Записки научных
семинаров ПОМИ
Том 432, 2015 г.

Н. Н. Васильев, О. Канжелева

ПОЛИНОМИАЛЬНАЯ ИНТЕРПОЛЯЦИЯ НАД КОЛЬЦАМИ ВЫЧЕТОВ Z_n

§1. Полиномиальная интерполяция функций над кольцом

Для заданного кольца R и множества значений (x, y), $x \in S \subset R$, $y \in R$, задача полиномиальной интерполяции заключается в построении такого полинома $P(x)$, что $P(x) = y$ для любого $x \in S$, если он существует. Полином, реализующий заданные значения, будем называть *интерполяционным*, а процесс его построения – *интерполяцией*.

Методы интерполяции Лагранжа и Ньютона не работают в случае интерполяции над кольцами, так как обратный элемент определен не для всякого элемента кольца.

П. Гопалан в работе [1] предложил алгоритм для полиномиальной интерполяции, основывающийся на следующих соображениях.

(1) Каждый полином может быть представлен в виде линейной комбинации базисных полиномов

$$N_0 = 1, \quad N_i(x) = \prod_{j=0}^{i-1}(x - j), \ i > 0. \tag{1}$$

(2) Для каждого кольца существует множество, называемое интерполяционным, значения полиномиальной функции на котором определяют все значения данной функции.

Стоит отметить, что, зная разложение $m = \prod\limits_{i=0}^{t} p_i^{k_i}$ составного числа m на простые множители и интерполяционные полиномы над $Z_{p_i^{k_i}}$, интерполяционный полином над Z_m можно построить с помощью *китайской теоремы об остатках*.

Поэтому дальнейшие рассуждения будут касаться колец Z_{p^k}.

Ключевые слова: кольцо вычетов, нуль-полиномы, многочлены Фробениуса, пермутационные полиномы над кольцом.

Работа первого автора поддержана грантом РНФ 14-11-00581.

1.1. Интерполяционное множество.

Определение 1.1. Для заданного кольца R и множества точек $I \subset R$ подмножество $S \subset I$ называется *интерполяционным множеством* для I, если значения любого полинома на множестве I определяются его значениями на множестве S.

Для того чтобы найти множество S для заданного кольца и множества точек I, необходимо найти минимальное множество, задающее полином с нулевыми значениями на I. Таким образом, S должно удовлетворять следующему условию [1]:

$$\text{Если } P(\alpha) \equiv 0 \quad \forall \alpha \in S, \quad \text{то } P(\beta) \equiv 0 \quad \forall \beta \in I.$$

Данную идею реализует следующий жадный алгоритм.

Input: Множество $I \in Z_{p^a}$.
Output: Интерполяционное множество S для I.
Выбираем произвольный элемент $\alpha_0 \in I$. Полагаем $S = \{\alpha_0\}, i = 1$.
Полагаем $N_i^S(X) = \prod\limits_{j<i}(X - \alpha_j)$.

В цикле
если $N_i^S(x) \equiv 0$ для всех $x \in I$,
Output $S = \{\alpha_0, \ldots, \alpha_{i-1}\}$. Выход.
Иначе
Находим $x \in I$, минимизирующий $valp(N_i^S(x))$.
Полагаем $\alpha_i = x, i = i + 1$.

Сложность данного алгоритма есть $O(|S|^2 \cdot |I|)$.

1.2. Алгоритм интерполяции. Для найденного интерполяционного множества S определим базисные полиномы следующим образом:

$$N_i^S(x) = \prod_{j<i}(x - \alpha_j), \ x \in S. \tag{2}$$

Для любого $i > m$, где $m = |S|$, базисные полиномы $N_i^S(x)$ тождественно равны нулю в кольце Z_{p^k}.

Тогда любой полином в Z_{p^k} можно представить в виде линейной комбинации

$$f(x) = \sum_{i=0}^{m} c_i N_i^S(x). \tag{3}$$

Таким образом, зная интерполяционное множество для Z_{p^k}, интерполяционный полином можно найти за k шагов.

§2. Связь с нуль-полиномами

Определение 2.1. Для заданного кольца R полином $P(x) \in R[x]$ называется *нуль-полиномом*, если $P(x) \equiv 0$ для любого $x \in R$.

В кольце Z_m нуль-полином минимальной степени может быть построен следующим образом:

$$\frac{m}{p}(x^p - x), \tag{4}$$

где p – минимальный простой делитель числа m.

Множество всех нуль-полиномов есть идеал. Но идеал нуль-полиномов является главным только в случае простых конечных полей, когда он порожден многочленом Фробениуса. Выяснение строения составляющих базиса Грёбнера этого идеала в случае колец вычетов по составному модулю является более сложной задачей. Заметим, что, поскольку речь идёт о полиномах одной переменной, базис Грёбнера определён канонически только самим идеалом, так как в этом случае имеется только одно естественное упорядочение мономов.

Для некоторых значений p и k в [2] получены значения $S(p^k)$ минимальной степени нормированных нуль-полиномов, имеющих единичный старший коэффициент:

$$S(p^k) = \begin{cases} p(k - \lfloor \frac{k}{p+1} \rfloor), & 1 \leqslant k \leqslant p(p+1), \\ p(k-1), & p+1 \leqslant k \leqslant 2p+1, \\ 2p^2, & k = 2p+2, \\ p^3, & k = p(p+1)+1, \\ p(p+1), & k = p+2, \\ p^n, & k = I_p(n). \end{cases} \tag{5}$$

Здесь числа $I_p(n)$ определяются рекуррентным соотношением

$$I_p(n) = \begin{cases} 0, & n = 0, \\ pI_p(n-1)+1, & n > 0. \end{cases} \tag{6}$$

Легко видеть, что минимальная степень нормированного нуль-полинома связана с размером интерполяционного множества. Действительно, размер интерполяционного множества, как видно из формулы (3), определяет максимальную степень полинома. А любой полином большей степени может быть редуцирован с помощью нуль-полиномов.

Таким образом, оценки, полученные в [2] для минимальной степени нуль-полинома, можно также рассматривать как оценки размера интерполяционного множества.

§3. Подсчет количества пермутационных полиномиальных функций над кольцом

Определение 3.1. Полином $f(x) \in R[x]$ называется *пермутационным*, если он задает биективную функцию в кольце R.

Подсчет пермутационных полиномиальных функций является нетривиальной задачей, так как разные полиномы могут задавать одну и ту же перестановку.

3.1. Случай кольца Z_{p^k}. Р. Ривест в [3] сформулировал следующий критерий для определения пермутационности полиномов над кольцами Z_{2^k}.

Теорема 1. *Пусть $P(x) = a_0 + a_1x + \dots + a_nx^n$ – полином с целыми коэффициентами. Тогда $P(x)$ является пермутационным полиномом в кольце $Z_{2^k}, k \geqslant 2$, тогда и только тогда, когда a_1 нечетно, сумма $a_2 + a_4 + a_6 + \dots$ четна и сумма $a_3 + a_5 + \dots$ четна.*

Данный критерий позволяет определить, является ли полином пермутационным, без вычисления значений функции, но неудобен для подсчета количества пермутационных полиномиальных функций.

Будем выводить рекуррентную формулу для количества пермутационных полиномиальных функций, основываясь на следующих фактах.

- Поскольку для любого пермутационного полинома $P(x)$ полином $P(x) + \mathrm{const}$ также является пермутационным, будем рассматривать полиномы, сохраняющие 0. А следовательно, и перестановки, сохраняющие 0.
- Каждая перестановка однозначно определяется первыми m своими элементами, где m – размер интерполяционного множества для Z_{2^k}.

3.1.1. *Случай кольца Z_{p^2}.* Рассмотрим случай перехода от кольца Z_p к кольцу Z_{p^2}, так как в кольце Z_p интерполяционное множество совпадает с множеством всех значений кольца.

Пусть перестановка $\{\alpha_0, \ldots, \alpha_{p-1}\}$ переходит в $\{\beta_0, \ldots, \beta_{p^2-1}\}$. Поскольку наши перестановки оставляют 0 на месте, $\alpha_0 = \beta_0 = 0$.

Для Z_{p^2} интерполяционное множество состоит из $2p$ элементов. Именно эти элементы задают полиномиальную перестановку, поэтому необходимо рассмотреть лишь поведение первых $2p$ элементов перестановки $\{\beta_0, \ldots, \beta_{p^2-1}\}$.

Элемент α_1 может перейти в одно из следующих мест: β_1, β_{p+1}, $\ldots$, $\beta_{p(p-1)+1}$; элемент α_2 может перейти в одно из следующих мест: β_2, β_{p+2}, $\ldots$, $\beta_{p(p-1)+2}$; $\ldots$; элемент α_{p-1} может перейти в одно из следующих мест: β_{2p-1}, β_{3p-1}, $\ldots$, β_{p^2-1}. Таким образом, для $\beta_1, \ldots, \beta_{p-1}$ существует p возможных значений.

Тогда для каждого из элементов $\beta_p, \ldots, \beta_{2p-1}$ возможны $p-1$ различных значений.

Следовательно, для Z_{p^2} полиномиальных перестановок имеется

$$N(p^2) = \frac{p!}{p} p^2 (p-1)^p (p)^{p-1} = p!(p-1)^p p^p. \tag{7}$$

3.1.2. *Случай кольца* $Z_{p^k}, k > 2$. Пусть известно $N(p^{k-1})$ – количество пермутационных полиномиальных функций в $Z_{p^{k-1}}$.

Для Z_{p^k} размер интерполяционного множества будет меньше p^{k-1}. Поэтому необходимо понять, каким образом элементы $\alpha_0, \ldots$, $\alpha_{S(p^k)-1}$, где $S(p^k)$ – размер интерполяционного множества для Z_{p^k}, формируют элементы $\beta_0, ..., \beta_{S(p^k)-1}$ для перестановок $\{\alpha_0, ..., \alpha_{p^{k-1}-1}\}$ и $\{\beta_0, \ldots, \beta_{p^k-1}\}$ соответственно.

Имеем $\alpha_0 = \beta_0 = 0$. Для каждого из вычетов $\beta_1, \ldots, \beta_{S(p^k)-1}$ возможно одно из p значений, удовлетворяющих сравнению $\beta_i \equiv \alpha_i + jp^{k-1} \mod p^k$.

Таким образом,

$$N(p^k) = \frac{N(p^{k-1})}{p^{k-1}} p^k p^{S(p^k)-1} = N(p^{k-1}) p^{S(p^k)}. \tag{8}$$

Тогда общая формула для произвольного k будет выглядеть следующим образом:

$$N(p^k) = \begin{cases} p!, & k = 1, \\ p!(p-1)^p p^p, & k = 2, \\ N(p^{k-1}) p^{S(p^k)}, & k > 2. \end{cases} \tag{9}$$

§4. Численные эксперименты

Опираясь на идеи работы [1] , мы реализовали следующие алгоритмы над кольцами Z_{p^k}:

- вычисление базиса Гребнера идеала нуль-полиномов;
- быстрое вычисление интерполяционного множества;
- подсчет количества пермутационных полиномиальных функций.

4.1. Построение базисов нуль-полиномов. В [2] были получены значения минимальной степени нормированных нуль-полиномов для некоторых колец. Нами для некоторых колец вычетов были построены базисы Гребнера идеалов нуль-полиномов. Так как мы работаем в кольце полиномов от одной переменной, эти базисы зависят только от самого идеала и не используют мономиальных упорядочений. Поэтому они определены канонически и являются естественным обобщением многочленов Фробениуса на случай кольца вычетов. Построение базиса происходит следующим образом: перебираются все нуль-полиномы, для каждого следующего найденного нуль-полинома проверяется, может ли он быть редуцирован до 0 с помощью найденных ранее базисных нуль-полиномов. Если нет, то он добавляется в множество базисных полиномов.

Для ускорения работы данного алгоритма были использованы следующие соображения.

- Если $P(x) \equiv 0$ на интерполяционном множестве, то $P(x) \equiv 0$ на всем множестве значений кольца.
- Если нуль-полиномы $P(x)$ и $Q(x)$ удовлетворяют соотношению $LT(P(x)) = LT(Q(x))$, то только один из них может быть базисным, так как их разность является нуль-полиномом меньшей степени и этот полином, или полиномы его задающие, уже должен присутствовать в базисе. Поэтому если найден нуль-полином, то нет смысла рассматривать нуль-полином с тем же самым главным членом.

Для вычислений использовалась система Singular, поддерживающая вычисления базиса Грёбнера над кольцами.

Некоторые найденные базисы:

$$Z_{16}: \qquad 8x^2 + 8x,$$
$$2x^4 + 12x^3 + 14x^2 + 4x,$$

$$x^6 + 15x^5 + 15x^4 + 13x^3 + 8x^2 + 12x;$$

Z_{32} :

$$16x^2 + 16x,$$
$$4x^4 + 24x^3 + 28x^2 + 8x,$$
$$2x^6 + 30x^5 + 30x^4 + 26x^3 + 8x,$$
$$x^8 + 2x^5 + 3x^4 + 6x^3 + 12x^2 + 8x;$$

Z_{64} :

$$32x^2 + 32x,$$
$$8x^4 + 16x^3 + 24x^2 + 16x,$$
$$4x^6 + 4x^5 + 4x^4 + 28x^3 + 24x^2,$$
$$x^8 + 2x^6 + x^4 + 28x^2 + 32x.$$

Ниже представлены теоретические значения степени минимального нормированного нуль-полинома для некоторых колец, а также результаты наших численных экспериментов:

степень основания кольца Z_{p^k}	минимальная степень нормированного нуль-полинома, полученная в [2]
$k < p+1$	pk
$k = p+1$	p^2
$p+2 <= k <= 2p+1$	$p(k-1)$
$k = p(p+1)+1$	p^3

степень основания кольца Z_{2^k}	экспериментально полученная степень минимального нормированного нуль-полинома
$k = 2$	4
$k = 3$	4
$k = 4$	6
$k = 5$	8
$k = 6$	8

4.2. Быстрое вычисление интерполяционного множества для кольца Z_n. Для кольца Z_n и множества $S = \{0, \ldots, n-1\}$ интерполяционное множество I может быть построено за линейное время от I.

Согласно идее Гопалана, необходимо найти такое множество $\{0, \ldots, m-1\}$, что

$$N_i(x) \equiv 0 \; \forall x = \overline{0, n} \; \forall i = \overline{0, m}, \text{ где } N_i(x) \text{ определено в (1).}$$

Заметим, что

$$\text{для любого } x \text{ и } i > k \text{ или } x = \overline{0, i-1}, \;\; N_i(x) \equiv 0,$$
$$N_i(i) = i!,$$
$$N_i(i+a) = \tfrac{(i+a)!}{a!}.$$

Таким образом, если $N_i(i) \equiv 0$, то $N_i(i+a) \equiv 0$ для любого a.

Это значит, что минимальное m, такое, что $m! \equiv 0 \bmod (n)$, ограничивает интерполяционное множество.

4.3. Подсчет количества пермутационных полиномиальных функций в Z_{p^k}. В поле Z_p имеется $p!$ перестановок, и все они полиномиальные. В кольце Z_{p^k} не все перестановки полиномиальные. Кроме того, одна и та же перестановка может быть реализована разными полиномами. Стоит также отметить, что пермутационные полиномиальные функции над кольцом образуют группу.

Для подсчета количества перестановок над кольцом мы перечисляем все перестановки и проверяем их на полиномиальность, используя интерполяционное множество.

Алгоритм подсчета основывается на следующих идеях.

- Поскольку любой полином задается с помощью интерполяционного множества, любая полиномиальная перестановка задается своим подмножеством, размер которого равен размеру интерполяционного множества. Поэтому нет необходимости перечислять все перестановки, необходимо рассмотреть лишь все перестановки m-мерных подмножеств множества $\{0, \ldots, p_k - 1\}$, где m – размер интерполяционного множества.
- Поскольку для любого пермутационного полинома $P(x)$ полином $P(x) + \mathrm{const}$ также является пермутационным, можно рассматривать лишь перестановки, сохраняющие 0. В таком случае рассматривать необходимо лишь перестановки $(m-1)$-мерных подмножеств $(p^k - 1)$-мерного множества.

Для случая Z_{2^k} при рассмотрении перестановок, сохраняющих 0, можно дополнительно использовать условие:

- перестановка является полиномиальной, если на четных позициях (считая 0) у нее стоят четные значения, а на нечетных позициях – нечетные.

Данное условие получается из следующих соотношений:

$$P(x+2^{k-1}) \equiv P(x) + 2^{k-1} \mod 2^k,$$
$$P(x+2^{k-2}) \equiv P(x) + 2^{k-2} \mod 2^{k-1},$$
$$\dots$$
$$P(x+2) \equiv P(x) + 2 \mod 4.$$

Результаты численного эксперимента подтверждают эту формулу:

основание кольца	количество пермутационных полиномиальных функций
4	$8 = 2^3$
8	$128 = 2^7$
16	$8192 = 2^1 3$
32	$2097152 = 2^{21}$
64	$536870912 = 2^{29}$
9	$1296 = 3!2^3 3^3$
27	$25509168 = 3!2^3 3^3 3^9$
25	$384000000 = 5!4^5 5^5$

§5. Использованные технологии

Представленные алгоритмы были реализованы на языке Java с использованием библиотек Concurrent. Для расчетов количества пермутационных полиномиальных функций использовался 12-ядерный узел на кластере delta-force.cluster.spbstu.ru. Запуск задач осуществлялся через Sun Grid Engine с помощью утилиты qsub. Представленный алгоритм перечисления пермутационных полиномиальных функций был реализован с помощью языка C и библиотеки OpenMPI.

Стоит отметить, что задача перечисления пермутационных полиномиальных функций является удобной для распараллеливания. Ускорение при использовании нескольких ядер близко к оптимальному.

Литература

1. P. Gopalan, *Query-efficient algorithms for polynomial interpolation over composites*. — SIAM J. Comput. **38**, No. 3 (2008), 1033–1057.
2. S. Li, *Null polynomials modulo m*; `arXiv:math/0510217v2`.
3. R. Rivest, *Permutation polynomials modulo* 2^w. — Finite Fields Appl. **7** (2001), 287–292.

Vasiliev N. N., Kanzheleva O. Polynomial interpolation over the residue rings Z_n.

We consider the problem of polynomial interpolation over the residue rings Z_n. The general case can be easily reduced to the case of $n = p^k$ due to the Chinese reminder theorem. In contrast to the interpolation problem over fields, the case of rings is much more complicated due to the existence of nonzero polynomials representing the zero function. Also, the result of interpolation is not unique in the general case. We compute, in the frame of the CAS system Singular, the Gröbner bases of ideals of null-polynomials over residue rings. This allows us to obtain a canonical form for the results of interpolation. We also describe a connection between estimates of the cardinality of interpolating sets and of the total number of permutation polynomials over the ring. As a consequence, we give a recurrence formula for the number of permutation polynomials over Z_p^k .

С.-Петербургское отделение
Математического института
им. В. А. Стеклова РАН,
наб. р. Фонтанки, д. 27,
С.-Петербург 191023, Россия
E-mail: `nn.vasiliev@gmail.com`

Поступило 4 ноября 2014 г.

С.-Петербургский
государственный политехнический
университет, С.-Петербург, Россия;
Google Corporation, Irvine, USA
E-mail: `olga.kanzheleva@gmail.com`