Методическая разработка
для проведения лекции
Занятие 4. Алгебраические структуры. Поля Галуа

Учебные вопросы занятия:

1. Недвоичные поля Галуа. Основные понятия.
2. Полиномиальная форма представления недвоичного поля.

Заключительная часть

Содержание занятия:

1. Недвоичные поля Галуа. Основные понятия.

В различных практических приложениях (кодировании, криптографии) чаще всего приходится использовать четыре основные *операции* (*сложение*, *вычитание*, *умножение* и *деление*). Это обуславливает необходимость применения в рамках этих приложений теории множеств и конкретно полей. Однако, с учетом того, что вся обработка ведется с применением ПЭВМ, положительные целые числа сохраняются там как n-битовые слова, в которых n обычно принимает значения 8, 16, 32, 64, и так далее. Это означает, что *диапазон* используемых целых чисел — от 0 до 2^n– 1. Значение *модуля* равно 2^n. Так что в этой ситуации принципиально возможны два варианта, если есть необходимость использовать *поле*.

1. Мы можем задействовать GF(p) с множеством Z_p, где p — наибольшее *простое число*, меньшее, чем 2^n. Но эта схема нецелесообразна, так как мы не можем использовать целые числа от p до 2^n- 1, и поэтому придется использовать дополнительные процедуры по их исключению. Например, если n = 4, то наибольшее *простое число*, меньшее, чем 2^4, — это 13. Это означает, что мы не можем использовать целые числа 13, 14 и 15. Если n = 8, наибольшее *простое число*, меньшее, чем 2^8, — это 251, так что мы не можем использовать 251, 252, 253, 254 и 255.

2. Мы можем работать в GF(2^n) и использовать множество 2^n элементов. Элементы в этом множестве — n-битовые слова. Например, если n = 3, множество представляет собой:

{000,001,010,011, 100,101,110,111}.

Однако мы не можем интерпретировать элементы как *целые числа* от 0 до 7, потому что к ним не могут быть применены обычные четыре *операции* (*модуль* 2^n — не *простое число*). Поэтому следует определить новое множество 2-битовых слов и две новые *операции*, которые удовлетворяют свойствам, определенным для поля.

Пример 1

Определим GF(2^2) как *поле*, в котором множество имеет четыре слова по 2 бита: {00, 01, 10, 11}. Мы можем переопределить операции *сложения* и *умножения* для этого поля таким образом, чтобы все свойства этих операций были удовлетворены, как это показано в таблице.

Нейтральный элемент 00

+	00	01	10	11
00	00	01	10	11
01	01	00	11	10
10	10	11	00	01
11	11	10	01	00

Нейтральный элемент 01

×	00	01	10	11
00	00	00	00	00
01	00	01	10	11
10	00	10	11	01
11	00	11	01	10

Каждое *слово* — аддитивная *инверсия* себя. Каждое *слово* (кроме 00) имеет мультипликативную *инверсию*. Мультипликативные обратные пары — (01,01) и (10, 11). *Сложение* и *умножение* будет определено в терминах полиномиалов.

2. Полиномиальная форма представления недвоичного поля.

Хотя мы можем непосредственно определить правила для операций сложения и умножения слов из двух бит, которые удовлетворяют свойствам в $GF(2^n)$, проще работать с полиномиальным степени n–1 побитным представлением слов. *Полиномиальное* выражение степени n–1 имеет форму

$$f(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \ldots\ldots + a_1x^1 + a_0x^0,$$

где x^i назван термином "i-тый элемент", а a_i называется коэффициентом i-того элемента. Хотя мы знаем полиномы в алгебре, но при представлении n-битовых слов полиномами необходимо следовать некоторым правилам:

1. степень x определяет позицию бита в n-битовых словах. Это означает, что крайний правый бит находится в нулевой позиции (связан с x^0), самый левый бит находится в позиции n–l (связан с x^{n-l});

2. коэффициенты сомножителей определяют значение битов. Каждый бит принимает только значение 0 или 1, поэтому наши полиномиальные коэффициенты могут иметь значение 0 или 1.

Пример 2

Использование полиномов для представления слова из 8 бит (10011001) показано на рисунке 1.

слово	1	0	0	1	1	0	0	1
	↓	↓	↓	↓	↓	↓	↓	↓
полином	$1x^7$	$+0x^6$	$+0x^5$	$+1x^4$	$+1x^3$	$+0x^2$	$+0x^1$	$+1x^0$

$1x^7+1x^4+1x^3+1x^0$

$x^7+x^4+x^3+x^0$

Рис. 1. Представление 8-ми битового слова полиномом

Заметим, что элемент пропущен, если его коэффициент равен 0, и пропущен только коэффициент, если это 1. Также заметим, что элемент x^0 равен 1.

Пример 3

Чтобы найти слово из 8 битов, связанное с полиномом x^5+x^2+x, мы сначала восстановим пропущенные сомножители. Имеем n = 8, что означает *полином* степени 7. Расширенный *полином* имеет вид

$0x^7 + 0x^6 + 1x^5 + 0x^4 + 0x^3 + 1x^2 + 1x^1 + 0x^0$

Он связан со словом из 8 битов 00100110.

Операции

Обратите внимание, что любая операция в полиномиальных формах фактически включает две операции: операции И коэффициентов двух полиномов. Другими словами, мы должны определить два поля: одно для коэффициентов и одно для полиномов. Коэффициенты равны 0 или 1; для этой цели мы можем использовать поле GF(2). Мы уже говорили о таком поле. Для полиномов нам нужно поле $GF(2^n)$, которое коротко обсудим ниже.

Таким образом, полиномы, представляющие n-битовые слова, используют два поля: GF(2) и $GF(2^n)$.

Модуль (неприводимый полином)

Перед определением операций на полиномах мы должны поговорить о полиномах-модулях. Сложение двух полиномов никогда не создает *полином*, выходящий из множества. Однако умножение двух полиномов может создать *полином* со степенью большей, чем n – 1. Это означает, что мы должны делить результат на модуль и сохранять только остаток, как это делается в модульной арифметике. Для множеств полиномов в $GF(2^n)$ группа полиномов степени n определена как модуль. Модуль в этом случае действует как *полиномиальное простое число*. Это означает, что никакие полиномы множества не могут делить этот *полином*. Простое полиномиальное число не может быть разложено в полиномы со степенью меньшей, чем n. Такие полиномы называются **неприводимые полиномы.** В таблице 1 показаны примеры полиномов 1-5 степеней.

Таблица 1.

Степень	Неприводимый полином
1	x; $x+1$
2	x^2+x+1
3	x^3+x^2+1; x^3+x+1
4	$x^4+x^3+x^2+x+1$; x^4+x^3+1; x^4+x+1
5	x^5+x^2+1; $x^5+x^3+x^2+x+1$; $x^5+x^4+x^3+x+1$; $x^5+x^4+x^3+x^2+1$; $x^5+x^4+x^2+x+1$

Для каждого значения степени часто существует более одного неприводимого *полинома*, — это означает, что когда мы определяем поле $GF(2^n)$, мы должны объявить, какой неприводимый *полином* мы используем как модуль.

Сложение

Теперь определим операцию сложения для полиномов с коэффициентом в GF(2). Операция сложения достаточно проста: мы складываем коэффициенты соответствующих элементов полинома в поле GF(2). Обратите внимание, что сложение двух полиномов степени n – 1 всегда дает *полином* со степенью n – 1 — это означает, что мы не должны использовать вычитание из модуля их результата.

Пример 4

Произведем сложение x^5+x^2+x и x^3+x^2+1 в $GF(2^8)$. Мы используем символ © для обозначения полиномиального сложения. Ниже показана процедура
$0x^7+0x^6+1x^5+0x^4+0x^3+1x^2+1x^1+0x^0$©$0x^7+0x^6+0x^5+0x^4+1x^3+1x^2+0x^1+1x^0=$
$0x^7+0x^6+1x^5+0x^4+1x^3+0x^2+1x^1+1x^0 \rightarrow x^5+x^3+x+1$.

В упрощенном полиноме сохранены элементы с коэффициентом 1 и удалены элементы с коэффициентом 0. Кроме того, удалены совпадающие элементы обоих полиномов, а несовпадающие сохраняются. Другими словами, x^5, x^3, и x^1 сохраняются, а x^2, который является совпадающим в этих двух полиномах, удален.

Пример 5

Поскольку сложение в GF(2) означает операцию *ИСКЛЮЧАЮЩЕЕ ИЛИ* (*XOR*), мы можем получить результат *ИСКЛЮЧАЮЩЕГО ИЛИ* для этих двух слов бит за битом. В предыдущем примере $x^5+ x^2+ x$ есть 00100110, и $x^3+ x^2+ 1$ есть 00001101. Результат — 00101011 или, в полиномиальном представлении, x^5+x^3+x+1.

Аддитивный нейтральный элемент — тождество. Аддитивный нейтральный элемент полинома — нулевой *полином* (*полином* со всеми коэффициентами, равными нулю), потому что, прибавляя этот *полином* к самому себе, в результате получаем исходный *полином*.

Аддитивная инверсия полинома с коэффициентами в GF(2) — сам *полином*. Это означает, что операция вычитания та же самая, что и операция сложения.

Операции сложения и вычитания на полиномах — одинаковые операции.

Умножение

Умножение в полиномах — сумма умножения каждого элемента одного полинома и каждого элемента второго полинома. Необходимо отметить три особенности.

Первая: умножение коэффициента проводится в поле GF(2).

Вторая: умножение x^i на x^j дает результат x^{i+j}.

Третья: умножение может создать элементы со степенью большей, чем n–1, и это означает, что степень должна быть уменьшена с использованием полинома-модуля.

Сначала проследим, как умножить два полинома согласно вышеупомянутому определению. Позже рассмотрим вариант алгоритма, который может использоваться в программе.

Пример 6

Найдите результат $(x^5+x^2+x)®(x^7+x^4+x^3+x^2+x)$ в $GF(2^8)$ с неприводимым (неразлагаемым) полиномом $(x^8+x^4+x^3+x+1)$. Обратите внимание, что для обозначения умножения двух полиномов используется символ ®.

Решение

Сначала умножаем эти два полинома так, как это делается в обычной алгебре. Обратите внимание, что в этом процессе пара элементов с равной степенью сокращается. Например, результат x^9+x^9 сокращается, потому что он является нулевым по результатам операции сложения.

$P_1®P_2=x^5(x^7+x^4+x^3+x^2+x)+x^2(x^7+x^4+x^3+x^2+x)+x(x^7+x^4+x^3+x^2+x)=x^{12}+x^9+x^8+x^7+x^6+x^9+x^6+x^5+ x^4+x^3+x^8+x^5+x^4+x^3+x^2=x^{12}+x^7+x^2 \bmod (x^8+x^4+x^3+x+1)=x^5+x^3+x^2+x+1.$

Чтобы найти конечный результат, разделим *полином* степени12 на *полином* степени 8 (модуль) и сохраним остаток. Процесс деления тот же самый, что и в обычной алгебре, но необходимо помнить, что в данной ситуации вычитание аналогично сложению.

$x^{12}+x^7+x^2$	$x^8+x^4+x^3+x+1$
$x^{12}+x^8+x^7+ x^5+x^4$	x^4+1
$x^8+x^5+x^4+ x^2$ $x^8+x^4+x^3+x+1$	
$x^5+x^3+x^2+x+1$	Остаток

Мультипликативное тождество — всегда равно 1. Например, в $GF(2^8)$ мультипликативная *инверсия* — в побитном изображении 00000001.

Мультипликативная инверсия. Поиск мультипликативной *инверсии* требует привлечения расширенного *алгоритма Евклида*. *Алгоритм Евклида* следует применять к модулю и полиному, причем алгоритм реализуется таким же образом, как и для целых чисел.

Пример 7

В $GF(2^4)$ найдите *инверсию* $(x^2+1) \bmod (x^4+x+1)$.

Решение

q	$\mathbf{r_j}$	$\mathbf{r_2}$	**r**	$\mathbf{t_j}$	$\mathbf{t_2}$	**t**
(x^2+1)	(x^4+x+1)	(x^2+1)	(x)	(0)	(1)	(x^2+1)
(x)	(x^2+1)	(x)	(1)	(1)	(x^2+1)	(x^3+x+1)
(x)	(x)	(1)	(0)	(x^2+1)	(x^3+x+1)	(0)
	(1)	(0)		(x^3+x+1)	(0)	

Это означает, что $(x^2+1)^{-1} \bmod (x^4+x+1)$ есть (x^3+x+1). Ответ может быть проверен просто: надо перемножить эти два полинома и найти остаток. В этом случае результат деления на модуль равен

$$(x^2+1)®(x^3+x+1) \bmod (x^4+x+1)=1.$$

Пример 8

В $GF(2^8)$ найдите *инверсию* $(x^5) \bmod (x^8+x^4+x^3+x+1)$.

Решение

q	$\mathbf{r_j}$	$\mathbf{r_2}$	**r**	$\mathbf{t_j}$	$\mathbf{t_2}$	**t**
(x^3)	$(x^8+x^4+x^3+x+1$	(x^5)	$(x^4+x^3+x^2+x+1)$	(0)	(1)	(x^3)
$(x+1)$	(x^5)	(x^4+x^3+x+1)	(x^3+x^2+1)	(1)	(x^3)	(x^4+x^3+1)

(x)	(x^4+x^3+x+1)	(x^3+x^2+1)	(1)	(x^3)	(x^4+x^3+1)	$(x^5+x^4+x^2+x)$
	(x^3+x^2+1)	(1)	(0)	(x^4+x^3+1)	$(x^5+x^4+x^2+x)$	(0)
	(1)	(0)		$(x^5+x^4+x^2+x)$	(0)	

Это означает, что $(x^5)^{-1}$ mod $(x^8+x^4+x^3+x+1)$ есть $(x^5+x^4+x^2+x)$.

Результат может быть легко проверен умножением этих двух полиномов и определением остатка деления по модулю

$$(x^5)®(x^5+x^4+x^2+x)\text{mod}(x^8+x^4+x^3+x+1)=1.$$

Умножение с использованием ПЭВМ

Операция деления порождает проблему написания эффективной программы умножения двух полиномов. Лучший алгоритм для компьютерной реализации использует неоднократное умножение полинома на x. Например, вместо того чтобы находить результат $(x^2)®(P_2)$, программа находит результат $(x® (x®P_2)$. Преимущество этой стратегии будет обсуждаться далее, но сначала рассмотрим пример, чтобы проиллюстрировать алгоритм.

Пример 9

Найдите результат умножения $P_1 = (x^5+ x^2+ x)$ на $P_2 = (x^7+ x^4+ x^3+ x^2+ x)$ в поле $GF(2^8)$ с неприводимым полиномом $(x^8+x^4+x^3+x+1)$, используя алгоритм, изложенный выше.

Решение

Результат показан в таблице. В начале находится промежуточный результат умножения x^0, x^1, x^2, x^3, x^4 и x^5. Заметим, что необходимы только три составляющие произведения $(x^m)®P_2$ для m от 0 до 5, каждое вычисление зависит от предыдущего результата.

Таблица. Алгоритм перемножения полиномов

Степень	Операция	Новый результат	Вычитание
$x^0®P_2$		$x^7+x^4+x^3+x^2+x$	НЕТ
$x^1®P_2$	$x®(x^7+x^4+x^3+x^2+x)$	x^5+x^2+x+1	ДА
$x^2®P_2$	$x®(x^5+x^2+x+1)$	$x^6+x^3+x^2+x$	НЕТ
$x^3®P_2$	$x®(x^6+x^3+x^2+x)$	$x^7+x^4+x^3+x^2$	НЕТ
$x^4®P_2$	$x®(x^7+x^4+x^3+x^2)$	x^5+x+1	ДА
$x^5®P_2$	$x® (x^5+x+1)$	x^6+x^2+x	НЕТ
$P_1®P_2=(x^6+x^2+x)+(x^6+x^3+x^2+x)+(x^5+x^2+x+1)=x^5+x^3+x^2+x+1$			

Рассмотренный выше алгоритм имеет два преимущества. Первое — умножение полинома на x может быть выполнено простым сдвигом одного бита в n-битовом слове; операция может быть реализована на любом языке программирования. Второе — результат может быть использован, если максимальная степень полинома n–1. В этом случае сокращение может быть сделано просто с помощью применения операции *ИСКЛЮЧАЮЩЕЕ ИЛИ* с заданным модулем. В нашем примере самая высокая степень — только 8. Мы можем разработать простой алгоритм для нахождения промежуточных результатов.

1. Если старший разряд предыдущего результата равен 0, тогда надо сдвинуть предыдущий результат на один бит влево.

2. Если старший бит предыдущего результата равен 1: *а*) надо сдвинуть на один бит влево; б) применить к нему операцию *ИСКЛЮЧАЮЩЕЕ ИЛИ* с модулем, исключив из этой операции старший разряд.

Повторим пример 9 для двоичной последовательности размером 8 бит. При этом P_1=00100110, P_2=10011110, модуль=100011010 (девять битов). Обозначим операцию *ИСКЛЮЧАЮЩЕЕ ИЛИ* как ©. Пример приведен в таблице.

Таблица. Эффективное умножение с применением n-битового слова

Степень	*Операция сдвига влево*	*ИСКЛЮЧАЮЩЕЕ ИЛИ*
x^0®P_2		10011110
x^1®P_2	00111100	(00111100)©(00011010)=00100111
x^2®P_2	01001110	01001110
x^3®P_2	10011100	10011100
x^4®P_2	00111000	(00111000)©(00011010)=00100011
x^5®P_2	01000110	01000110
P_1®P_2= (000100110)©(01001110)©(01000110)=00101111		

В этом случае для умножения данных полиномов необходимо только пять операций левого сдвига и четыре *ИСКЛЮЧАЮЩЕЕ ИЛИ*. Вообще, для умножения двух полиномов степени n-1 необходимо максимально (n–1) операций левого сдвига и 2n операций *ИСКЛЮЧАЮЩЕЕ ИЛИ*.

Таким образом, умножение полиномов в GF(2^n) может быть выполнено с помощью операций левого сдвига и ИСКЛЮЧАЮЩЕЕ ИЛИ.

Пример 10

Поле GF(2^3) состоит из 8 элементов. Рассмотрим процесс умножение и сложение таблиц для этого поля, используя неприводимый *полином* x^3+x^2+1. Будем оперировать с трехбитовым словом и полиномом. Заметим, что имеется два неприводимых полинома третьей степени (см. таблицу неприводимых полиномов). Другой *полином* (x^3+x+1) для умножения имеет таблицу, полностью отличающуюся от первой. Таблица сложения показывает результаты сложения.

Таблица сложения

©	**000 (0)**	**001 (1)**	**010 (x)**	**011 (x+1)**	**100 (x^2)**	**101 (x^2+1)**	**110 (x^2 +x)**	**111 (x^2+x+1)**
000 (0)	000 (0)	001 (1)	010 (x)	011 (x+1)	100 (x^2)	101 (x^2+1)	110 (x^2+x)	111 (x^2+x+1)
001 (1)	001 (1)	000 (0)	011 (x+1)	010 (x^2)	101 (x^2+1)	100 (x^2+x)	111 (x^2+x+1)	110 (x^2+x)
010 (x)	010 (x)	011 (x+1)	000 (0)	001 (1)	110 (x^2+x)	111 (x^2+x+1)	100 (x^2+x)	101 (x^2+1)

011 (x+1)	011 (x+1)	010 (x)	001 (1)	000 (0)	111 (x^2+x+1)	110 (x^2+x)	101 (x^2+1)	100 (x^2)
100 (x^2)	100 (x^2)	101 (x^2+1)	110 (x^2+x)	111 (x^2+x+1)	000 (0)	001 (1)	010 (x)	011 (x+1)
101 (x^2+1)	101 (x^2+1)	100 (x^2)	111 (x^2+x+1)	110 (x^2+x)	001 (1)	000 (0)	011 (x+1)	010 (x)
110 (x^2 + x)	110 (x^2+x)	111 (x^2+x+1)	100 (x^2)	101 (x^2+1)	010 (x)	011 (x+1)	000 (0)	001 (1)
111 (x^2+x+1)	111 (x^2+x+1)	110 (x^2+x)	101 (x^2+1)	100 (x^2)	011 (x+1)	010 (x)	001 (1)	000 (0)

Таблица умножения показывает результаты умножения.

Таблица умножения

®	**000 (0)**	**001 (1)**	**010 (x)**	**011 (x+1)**	**100 (x^2)**	**101 (x^2+1)**	**110 (x^2+x)**	**111 (x^2+x+1)**
000 (0)	000 (0)	000 (0)	000 (0)	000 (0)	000 (0)	000 (0)	000 (0)	000 (0)
001 (1)	000 (0)	001 (1)	010 (x)	011 (x+1)	100 (x^2)	101 (x^2+1)	110 (x^2+x)	111 (x^2+x+1)
010 (x)	000 (0)	010 (x)	100 (x)	110 (x^2+x)	101 (x^2+1)	111 (x^2+x+1)	001 (1)	011 (x+1)
011 (x+1)	000 (0)	011 (x+1)	110 (x^2+x)	101 (x^2+1)	001 (1)	010 (x)	111 (x^2+x+1)	100 (x^2)
100 (x^2)	000 (0)	100 (x^2)	101 (x^2+1)	001 (1)	111 (x^2+x+1)	011 (x+1)	010 (x)	110 (x^2+x)
101 (x^2+1)	000 (0)	101 (x^2+1)	111 (x^2+x+1)	010 (x)	011 (x+1)	110 (x^2+x)	100 (x^2)	001 (1)
110 (x^2 + x)	000 (0)	110 (x^2+x)	001 (1)	111 (x^2+x+1)	010 (x)	100 (x^2)	011 (x+1)	101 (x^2+1)
111 (x^2+x+1)	000 (0)	111 (x^2+x+1)	011 (x+1)	100 (x^2)	110 (x^2+x)	001 (1)	101 (x^2+1)	010 (x)

Использование генератора

Иногда проще определить элементы поля GF(2^n), используя *генератор*, называемый порождающим полиномом. В этом *поле* с неприводимым полиномом f(x) и элементом поля *a* нужно удовлетворить *отношение* f(*a*)=0. В частности, если g — *генератор* поля, то f(g)=0. Тогда можно доказать, что элементы поля могут быть сгенерированы как

{0, 1, g^2,.... g^n}, где N = 2^n–2.

Пример 11

Для генерирования элементов поля GF(2^4) используйте *полином* f(x)=x^4+x+1.

Решение

Элементы 0, g^0, g^1, g^2 и g^3 могут быть сгенерированы достаточно просто, потому что в 4-*битовом поле* они представлены 0, x^0, x^1, x^2 и x^3 (не требуется деления на *полином*). Элементы от g^4 до g^{14}, которые содержат от x^4 до x^{14}, формируются путем деления на неприводимый *полином*. Для такого деления следует использовать *полином* f(g)=g^4+g+1=0. Применив это *отношение*, мы

имеем $g^4= –g–1$. Поскольку *сложение* полей и *вычитание* полей — та же самая операция, $g^4=g+1$. Мы используем это *отношение*, чтобы найти *значение* всех элементов в виде 4-битовых слов:

$0 = 0 = 0 = 0 \rightarrow 0=(0000)$

$g^0 = g^1 = g^1 \rightarrow g^1 = (0010)$

$g^2 = g^2 = g^2 \rightarrow g^2 = (0100)$

$g^3 = g^3 = g^3 \rightarrow g^3 = (1000)$

$g^4 = g^4 = g + 1 \rightarrow g^4 = (0011)$

$g^5 = g(g + 1) = g^2 + g \rightarrow g^5 = (0110)$

$g^6 = g(g^2 + g) = g^3 + g^2 \rightarrow g^6 = (1100)$

$g^7 = g(g^3 + g) = g^3 + g + 1 \rightarrow g^7 = (1011)$

$g^8 = g(g^3 + g + 1) = g^2 + 1 \rightarrow g^8= (0101)$

$g^9 = g(g^2 + 1) = g^3 + g \rightarrow g^9 = (1010)$

$g^{10} = g(g^3 + g) = g^2 + g + 1 \rightarrow g^{10} = (0111)$

$g^{11} = g(g^2 + g + 1) = g^3 + g^2 + g \rightarrow g^{11} = (1110)$

$g^{12} = g(g^3 + g^2 + g) = g^3 + g^2 + g + 1 \rightarrow g^{12} = (1111)$

$g^{13} = g(g^3 + g^2 + g + 1) = g^3 + g^2 + 1 \rightarrow g^{13} = (1101)$

$g^{14} = g(g^3 + g^2 + 1) = g^3 + 1 \rightarrow g^{14} = (1001)$

Основная идея состоит в том, что *вычисление* элементов поля от g^4 до g^{14} сводится к использованию соотношения $g^4=g+1$ и результатов предыдущих вычислений. Например,

$g^{12}=g(g^{11})=g(g^3+g^2+g)=g^4+g^3+g^2=g^3+g^2+g+1$

После сокращения можно просто преобразовать степени в n-битовое *слово*. Например, g^3+1 эквивалентно 1001, потому что присутствуют элементы со степенью 0 и 3. Заметим, что элементы с одинаковой степенью при таком процессе вычисления взаимопоглощают друг друга. Например, $g^2+g^2=0$.

Инверсии

Нахождение *инверсий* при использовании приведенного выше метода представления достаточно просто.

Аддитивные инверсии

Аддитивная *инверсия* каждого элемента — непосредственно сам элемент, потому что сложение и вычитание в этом поле — одна и та же операция, например, $g^3=g^3$.

Мультипликативные инверсии

Найти мультипликативную *инверсию* каждого элемента также несложно. Например, можно найти мультипликативную *инверсию* элемента g^3, как показано ниже:

$(g^3)^{-1}=g^{-3}=g^{12}=g^3+g^2+g+1\rightarrow(1111)$.

Заметим, что в этом случае степень рассчитывается по модулю $2^n–1$, $2^4–1=15$.

Поэтому $–3 \bmod 15 = 12 \bmod 15$.

Можно легко доказать, что g^3 и g^{12} есть инверсные (обратные числа), потому что $g^3g^{12}=g^{15}=g^0=1$.

Сложение и вычитание

Сложение и вычитание — это одинаковые операции. Промежуточные результаты могут быть упрощены, как проиллюстрировано в следующем примере.

Пример 12

Этот пример показывает результаты операций сложения и вычитания:

a. $g^3+g^{12}+g^7=g^3+(g^3+g^2+g+1)+(g^3+g+1)= g^3+g^2\rightarrow(1100)$.

b. $g^3-g^6=g^3+g^6=g^3+(g^3+g^2)=g^2\rightarrow(0100)$.

Умножение и деление

Умножение есть сложение степени по модулю $2^n–1$. Деление — это умножение, которое использует мультипликативную *инверсию*.

Пример 13

a. $g^9\times g^{11}=g^{20}=g^{20 \bmod 15}=g^5=g^2+g\rightarrow(0110)$.

б. $g^3/g^8=g^3\times g^7=g^{10}=g^2+g+1\rightarrow(0111)$.

Заключительная часть

Подводя итоги, следует отметить, что конечное *поле* GP (2^n) может использоваться для того, чтобы определить четыре *операции — сложение, вычитание*, *умножение* и *деление* n-битных слов. Только *деление* на нуль не определено. Каждое n-битовое *слово* может быть представлено как *полином* степени n–1 с коэффициентами в GF(2), — это означает, что *операции* на n-битовых словах могут быть представлены как *операции* на этом полиноме. При умножении двух полиномов необходимо сделать эти *операции* операциями по модулю. Для этого мы должны определить неприводимый *полином* степени n. Чтобы найти мультипликативные *инверсии* к полиномам, может быть применен расширенный *алгоритм Евклида*.

Контрольные вопросы

1. Покажите, как *полином* может представить n-битовое слово.
2. Определите неприводимый *полином*.