Finite Fields

Division – GF( 2^8 )



# Division

We can easily add, subtract, and multiply polynomials in ZP[X], but division is a little more subtle. L et’s look a t an example. The polynomial X 8 + X 4 + X 3 + X + 1 is irreducible in ZzlXJ (although there are faster methods, one way to show it is irreducible is to divide it by all polynomials of smaller degree in Z 2[X']). Consider the field

G F ( 28) = Z 2[X] (mod X8 + X4 + X3 + X + 1).

Since X7 + X6 + X3 + X + 1 is not 0, it should have an inverse. The inverse is found using the analog of the extended Euclidean algorithm. First, perform the gcd calculation for gcd( X7 + X6 + X3 + X + 1 ,X8 + X4 + X3 + X + 1).

The procedure (remainder —> divisor —> dividend —> ignore) is the same as for integers:

X8 + X4 + X3 + X + l = ( X + l) ( X7 + X6 + X3 + X -l)-( X6 + X2 + X )

X7 + X6 + X3 + X + l = (X + 1 )(X6 + X2 + X )+ 1 .

The last remainder is 1, which tells us that the "greatest common divisor" of X7 + X6 + X3 + X + 1 and X8 + X4 + A3 + X + 1 is 1. Of course, this must be the case, since X8 + X4 + X3 + X + 1 is irreducible, so its only factors are 1 and itself.

Now work back through the calculation to express 1 as a linear combination of X 7 + X G+ X 3 + X + 1 and X8 + X4 + X3 + X + 1 (or use the formulas for the extended Euclidean algorithm). Recall th a t in each step we take the Inst unused remainder and replace it by the dividend minus the quotient times the divisor; since we are working mod 2, the minus signs disappear.

1 = (X7 + X6 + X3 + X -l)+ ( X + 1 ) (X ° + X 2-I-X)

= (X7 + X6 + X3 + X -l) + (X + 1 ) (X8 + X4 + X3 + X + 1)+ ( X + 1 ) (X7 + X6 + X3 + X -l)

= ( l + ( X + l )2 ( X7 + X6 + X3 + X -l) + ( X + l) (X8 + X4 + X3 + X + 1)

= (X2)( X7 + X6 + X3 + X + l) + ( X + l ) (X8 + X4 + X3 + X + 1) .

Therefore,

1. = (X2)( X7 + X6 + X3 + X + l) + ( X + l ) (X8 + X4 + X3 + X + 1).

Reducing mod X8 + X4 + X3 + X + 1, we obtain

(X2)( X7 + X6 + X3 + X + l) = 1 (mod X8 + X4 + X3 + X + 1).

which means th at X 2 is the multiplicative inverse of X7 + X6 + X3 + X -l. Whenever we need to divide by X8 + X4 + X3 + X + 1, we can instead multiply by X2. This is the analog of what we did when working with the usual integers mod p.

# GF (28)

Lliter in this book, we shall discuss Rijndael, which uses G F (28), so let’s look a t this field a little more closely. We'll work mod the irreducible polynomial X s + X\*1 + X 3 + X + 1, since th at is the one used by Rijndael. However, there are other irreducible polynomials of degree 8, and any one of them would lead to similar calculations. Every element can be

represented uniquely as a polynomial

b7X7 + b6X6 + b5X5 + b4X4 + b3X3 + b2X2 + b1X+ b0

where each bi is 0 or 1. The **8** bits **b6b5b4b3b2b1b0** represent a byte, so we can represent the elements of G F (**2**8) as **8**-bit bytes. For example, the polynomial X7 + X6 + X3 + X + l becomes 11001011. Addition is the XOR of the bits:

(X7 + X6 + X3 + X + l) + (X4 + X3 + l)

— >11001011 00011001 = 11010010

—>X7 + X6 + X3 + X + l

M ultiplication is more subtle and does not have os easy an interpretation. That is because we are working mod the polynomial X8 + X4 + X3 + X + 1, which we can represent by the 9 bits 100011011. First, let’s multiply X7 + X6 + X3 + X + l by X. With polynomials, we calculate

(X7 + X6 + X3 + X + l)(X) = X8 + X7 + X4 + X3 + X

= (X7 + X6 + X3 + X + l)-( X8 + X7 + X4 + X3 + X)

= X7 + X6 + X3 + X + l (mod X8 + X7 + X4 + X3 + X).

The same operation with bits becomes

11001011 —> 110010110 (shift left and append a 0)

—> 110010110 100011011 (subtract X8 + X7 + X4 + X3 + X)

= 010001101,

which corresponds to the preceding answer. In general, we can multiply by X by the following algorithm:

1. Shift left and append a 0 as the lost bit,

2. If the first bit is 0, stop.

3. If the first bit is 1, X O R with 100011011.

The reason we stop in step 2 is th at if the first bit is 0 then the polynomial still has degree less than 8 after we multiply by X , so it does not need to be reduced. To multiply by higher powers of X , multiply by X several times. For example, m ultiplication by X3 can be done with three shifts and at most three XORs. M ultiplication by an arbitrary polynomial can be accomplished by multiplying by the various powers of X appearing in that polynomial,

then adding (i.e., XORing) the results.

In summary, we see th at the fields operations of addition and multiplication in G F (28) can be carried out very efficiently. Similar considerations uppiy to any finite field.

The analogy between the integers mod a prime and polynomials mod an irreducible polynomial is quite remarkable. We summarize in the following.

integers <— > Zp[X]

prime number q <— > irreducible P ( X ) of degree n

Zq <— > Zp [X] (mod P( X ) )

field w ith q elements <— > field with p n elements

Let GF( pn)\* denote the nonzero elements of GF( p n). This set, which has pn — 1 elements, is closed under ultiplication, just as the integers not congruent to 0 mod p are closed under multiplication. It can be shown that there is a generating polynomial g( X) such th at every element in G F (pn)m can be expressed as a power of g(X). This also means th at the smallest exponent k such th at g (X)k ≡ 1 is pn — 1. This is the analog of a primitive root for primes. There are 0(pn — 1) such generating polynomials, where 0 is Euler's function. An Interesting situation occurs when p = 2 and 2" — I

is prime. In this case, every nonzero polynomial f ( X ) ≠ 1 in G F(2n) is a generating polynomial. (Remark, fo r those who know some group theory: The set G F (2n)\* is a group of prime order in this cose, so every element

except the identity is a generator.)

The discrete problem mod a prime, which we’ll discuss in Chapter 7, has an analog for finite fields; namely, given h(x), find an integer k such thath{ X) = g ( X )k in G F(p"). Finding such a k is believed to be very hard in most situations.