1 Polynomial Practice

(a) If f and g are non-zero real polynomials, how many roots do the following polynomials have at least? How many can they have at most? (Your answer may depend on the degrees of f and g.)

(i) $f+g$

(ii) $f \cdot g$

(iii) f/g, assuming that f/g is a polynomial

(b) Now let f and g be polynomials over $\mathrm{GF}(p)$.

(i) If $f \cdot g = 0$, is it true that either $f = 0$ or $g = 0$?

(ii) If $\deg f \geq p$, show that there exists a polynomial h with $\deg h < p$ such that $f(x) = h(x)$ for all $x \in \{0, 1, ..., p-1\}$.

(iii) How many f of degree *exactly* $d < p$ are there such that $f(0) = a$ for some fixed $a \in \{0, 1, \ldots, p-1\}$?

(c) Find a polynomial f over $\mathrm{GF}(5)$ that satisfies $f(0) = 1, f(2) = 2, f(4) = 0$. How many such polynomials are there?

Solution:

(a) (i) It could be that $f+g$ has no roots at all (example: $f(x) = 2x^2 - 1$ and $g(x) = -x^2 + 2$), so the minimum number is 0. However, if the highest degree of $f+g$ is odd, then it has to cross the x-axis at least once, meaning that the minimum number of roots for odd degree polynomials is 1 (we did not look for this case when grading). On the other hand, $f+g$ is a polynomial of degree at most $m = \max(\deg f, \deg g)$, so it can have at most m roots. The one exception to this expression is if $f = -g$. In that case, $f+g = 0$, so the polynomial has an infinite number of roots!

(ii) A product is zero if and only if one of its factors vanishes. So if $f(x) \cdot g(x) = 0$ for some x, then either x is a root of f or it is a root of g, which gives a maximum of $\deg f + \deg g$ possibilities. Again, there may not be any roots if neither f nor g have any roots (example: $f(x) = g(x) = x^2 + 1$).

(iii) If f/g is a polynomial, then it must be of degree $d = \deg f - \deg g$ and so there are at most d roots. Once more, it may not have any roots, e.g. if $f(x) = g(x)$.

(b) (i) $x^{p-1}-1$ and x are both non-zero polynomials on $GF(p)$ for any p. x has a root at 0, and by Little Fermat, $x^{p-1}-1$ has a root at all non-zero points in $GF(p)$. So, their product $x \cdot x^{p-1}$ must have a zero on all points in $GF(p)$.

(ii) Little Fermat tells us that $x^s \equiv x \cdot x^{(s-1) \bmod (p-1)} \pmod p$ (note that we have to factor one x out to account for the possibility that $x = 0$), and since $(s-1) \bmod (p-1) \leq p-2$, writing $f(x) = \sum_{k=0}^{n} a_k x^k$, we have that $h(x) = a_0 + \sum_{k=1}^{n} a_k x \cdot x^{(k-1) \bmod (p-1)}$ is a polynomial of degree $\leq p-1$ with $f(x) = h(x)$.

(iii) We know that in general each of the $d+1$ coefficients of $f(x) = \sum_{k=0}^{d} c_k x^k$ can take any of p values. However, the conditions $f(0)$ and $\deg f = d$ impose constraints on the constant coefficient $f(0) = c_0 = a$ and the top coefficient $x_d \neq 0$. Hence we are left with $(p-1) \cdot p^{d-1}$ possibilities.

(c) We know by part (b) that any polynomial over GF(5) can be of degree at most 4. A polynomial of degree ≤ 4 is determined by 5 points (x_i, y_i). We have assigned three, which leaves $5^2 = 25$ possibilities. To find a specific polynomial, we use Lagrange interpolation:

$$\Delta_0(x) = 2(x-2)(x-4) \qquad \Delta_2(x) = x(x-4) \qquad \Delta_4(x) = 2x(x-2),$$

and so $f(x) = \Delta_0(x) + 2\Delta_2(x) = 4x^2 + 1$.

2 The CRT and Lagrange Interpolation

Let $n_1, \ldots n_k$ be pairwise coprime, i.e. n_i and n_j are coprime for all $i \neq j$. The Chinese Remainder Theorem (CRT) tells us that there exist solutions to the following system of congruences:

$$x \equiv a_1 \pmod{n_1} \tag{1}$$
$$x \equiv a_2 \pmod{n_2} \tag{2}$$
$$\vdots \tag{$\vdots$}$$
$$x \equiv a_k \pmod{n_k} \tag{k}$$

and all solutions are equivalent $\pmod{n_1 n_2 \cdots n_k}$. For this problem, parts (a)-(c) will walk us through a proof of the Chinese Remainder Theorem. We will then use the CRT to revisit Lagrange interpolation.

(a) We start by proving the $k = 2$ case: Prove that we can always find an integer x_1 that solves (1) and (2) with $a_1 = 1, a_2 = 0$. Similarly, prove that we can always find an integer x_2 that solves (1) and (2) with $a_1 = 0, a_2 = 1$.

(b) Use part (a) to prove that we can always find at least one solution to (1) and (2) for any a_1, a_2. Furthermore, prove that all possible solutions are equivalent $\pmod{n_1 n_2}$.

(c) Now we can tackle the case of arbitrary k: Use part (b) to prove that there exists a solution x to (1)-(k) and that this solution is unique $\pmod{n_1 n_2 \cdots n_k}$.

(d) For two polynomials $p(x)$ and $q(x)$, mimic the definition of $a \bmod b$ for integers to define $p(x) \bmod q(x)$. Use your definition to find $p(x) \bmod (x-1)$.

(e) Define the polynomials $x-a$ and $x-b$ to be coprime if they have no common divisor of degree 1. Assuming that the CRT still holds when replacing x, a_i and n_i with polynomials (using the definition of coprime polynomials just given), show that the system of congruences

$$\begin{aligned} p(x) &\equiv y_1 \pmod{(x-x_1)} && (1') \\ p(x) &\equiv y_2 \pmod{(x-x_2)} && (2') \\ &\vdots && (\vdots) \\ p(x) &\equiv y_k \pmod{(x-x_k)} && (k') \end{aligned}$$

has a unique solution $\pmod{(x-x_1)\cdots(x-x_k)}$ whenever the x_i are pairwise distinct. What is the connection to Lagrange interpolation?

Solution:

(a) Since $\gcd(n_1, n_2) = 1$, there exist integers k_1, k_2 such that $1 = k_1n_1 + k_2n_2$. Setting $x_1 = k_2n_2 = 1 - k_1n_1$ and $x_2 = k_1n_1 = 1 - k_2n_2$ we obtain the two desired solutions.

(b) Using the x_1 and x_2 we found in Part (a), we show that $a_1x_1 + a_2x_2 \pmod{n_1n_2}$ is a solution to the desired equivalences:

$$\begin{aligned} a_1x_1 + a_2x_2 &\equiv a_1 \cdot 1 + a_2 \cdot 0 \equiv a_1 \pmod{n_1} \\ a_1x_1 + a_2x_2 &\equiv a_1 \cdot 0 + a_2 \cdot 1 \equiv a_2 \pmod{n_2}. \end{aligned}$$

Uniqueness now follows directly from problem 5c in HW4.

(c) We use induction on k. Part (b) handles the base case, $k = 2$. For the inductive hypothesis, assume for $k \le l$, the system (1)-(k) has a unique solution $a \pmod{n_1 \cdots n_k}$. Now consider $k = l+1$, so we add the equation $x \equiv a_{l+1} \pmod{n_{l+1}}$ to our system, resulting in

$$\begin{aligned} x &\equiv a \pmod{n_1 \cdots n_l} \\ x &\equiv a_{l+1} \pmod{n_{l+1}}. \end{aligned}$$

Since the n_i are pairwise coprime, $n_1n_2 \cdots n_l$ and n_{l+1} are coprime. Part (b) tells us that there exists a unique solution $a' \pmod{n_1 \cdots n_l n_{l+1}}$. We conclude that a' is the unique solution to (1)-($l+1$), when taken $\pmod{n_1n_2 \cdots n_l n_{l+1}}$.

(d) $a \bmod b$ is defined as the remainder after division by b. But we know how to divide polynomials and compute remainders too! In particular, we know that we can write $p(x) = q'(x)q(x) + r(x)$ where $\deg r < \deg q$. So we define $p(x) \bmod q(x) = r(x)$.

To compute $p(x) \bmod (x-1)$ then, we write $p(x) = (x-1)q'(x) + r(x)$. We know that $\deg r < \deg(x-1) = 1$ and so r must be a constant. Which constant is it? Plugging in $x = 1$ gives $p(1) = r(1)$ and so $r(x) = p(1)$ for all x.

(e) We only need to check that $q_i(x) = (x - x_i)$ and $q_j(x) = (x - x_j)$ are coprime whenever $x_i \neq x_j$; that is, that they don't share a common divisor of degree 1. If $d_i(x) = a_i x + b_i$ is a divisor of $q_i(x)$, then $q_i(x) = q'(x)(a_i x + b_i)$ for some polynomial $q'(x)$. But since $q_i(x)$ is of degree 1, $q'(x)$ must be of degree 0 and hence a constant, so $d_i(x)$ must be a constant multiple of $q_i(x)$. Similarly, any degree 1 divisor d_j of $q_j(x)$ must be a constant multiple of $q_j(x)$, and if $x_i \neq x_j$, then none of these multiples overlap, so $q_i(x)$ and $q_j(x)$ are coprime.

From our result in part (d), the congruences $(1')$-(k') assert that we are looking for a polynomial $p(x)$ such that $p(x_i) = y_i$. The CRT then establishes the existence of $p(x)$, and that it is unique modulo a degree k polynomial. That is, $p(x)$ is unique if its degree is at most $k-1$. Lagrange interpolation finds $p(x)$.

3 Old secrets, new secrets

In order to share a secret number s, Alice distributed the values $(1, p(1)), (2, p(2)), \ldots, (n+1, p(n+1))$ of a degree n polynomial p with her friends $\text{Bob}_1, \ldots, \text{Bob}_{n+1}$. As usual, she chose p such that $p(0) = s$. Bob_1 through Bob_{n+1} now gather to jointly discover the secret. Suppose that for some reason Bob_1 already knows s, and wants to play a joke on $\text{Bob}_2, \ldots, \text{Bob}_{n+1}$, making them believe that the secret is in fact some fixed $s' \neq s$. How can he achieve this?

Solution:

We know that in order to discover s, the Bobs would compute

$$s = y_1 \Delta_1(0) + \sum_{k=2}^{n+1} y_k \Delta_k(0), \tag{1}$$

where $y_i = p(i)$. Bob_1 now wants to change his value y_1 to some y_1', so that

$$s' = y_1' \Delta_1(0) + \sum_{k=2}^{n+1} y_k \Delta_k(0). \tag{2}$$

Subtracting (1) from (2) and solving for y_1', we see that

$$y_1' = (\Delta_1(0))^{-1} \left(s' - s\right) + y_1,$$

where $(\Delta_1(0))^{-1}$ exists, because $\deg \Delta_1(x) = n$ with its n roots at $2, \ldots, n+1$ (so $\Delta_1(0) \neq 0$).

4 Berlekamp-Welch for General Errors

Suppose that Hector wants to send you a length $n = 3$ message, m_0, m_1, m_2, with the possibility for $k = 1$ error. For all parts of this problem, we will work mod 11, so we can encode 11 letters as shown below:

A	B	C	D	E	F	G	H	I	J	K
0	1	2	3	4	5	6	7	8	9	10

Hector encodes the message by finding the degree ≤ 2 polynomial $P(x)$ that passes through $(0,m_0)$, $(1,m_1)$, and $(2,m_2)$, and then sends you the five packets $P(0),P(1),P(2),P(3),P(4)$ over a noisy channel. The message you receive is

$$\text{DHACK} \Rightarrow 3,7,0,2,10 = r_0,r_1,r_2,r_3,r_4$$

which could have up to 1 error.

(a) First, let's locate the error, using an error-locating polynomial $E(x)$. Let $Q(x) = P(x)E(x)$. Recall that

$$Q(i) = P(i)E(i) = r_i E(i), \quad \text{for} \quad 0 \leq i < n+2k.$$

What is the degree of $E(x)$? What is the degree of $Q(x)$? Using the relation above, write out the form of $E(x)$ and $Q(x)$ in terms of the unknown coefficients, and then a system of equations to find both these polynomials.

(b) Solve for $Q(x)$ and $E(x)$. Where is the error located?

(c) Finally, what is $P(x)$? Use $P(x)$ to determine the original message that Hector wanted to send.

Solution:

(a) The degree of $E(x)$ will be 1, since there is at most 1 error. The degree of $Q(x)$ will be 3, since $P(x)$ is of degree 2. $E(x)$ will have the form $E(x) = x+e$, and $Q(x)$ will have the form $Q(x) = ax^3+bx^2+cx+d$. We can write out a system of equations to solve for these 5 variables:

$$\begin{aligned}
d &= 3(0+e) \\
a+b+c+d &= 7(1+e) \\
8a+4b+2c+d &= 0(2+e) \\
27a+9b+3c+d &= 2(3+e) \\
64a+16b+4c+d &= 10(4+e)
\end{aligned}$$

Since we are working mod 11, this is equivalent to:

$$\begin{aligned}
d &= 3e \\
a+b+c+d &= 7+7e \\
8a+4b+2c+d &= 0 \\
5a+9b+3c+d &= 6+2e \\
9a+5b+4c+d &= 7+10e
\end{aligned}$$

(b) Solving this system of linear equations we get

$$Q(x) = 3x^3+6x^2+5x+8.$$

Plugging this into the first equation (for example), we see that:

$$d = 8 = 3e \quad \Rightarrow \quad e = 8 \cdot 4 = 32 \equiv 10 \bmod 11$$

This means that

$$E(x) = x + 10 \equiv x - 1 \pmod{11}.$$

Therefore, the error occurred at $x = 1$ (so the second number sent in this case).

(c) Using polynomial division, we divide $Q(x) = 3x^3 + 6x^2 + 5x + 8$ by $E(x) = x - 1$:

$$P(x) = 3x^2 + 9x + 3$$

Then, $P(1) = 3 + 9 + 3 = 15 \equiv 4 \bmod 11$. This means that our original message was

$$3, 4, 0 \quad \Rightarrow \quad \text{DEA}.$$

Note: In Season 4 of Breaking Bad, Hector Salamanca (who cannot speak), uses a bell to spell out "DEA" (Drug Enforcement Agency).

5 Error-Detecting Codes

Suppose Alice wants to transmit a message of n symbols, so that Bob is able to *detect* rather than *correct* any errors that have occured on the way. That is, Alice wants to find an encoding so that Bob, upon receiving the code, is able to either

(I) tell that there are no errors and decode the message, or

(II) realize that the transmitted code contains at least one error, and throw away the message.

Assuming that we are guaranteed a maximum of k errors, how should Alice extend her message (i.e. by how many symbols should she extend the message, and how should she choose these symbols)? You may assume that we work in GF(p) for very large prime p. Show that your scheme works, and that adding any lesser number of symbols is not good enough.

Solution:

Since k bits can break, it seems reasonable to extend our message by k symbols for a total of $n+k$. And indeed, we show that this works: Let Alice generate her message $y_0, \ldots y_{n-1}$ of length n by constructing the unique polynomial f of degree $\leq n-1$ that passes through (i, y_i) for $i \in \{0, \ldots, n-1\}$, and add the k extra symbols $y_j = f(j)$, where $j \in \{n, \ldots, n+k-1\}$. Now Bob receives the message $r_i, i \in \{0, \ldots, n+k-1\}$, upon which he interpolates the unique degree $\leq n-1$ polynomial g that passes through the points $(0, r_0), \ldots, (n-1, r_{n-1})$. We claim that the message is corrupted if and only if $g(i) \neq r_i$ for some $i \in \{n, \ldots, n+k-1\}$.

The backward direction becomes clear when stated as its contrapositive: If the message contains no error, then $g(i)$ and $f(i)$ coincide on all of n points $\{0, \ldots, n-1\}$. Since they are both of degree $n-1$, they must be the same polynomial and hence $g(i) = f(i) = r_i$ for all i.

Let us prove the forward direction: Since we know that at most k errors occured, there must exist a subset $A \subset \{0,\ldots,n+k-1\}$ of size n on which $r_i = y_i$. Now either

1. $A = \{0,\ldots,n-1\}$, in which case $g = f$ and at least one error must have occured for some $j_0 \in \{n,\ldots,n+k-1\}$. But then $r_{j_0} \neq y_{j_0} = f(j_0) = g(j_0)$, which is what we wanted to show.
2. Or at least one error occured for an index $i \in \{0,\ldots,n-1\}$ in which case $g \neq f$. But since g and f are of degree $n-1$ and $\mid A \mid = n$, f and g cannot take the same values on A, so there must be some element $j_0 \in A$, $j_0 \in \{n,\ldots,n+k-1\}$ for which $g(j_0) \neq f(j_0) = y_{j_0} = r_{j_0}$.

Lastly, we need to show that our algorithm doesn't work if Alice extends her message by less than k symbols, which we can do by crafting a counterexample: Assume Alice sends $m < n+k-1$ symbols in the same fashion as above, then we may corrupt $y_{n-1},\ldots y_{m-1}$ by setting $r_{n-1} \neq y_{n-1}$ and $r_j = h(j)$ for $j \in \{n,\ldots,m-1\}$, where h is the unique polynomial of degree $\leq n-1$ passing through $(0,y_0),\ldots(n-2,y_{n-2}),(n-1,r_{n-1})$. Since Bob is going to reconstruct $g = h$, $g(j) = r_j$ for all $j \in \{n,\ldots,m-1\}$ and he will not notice the corruption.