набор функций, каждая из которых описывается одним или несколькими предложениями. Основная используемая структура данных – список, что позволяет программисту не заботиться о технических вопросах размещения информации в памяти ЭВМ. Арифметические операции с целыми числами произвольной длины реализованы в РЕФАЛЕ на уровне языка, т.е. для выполнения этих операций не требуется разрабатывать какие-либо пакеты или пользоваться уже имеющимися. Наконец, само название языка РЕФАЛ, которое представляет собой аббревиатуру от РЕкурсивных функций АЛгоритмический язык, говорит о том, что этот язык рассчитан на программирование рекурсивных процедур.

В описываемом комплексе программ реализован алгоритм факторизации многочленов от одной переменной, основанный на алгоритме Берлекэмпа и

лемме Гензеля. Нахождение делителей исходного многочлена с целыми коэффициентами осуществляется путем перебора различных вариантов произведений многочленов с коэффициентами из колец вычетов. Используется версия алгоритма, описанная в статье Калтофена в книге [6]: старший коэффициент исходного многочлена после завершения алгоритма Берлекэмпа присваивается делителю максимальной степени, этот делитель считается первым в разложении, в результате применения леммы Гензеля старший коэффициент первого делителя остается равным старшему коэффициенту исходного многочлена (по соответствующему модулю), старшие коэффициенты остальных сомножителей приравниваются единице. При переборе первый множитель исключается из рассмотрения.

Кроме алгоритма факторизации, в описываемом комплексе программ реализована полиномиальная арифметика для многочленов от одной или нескольких неизвестных с коэффициентами из кольца целых чисел и из колец вычетов. Реализован алгоритм нахождения НОД многочленов от нескольких переменных и некоторые другие функции.

Ограничения по памяти ЭВМ типа СМ-4 (16-битовое адресное пространство) не позволяет факторизовать с использованием данных программ многочлены высокой степени (выше 20). Хотя РЕФАЛ удобен для написания подовных программ, он далеко не самым оптимальным образом использует ресурсы, прежде всего по памяти. Уже алгоритм Берлекэмпа не позволяет разлагать на множители многочлены степени выше 20-25.

Желание использовать ЭВМ для решения задач факторизации большего размера потребовало использование других языков программирования. В частности, в 1987 году алгоритм Берлекэмпа был реализован А.Ф.Слепухиным на языке программирования СИ, что позволило при тех же ограничениях на память применять его к многочленам степени до 115.

Хотя асимптотические оценки сложности алгоритма 6.2.6, основанного на вычислении комплексного корня многочлена и нахождении минимального вектора решетки, лучше, чем у варианта алгоритма Берлекэмпа-Гензеля, реализация которого описана выше, практическое его использование на малых ЭВМ (типа СМ-4) представляется малоэффективным.

По-видимому, константа S, вычисляемая по формуле 6.2.5, выбирается с существенным запасом и в результате уже для многочленов малой степени требуются вычисления с большими целыми числами, хотя результирующие коэффициенты весьма невелики. Представляет интерес исследование возможности факторизации многочленов с использованием фортрановских типов данных. Отметим, что константу B из алгоритма 6.2.6 можно удалить, заменив строку

. если ||L.базис[0]|| < B

следующей:

. если L.базис[0](x) делит f(x).

В 1986 году группой студентов был проведен следующий численный эксперимент. Предположим, что α – вещественый корень. Описанным выше способом исключим из алгоритма константу B. Точность S определим, исходя из используемых типов данных, т.е. если переменная α имеет тип REAL*4, то $S \approx 25$. Таким образом, остается неопределенным только значение c. Меняем это значение в некоторых пределах. Отметим, что как малые, так и слишком большие значения c не приведут к результату. При малых значениях c норма || || не позволит отделить минимальный многочлен для α, при больших – возникают эффекты, связанные с потерей значащих цифр, и различные переполнения.

Комплекс программ, разработанный на механико-математическом факультете МГУ, позволяет факторизовать таким образом многочлены примерно до 10 степени без использования вычислений с более высокой точностью, чем REAL*4. Для сравнения отметим, что формула 6.2.5 дает в этом случае $S >> 100$, а при вычислении c, описанного в виде REAL*4, по формуле 6.2.4 обычно возникают переполнения.

ЛИТЕРАТУРА

1. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. М.: Мир, 1979.

2. Бахвалов Н.С. и др. Практикум по программированию. М.: Изд-во МГУ, 1986.

3. Ван дер Варден Б.Л. Алгебра. М.: Наука, 1979.

4. Григорьев Д.Ю. Разложение многочленов над конечным полем и решение систем алгебраических уравнений // Зап. научн. семин. ЛОМИ АН СССР. 1984. Т. 137. С. 20-79.

5. Кнут Д.Е. Искусство программирования на ЭВМ. Т. 2. Получисленные алгоритмы. М.: Мир, 1977.

6. Компьютерная алгебра. Символьные и аналитические вычисления. М.: Мир, 1986.

7. Чистов А.Л. Алгоритм полиномиальной сложности для разложения многочленов и нахождение компонент многообразия в субэкспоненциальное время // Зап. научн. семин. ЛОМИ АН СССР. 1984. Т. 137. С. 124-188.

8. Lenstra A.K. Polynomial factorization by root approximation // Lect. Notes Comput. Sci. 1984. V. 174. P. 272-276. 1983 v. 154 p 458-65.

9. Lenstra A.K., Lenstra C.W., Lovasz L. Factoring polynomials with rational coefficients // Math. Ann. 1982. V. 261. P. 515-534.

10. Minotte M. Some inequalities about univariate polynomials // SYMSAC 1981. P. 195-199.

11. Pinkert J.R. An exact method for finding the roots of a complex polinomial // TOMS 1976. V. 2. P. 351-363.

Панкратьев Евгений Васильевич

КОМПЬЮТЕРНАЯ АЛГЕБРА.
ФАКТОРИЗАЦИЯ МНОГОЧЛЕНОВ

Зав. редакцией С.И.Зеленский
Редактор Л.А.Николова
Художественный редактор Н.Ю.Калмыкова
Технический редактор К.С.Чистякова

Н/К
Подписано в печать 22.03.88
Формат 60x90/16 Бумага офс. № 2
Офсетная печать
Усл. печ. л. 5,5 Уч.-изд. л. 4,19
Тираж 500 экз. Заказ №1251 . Заказное. Изд.№ 653
Цена 15 коп.

Ордена "Знак Почета" издательство Масковского университета.
103009, Москва, ул. Герцена, 5/7.
Типография ордена "Знак Почета" изд-ва МГУ.
119899, Москва, Ленинские горы