Конечные поля

М.Д. Малых, РУДН

11 октября 2022 г.

Содержание

1. Конечные кольца

Как мы уже знаем, всякий идеал кольца $\mathbb{Z}$ является главным. Всякое натуральное число q порождает главный идеал $q\mathbb{Z}$ кольца $\mathbb{Z}$, образованный всеми числами, которые делятся на q. Факторкольцо $\mathbb{Z}/q\mathbb{Z}$ состоит из q элементов

$$[0], [1], \ldots, [q-1],$$

причем элемент $[n]$ факторкольца — множество всех целых чисел, которые представимы в виде

$$n + qm, \quad m \in \mathbb{Z}.$$

По этой причине часто его обозначают так:

$$[n] = n + q\mathbb{Z}.$$

В модулярной арифметике, созданной Гауссом, это кольцо описывают без отсылки к идеалам. В качестве $\mathbb{Z}/q\mathbb{Z}$ предлагают рассмотреть множество чисел $0, 1, \ldots q-1$, действия с которыми заданы по правилам: суммой будет остаток от деления на q их суммы как целых чисел, а произведением — остаток от деления на q их произведения как целых чисел. Кратко указание на то, что от выражения нужно перейти к его остатку от деления на q записывают при помощи символа %:

```
sage: 2*3 % 5
1
```

Читают: 2 на 3 по модулю 5.

Удивительным фактом являются то, что такие конечные множества с так введенными действиями являются кольцами. В прошлой главе мы убедились в этом, интерпретировав $\mathbb{Z}/q\mathbb{Z}$ как факторкольцо по идеалу $q\mathbb{Z}$.

2. Простые числа

Определение 1. Натуральное число, которое больше 1 и которое нельзя представить в виде произведения двух меньших натуральных чисел, называют простым (prime).

Теорема 1. Существует бесконечно много простых чисел.

Доказательство. Допустим противное. Пусть p — наибольшее простое число, тогда натуральное $p! + 1$ не является простым и делится нацело на некоторое простое число $q > 1$, то есть $p! + 1 \in q\mathbb{Z}$. Число q — простое, поэтому оно не превосходит p и, следовательно, присутствует среди сомножителей $p!$. Но тогда $1 \in q\mathbb{Z}$ и поэтому $q\mathbb{Z} = \mathbb{Z}$. Это означает, что $q = 1$, что невозможно. □

Algorithm 1 Алгоритм отыскания списка всех простых чисел, меньших P

```
def prime_list(P):
    N=[n for n in range(2,P)]
    for i in range(2,floor(sqrt(P))+1):
        N=[n for n in N if n % i !=0 or n==i]
    return N
```

Мы можем весьма просто, но и весьма медленно найти все положительные простые числа, которые меньше заданного числа P. Для этого используется алгоритм, известный как решето Эратосфена. Всякое составное число, которое меньше P, является произведением двух натуральных чисел, из которых одно — меньше $\sqrt{P}$. Поэтому достаточно создать список всех натуральных чисел от $2, \ldots, N$, и выкинуть из него последовательно все числа, которые делятся на i, значения которого заключены между 2 и $\sqrt{P}$.

В Sage сказанное можно реализовать в виде функции `prime_list`, см. алгоритм 1. Напр., выпишем все простые числа до 100:

```
sage: prime_list(10^2)
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43,
    47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97]
```

Этот алгоритм является очень затратным.

В Sage имеется встроенная функция `is_prime`, которая позволяет выяснить, является ли число простым:

```
sage: ZZ(19).is_prime()
True
sage: 19.is_prime()
True
```

При ее помощи нетрудно составить список всех простых чисел, меньших некоторого заданного числа. Вот список всех простых чисел, меньших 100, найденный при ее помощи:

```
sage: [n for n in range(1,101) if ZZ(n).is_prime()]
```

```
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43,
    47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97]
```

3. Конечные поля

Теорема 2. Если q — не является простым числом, то факторкольцо $\mathbb{Z}/q\mathbb{Z}$ имеет делители нуля.

Доказательство. Поскольку q не является простым, то его можно представить в виде произведения двух натуральных чисел a и b, заключенных между 2 и $q-1$. Но тогда $a+q\mathbb{Z}$ и $b+q\mathbb{Z}$ отличаются от $0+q\mathbb{Z}$, то есть нуля факторкольца, и в то же время

$$(a+q\mathbb{Z})\cdot(b+q\mathbb{Z}) = ab+q\mathbb{Z} = q\mathbb{Z}$$

то есть $a+q\mathbb{Z}$ и $b+q\mathbb{Z}$ являются делителями нуля в факторкольце. □

Теорема 3. Если p — простое число, то факторкольцо $\mathbb{Z}/p\mathbb{Z}$ является полем.

Доказательство. Пусть $a+p\mathbb{Z}$ — произвольный его элемент, отличный от нуля. Нам нужно предъявить такой элемент этого кольца, скажем, $x+p\mathbb{Z}$, что

$$(a+p\mathbb{Z})\cdot(x+p\mathbb{Z}) = 1+p\mathbb{Z}.$$

или

$$ax+p\mathbb{Z} = 1+p\mathbb{Z}$$

Поскольку всякий идеал кольца $\mathbb{Z}$ является главным, идеал (a,p) можно записать как идеал $b\mathbb{Z}$. Запись $a,p\in b\mathbb{Z}$ означает, что a и p делятся на b. Поскольку p — простое, или $b=p$, или $b=1$. Во первом случае, a делится на p и $a+p\mathbb{Z}$ — нуль факторкольца, что невозможно. Во втором случае $(a,p)=\mathbb{Z}$ и в частности $1\in(a,p)$. Это означает, что существуют такие целые числа u и v, что $1=ua+vp$. Но тогда

$$au+p\mathbb{Z} = 1+p\mathbb{Z}$$

то есть $u + p\mathbb{Z}$ — элемент, обратный к $a + p\mathbb{Z}$. □

Напр.,

```
sage: ZZ.quo(2*ZZ).is_field()
True
sage: ZZ.quo(3*ZZ).is_field()
True
sage: ZZ.quo(4*ZZ).is_field()
False
sage: ZZ.quo(5*ZZ).is_field()
True
```

Определение 2. Конечное поле, или поле Галуа (Galois field), — поле, состоящее из конечного числа элементов; это число называется порядком поля.

Если p — простое число, то факторкольцо $\mathbb{Z}/p\mathbb{Z}$ является полем и состоит из p элементов. Оно кратко обозначается как $\mathrm{GF}(p)$, хотя Sage различает эти конструкции.

```
sage: GF(2)
Finite Field of size 2
sage: ZZ.quo(2*ZZ)
Ring of integers modulo 2
sage: ZZ.quo(2*ZZ) == GF(2)
False
```

Пример 1. Поле $\mathrm{GF}(2)$ содержит 2 элемента, 0 и 1. Традиционно эти элементы интерпретируют как Истину и Ложь. При этом:

$$0 + 0 = 0, \quad 0 + 1 = 1, \quad 1 + 1 = 0,$$

то есть сложение эквивалентно логической операции «исключающее или», а

$$0 \cdot 0 = 0, \quad 0 \cdot 1 = 0, \quad 1 \cdot 1 = 1,$$

то есть умножение эквивалентно логической операции «и».

Пример 2. Поле GF(3) содержит 3 элемента, 0, 1 и 2. Действия определены в соответствии с таблицами сложения и умножения:

+	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

·	0	1	2
0	0	0	0
1	0	1	2
2	0	2	1

4. Уравнения над конечными полями

Линейное уравнение

$$ax = b, \quad a \neq 0,$$

всегда имеет корень в конечном поле и притом только один. Его можно найти перебором.

Пример 3. Найдем корень уравнения

$$2x = 1$$

в воле GF(3). Это поле содержит 3 элемента, 0, 1 и 2, которые после умножения на 2 превращаются в 0, 2 и 1. Поэтому корнем служит $x = 1$.

Пример 4. Найдем корень уравнения

$$2x = 1$$

в воле GF(97). Это поле содержит 97 элементов.

```
sage: p=97
sage: a=2
sage: [x for x in range(1,p) if ZZ(x*a) % p == 1]
[49]
```

Разумеется, в Sage решение можно найти, поделив 1 на 2, указав, что эти числа рассматриваются как элементы поля Галуа GF(97):

```
sage: F=GF(97)
sage: F(1)/F(2)
49
```

Развитую выше теорию для систем линейных уравнений можно применять не только над $\mathbb{Z}$, но и над конечными полями. При этом алгоритм Гаусса сохраняет свою силу, а треугольная система наибольшего ранга всегда имеет и притом единственное решение в поле Галуа.

Пример 5. Рассмотрим систему с тремя неизвестными:

```
sage: var("x,y,z")
(x, y, z)
sage: S=[5*x-2*y+z+1,3*x+y-z+1, x+y+z+5]
```

Уравнение системы можно рассматривать и как элементы $\mathbb{Z}[x, y, z]$, и как элементы $\mathrm{GF}(p)[x, y, z]$. В первом случае мы будем искать решения в поле частных $\mathbb{Q}$, во втором — в поле $\mathrm{GF}(p)$. Сравним результаты.

В кольце $\mathbb{Z}[x, y, z]$ имеем:

```
sage: K=QQ[x,y,z]
sage: triangulation([K(s) for s in S])
[5*x - 2*y + z + 1, 11*y - 8*z + 2, 100*z + 250]
sage: tsolve(triangulation([K(s) for s in S]))
{z: -5/2, y: -2, x: -1/2}
```

Проверка:

```
sage: ans=tsolve(triangulation([K(s) for s in S]))
sage: [K(s).subs(ans) for s in  S]
[0, 0, 0]
```

В кольце $\mathrm{GF}(2)[x, y, z]$ имеем:

```
sage: K=GF(2)[x,y,z]
sage: triangulation([K(s) for s in S])
```

```
[x + z + 1, y]
```

Неизвестной z можно придать любое из двух значение, при этом в любом случае $y = 0$, а $x = -1 - z = 1 - z$ по $\mod 2$. Поэтому мы имеем два решения:

$$x = 1,\ y = z = 0, \quad x = y = 0,\ z = 1$$

Проверка:

```
sage: [K(s)(1,0,0) for s in  S]
[0, 0, 0]
sage: [K(s)(0,0,1) for s in  S]
[0, 0, 0]
```

В кольце $\mathrm{GF}(3)[x, y, z]$ имеем:

```
sage: K=GF(3)[x,y,z]
sage: triangulation([K(s) for s in S])
[-x + y + z + 1, y - z + 1, -z - 1]
sage: tsolve(triangulation([K(s) for s in S]))
{z: -1, y: 1, x: 1}
```

В этом случае система имеет единственное решение. Проверка:

```
sage: ans=tsolve(triangulation([K(s) for s in S]))
sage: [K(s).subs(ans) for s in  S]
[0, 0, 0]
```

В кольце $\mathrm{GF}(5)[x, y, z]$ имеем:

```
sage: K=GF(5)[x,y,z]
sage: triangulation([K(s) for s in S])
[-2*x + y - z + 1, -2*y + z + 1]
```

Решений опять несколько, неизвестную z мы можем брать любой.

```
sage: tsolve(triangulation([K(s) for s in S])+[K(z)
```

```
    ])
{z: 0, y: -2, x: 2}
sage: tsolve(triangulation([K(s) for s in S])+[K(z
    -1)])
{z: 1, y: 1, x: -2}
sage: tsolve(triangulation([K(s) for s in S])+[K(z
    -2)])
{z: 2, y: -1, x: -1}
```

Таким образом, получается 3 решения. Проверка:

```
sage: ans=tsolve(triangulation([K(s) for s in S])+[K
    (z)])
sage: [K(s).subs(ans) for s in  S]
[0, 0, 0]
sage: ans=tsolve(triangulation([K(s) for s in S])+[K
    (z-1)])
sage: [K(s).subs(ans) for s in  S]
[0, 0, 0]
sage: ans=tsolve(triangulation([K(s) for s in S])+[K
    (z-2)])
sage: [K(s).subs(ans) for s in  S]
[0, 0, 0]
```

В кольце $\mathrm{GF}(7)[x, y, z]$ имеем:

```
sage: K=GF(7)[x,y,z]
sage: triangulation([K(s) for s in S])
[-2*x - 2*y + z + 1, -3*y - z + 2, -3*z + 3]
sage: tsolve(triangulation([K(s) for s in S]))
{z: 1, y: -2, x: 3}
```

В этом случае система имеет единственное решение. Проверка:

```
sage: ans=tsolve(triangulation([K(s) for s in S]))
sage: [K(s).subs(ans) for s in  S]
[0, 0, 0]
```

В кольце $\mathrm{GF}(11)[x, y, z]$ имеем:

```
sage: K=GF(11)[x,y,z]
sage: triangulation([K(s) for s in S])
[5*x - 2*y + z + 1, -4*y + 4*z + 2, 3*z + 2]
sage: tsolve(triangulation([K(s) for s in S]))
{z: 3, y: -2, x: 5}
```

В этом случае система имеет единственное решение. Проверка:

```
sage: ans=tsolve(triangulation([K(s) for s in S]))
sage: [K(s).subs(ans) for s in  S]
[0, 0, 0]
```

Пусть система S уравнений над кольцом $\mathbb{Z}$ имеет и притом единственное решение в его поле частных $\mathbb{Q}$, скажем,

$$x_1 = a_1/b_1, \ldots, x_n = a_n/b_n, \quad a_1, \ldots, b_n \in \mathbb{Z}.$$

Пусть числа $b_1, \ldots, b_n$ как элементы поля $\mathrm{GF}(p)$ не равны нулю, тогда одно из решений этой системы в $\mathrm{GF}(p)$ дается формулой

$$x_1 = [a_1]/[b_1], \ldots, x_n = [a_n]/[b_n].$$

Пример 6. Обратимся вновь к системе из примера 5 из $\mathbb{Q}$. При $p = 3$ имеем:

```
sage: p=3
sage: GF(p)(-1)/GF(p)(2)
1
sage: GF(p)(-2)
1
```

```
sage: GF(p)(-5)/GF(p)(2)
2
sage: K=GF(p)[x,y,z]
sage: tsolve(triangulation([K(s) for s in S]))
{z: -1, y: 1, x: 1}
```

При $p = 5$ имеем:

```
sage: p=5
sage: GF(p)(-1)/GF(p)(2)
2
sage: GF(p)(-2)
3
sage: GF(p)(-5)/GF(p)(2)
0
sage: K=GF(p)[x,y,z]
sage: tsolve(triangulation([K(s) for s in S]))
{y: -2*z - 2, x: z + 2}
```

Видно, что получается одно из решений.

5. Протокол Диффи–Хеллмана

Конечные поля лежат в основе современной криптографии. Опишем кратко первоначальную версию протокола Диффи–Хеллмана обмена криптографическими ключами[1].

В открытом доступе находятся довольно большое простое число p и еще некоторое число g. При этом список степеней g как элементов $\mathrm{GF}(p)$ выглядит как случайный набор чисел:

```
sage: p=97
```

[1] Фергюсон, Нильс, Шнайер, Брюс. Практическая криптография. : Пер. с англ. — М. : Издательский дом «Вильямс», 2004. Гл. 12.

```
sage: g=5                                                                    113
sage: [GF(p)(g)^i for i in range(p)]                                         114
[1, 5, 25, 28, 43, 21, 8, 40, 6, 30, 53, 71, 64, 29,                        115
   48, 46, 36, 83, 27, 38, 93, 77, 94, 82, 22, 13,
  65, 34, 73, 74, 79, 7, 35, 78, 2, 10, 50, 56, 86,
  42, 16, 80, 12, 60, 9, 45, 31, 58, 96, 92, 72, 69,
   54, 76, 89, 57, 91, 67, 44, 26, 33, 68, 49, 51,
  61, 14, 70, 59, 4, 20, 3, 15, 75, 84, 32, 63, 24,
  23, 18, 90, 62, 19, 95, 87, 47, 41, 11, 55, 81,
  17, 85, 37, 88, 52, 66, 39, 1]
```

Если взять p достаточно большим, то можно подобрать g так, чтобы было невозможно программным путем перебрать все элементы этого списка.

Имеются два пользователя А и Б. Пользователь А случайным образом выбирает натуральное число x между 1 и $p-1$, и отправляет пользователю Б число g^x как элемент $\mathrm{GF}(p)$, то есть остаток r_1 от деления g^x по модулю p. Это его публичный ключ. Пользователь Б выбирает случайным образом число y посылает пользователю А g^y как элемент $\mathrm{GF}(p)$, то есть остаток r_2 от деления g^y по модулю p. Это его публичный ключ. В результате оба пользователя знают число g^{xy} как элемент $\mathrm{GF}(p)$. Пользователь А вычисляет его как r_2^x, поскольку $[g^{xy}] = [g^y]^x$, а пользователь Б — как r_1^y. Сторонний наблюдатель, подслушав r_1 и r_2, не может найти g^{xy} как элемент $\mathrm{GF}(p)$.

Пример 7. Пусть в общем доступе имеются константы:

```
sage: p=97                                                                   116
sage: g=5                                                                    117
```

Пусть пользователи выбрали

```
sage: x=13                                                                   118
sage: y=49                                                                   119
```

Разумеется, никому, даже друг другу, они об этом выборе не сказали. Пользователь А передает

```
sage: r1=GF(p)(g^x)
sage: r1
29
```

а пользователь Б передает

```
sage: r2=GF(p)(g^y)
sage: r2
92
```

Тогда число

```
sage: GF(p)(g^(x*y))
68
```

пользователь А может найти как

```
sage: GF(p)(r2^x)
68
```

а пользователь Б — как

```
sage: GF(p)(r1^y)
68
```

Сторонний наблюдатель может найти это число, если перебором подберет x:

```
sage: [i for i in range(p) if GF(p)(g)^i==r1]
[13]
```

Поэтому p нужно взять столь большим, чтобы этот перебор фактически нельзя было осуществить.

6. Задания

Теоретические задания.

1) Докажите, что существует бесконечно много простых чисел.

2) При каких условиях факторкольцо $\mathbb{Z}/p\mathbb{Z}$ является полем.

3) Что такое поле Галуа?

Практические задания.

1) Дана система уравнений

$$x - 5y + z = 3, \quad 3x - 2y + 2z = 1, \quad 2y + 2z = -3$$

Найдите ее решение в полях Галуа при $p = 2, 3, 5, 7, 11$.

2) Определите ранг системы

а) $x - 5y + z = 3, \quad 3x - 2y + 2z = 1, \quad 8y + 2z = -3$

б) $x - 5y + z = 3, \quad 3x - 2y + 2z = 1, \quad -7x - 4y - 4z = 3$

в) $x - 5y + z = 3, \quad 3x - 2y + 2z = 1, \quad -7x - 4y - 4z = 4$

г) $x - 5y + z = 3, \quad 3x - 2y + 2z = 1$

из $\mathrm{GF}(3)[x, y, z]$.

3) Сколько решений имеет система

$$x + y + z = 1, \quad x - y + z = 2, 2x + 2z = 3$$

в полях Галуа при $p = 2, 3, 5, 7, 11$.

4) Найдите матрицу, обратную к матрице

$$\begin{pmatrix} 1 & 2 & 3 \\ 3 & 3 & 2 \\ 3 & 3 & 0 \end{pmatrix}$$

над $\mathrm{GF}(p)$ при $p = 5, 7, 11$. Всегда ли матрица обратима?