CS 70 Discrete Mathematics and Probability Theory
Spring 2026 Sinclair, Song

HW 05

1 Equivalent Polynomials

Note 7
Note 8

This problem is about polynomials with coefficients in GF(p) for some prime $p \in \mathbb{N}$. We say that two such polynomials f and g are *equivalent* if $f(x) \equiv g(x) \pmod{p}$ for every $x \in \text{GF}(p)$.

(a) Show that $f(x) = x^{p-1}$ and $g(x) = 1$ are **not** equivalent polynomials under GF(p).

(b) Use Fermat's Little Theorem to find a polynomial with degree strictly less than 13 that is equivalent to $f(x) = x^{13}$ over GF(13); then find a polynomial with degree strictly less than 7 that is equivalent to $g(x) = 2x^{74} + 6x^7 + 3$ over GF(7).

(c) In GF(p), prove that whenever $f(x)$ has degree $\geq p$, it is equivalent to some polynomial $\tilde{f}(x)$ with degree $< p$.

Solution:

(a) For f and g to be equivalent, they must satisfy $f(x) \equiv g(x) \pmod{p}$ for all values of x, including zero. But $f(0) \equiv 0 \pmod{p}$ and $g(0) \equiv 1 \pmod{p}$, so they are not equivalent.

(b) Fermat's Little Theorem says that for any nonzero integer a and any prime number p, $a^{p-1} \equiv 1 \bmod p$. We're allowed to multiply through by a, so the theorem is equivalent to saying that $a^p \equiv a \bmod p$; note that this is true even when $a = 0$, since in that case we just have $0^p \equiv 0 \pmod{p}$.

The problem asks for a polynomial $\tilde{f}(x)$, different from $f(x)$, with the property that $\tilde{f}(a) \equiv a^{13} \bmod 13$ for any integer a. Directly using the theorem, $\tilde{f}(x) = x$ will work. We can do something similar with $g(x) = 2x^{74} + 6x^7 + 3$ modulo 7; since $x^7 \equiv x \pmod{7}$, we repeatedly substitute x^7 with x, effectively reducing the exponent by 6. We can only do this as long as the exponent remains greater than or equal to 7, so we end up with $\tilde{g}(x) = 2x^2 + 6x + 3$.

(c) One proof uses Fermat's Little Theorem. As a warm-up, let $d \geq p$; we'll find a polynomial equivalent to x^d. For any integer, we know

$$\begin{aligned} a^d &= a^{d-p}a^p \\ &\equiv a^{d-p}a \pmod{p} \\ &\equiv a^{d-p+1} \pmod{p}. \end{aligned}$$

In other words x^d is equivalent to the polynomial $x^{d-(p-1)}$. If $d-(p-1) \geq q$, we can show in the same way that x^d is equivalent to $x^{d-2(p-1)}$. Since we subtract $p-1$ every time, the sequence $d, d-(p-1), d-2(p-1), \ldots$ must eventually be smaller than p. Now if $f(x)$ is

any polynomial with degree $\geq p$, we can apply this same trick to every x^k that appears for which $k \geq p$.

Another proof uses Lagrange interpolation. Let $f(x)$ have degree $\geq p$. By Lagrange interpolation, there is a unique polynomial $\tilde{f}(x)$ of degree at most $p-1$ passing through the points $(0, f(0)), (1, f(1)), (2, f(2)), ..., (p-1, f(p-1))$, and we know it must be equivalent to $f(x)$ because f also passes through the same p points.

2 Lagrange's Residents

Note 8

A group of humans has settled at the Earth–Moon L5 point, a Lagrange Point near earth. They have a message for their friends on Earth, and its your job to decode it.

A four packet message is sent using a degree 3 polynomial $P(x)$, where $P(0) = m_1$, $P(1) = m_2$, $P(2) = m_3$, and, $P(3) = m_4$. $P(4)$ and $P(5)$ are also sent.

Unfortunately, the channel lost $P(0)$ and $P(3)$, so the earthlings only received:

$(1,3), (2,7), (4,-90), (5,-335)$

Using Lagrange interpolation and a graphical calculator (eg. Desmos), recover $P(0)$ and $P(3)$ to unlock the space explorers' message.

Solution:

(a)
$$\Delta_1(x) = \frac{(x-2)(x-4)(x-5)}{(1-2)(1-4)(1-5)} = \frac{(x-2)(x-4)(x-5)}{-12}$$

(b)
$$\Delta_2(x) = \frac{(x-1)(x-4)(x-5)}{(2-1)(2-4)(2-5)} = \frac{(x-1)(x-4)(x-5)}{6}$$

(c)
$$\Delta_4(x) = \frac{(x-1)(x-2)(x-5)}{(4-1)(4-2)(4-5)} = \frac{(x-1)(x-2)(x-5)}{-6}$$

(d)
$$\Delta_5(x) = \frac{(x-1)(x-2)(x-4)}{(5-1)(5-2)(5-4)} = \frac{(x-1)(x-2)(x-4)}{12}$$

(e)
$$\begin{aligned} p(x) &= 3 \cdot \Delta_1(x) + 7 \cdot \Delta_2(x) - 90 \cdot \Delta_4(x) - 335 \cdot \Delta_5(x) \\ &= \frac{-24x^3 + 133x^2 - 223x + 120}{2} \end{aligned}$$

(f)
$$p(0) = 60$$

(g)

$$p(3) = 0$$

60 is the ASCII code for <.

<3 70

Turns out even space explorers enjoy discrete maths!

3 Cal Football's Secrets

Note 8

After a tough defeat, the Cal Football team has created a new set of top-secret plays. They're worried about leaks, however, and have asked you to devise a secret sharing scheme to protect their strategy.

The team has one head coach, six assistant coaches, and thirty two players. All plays are encrypted and we know that:

- The head coach along with one assistant coach should be able to access the plays.
- The majority (4+) of assistant coaches should be able to access the plays.
- All of the players should be able to access the plays together.
- Sixteen players and two assistant coaches should be able to access the plays.

Design a secret sharing scheme to make this work.

Solution: We will create polynomials that satisfy each condition and distribute points to coaches / players appropriately.

$S_i(x)$ will represent a polynomial with a y-intercept that contains the secret code for the plays.

First, we will create a polynomial that the coaches can use to get the secret code. $S_1(x)$ will be a degree 3 polynomial that coaches can use to access the secret code. The head coach will get three points on $S_1(x)$ and each assistant coach will get one point on $S_1(x)$. This allows our scheme to satisfy the first two requirements.

Next, we will create a polynomial that the players can use to get the secret code. $S_2(x)$ will be a degree 31 polynomial. Each player will get one point on $S_2(x)$. This allows our scheme to satisfy the third requirement.

Lastly, we need a way for two coaches and sixteen players to access the secret code. $S_3(x)$ will be a degree 1 polynomial. The players will be able to access one point on this polynomial $(1, a)$. The assistant coaches will be able to access another point on this polynomial $(2, b)$.

For the players to be able to access their point on $S_3(x)$, we will create $P_1(x)$, a degree 15 polynomial. $P_1(0) = a$. Each player will get one point and 16 players will be able to recover the polynomial, and therefore, the y-intercept (which stores their point, a).

For the assistant coaches to be able to access their point on $S_3(x)$, we will create $P_2(x)$, a degree 1 polynomial. $P_2(0) = b$. Each assistant coaches will get one point and 2 coaches will be able to recover the polynomial, and therefore, the y-intercept (which stores their point, b).

When the players have recovered a and the coaches have recovered b, they can recover $S_3(x)$ together using their two points. This allows our scheme to satisfy the final requirement.

4 Alice and Bob

Note 8
Note 9

(a) Alice decides that instead of encoding her message as the values of a polynomial, she will encode her message as the coefficients of a degree 2 polynomial $P(x)$. For her message $[m_1, m_2, m_3]$, she creates the polynomial $P(x) = m_1x^2 + m_2x + m_3$ and sends the five packets $(0, P(0))$, $(1, P(1))$, $(2, P(2))$, $(3, P(3))$, and $(4, P(4))$ to Bob. However, one of the packet y-values (one of the $P(i)$ terms; the second attribute in the pair) is changed by Eve before it reaches Bob. If Bob receives

$$(0,1), (1,3), (2,0), (3,1), (4,0)$$

and knows Alice's encoding scheme and that Eve changed one of the packets, can he recover the original message? If so, find it as well as the x-value of the packet that Eve changed. If he can't, explain why. Work in mod 7. Also, feel free to use a calculator or online systems of equations solver, but make sure it can work under mod 7.

(b) Bob gets tired of decoding degree 2 polynomials. He convinces Alice to encode her messages on a degree 1 polynomial. Alice, just to be safe, continues to send 5 points on her polynomial even though it is only degree 1. She makes sure to choose her message so that it can be encoded on a degree 1 polynomial. However, Eve changes two of the packets. Bob receives $(0,5)$, $(1,7)$, $(2,x)$, $(3,5)$, $(4,0)$. If Alice sent $(0,5)$, $(1,7)$, $(2,9)$, $(3,-2)$, $(4,0)$, for what values of x will Bob not uniquely be able to determine Alice's message? Assume that Bob knows Eve changed two packets. Work in mod 13. Again, feel free to use a calculator or graphing calculator software.

Hint: Observe that since Bob knows that Eve changed two packets, he's looking for a polynomial that passes through at least 3 of the given points. Think about what must happen in order for Bob to be unable to uniquely identify the original polynomial.

(c) Alice wants to send a length n message to Bob. There are two communication channels available to her: Channel X and Channel Y. Only 6 packets can be sent through channel X. Similarly, Channel Y will only deliver 6 packets, but it will also corrupt (change the value) of one of the delivered packets. Using each of the two channels once, what is the largest message length n Alice can send such that Bob so that he can always reconstruct the message?

Solution:

(a) We can use Berlekamp and Welch. We have: $Q(x) = P(x)E(x)$. $E(x)$ has degree 1 since we know we have at most 1 error. $Q(x)$ is degree 3 since $P(x)$ is degree 2. We can write a system of linear equations and solve for the coefficients of $Q(x) = ax^3 + bx^2 + cx + d$ and

$E(x) = (x-e)$ by writing the equation $Q(i) = r_i \cdot E(i)$ for $0 \leq i \leq 4$, where r_i is the ith received point.

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

Since we are working in mod 7, this is equivalent to:

$$\begin{aligned} d &= -e \\ a+b+c+d &= 3-3e \\ a+4b+2c+d &= 0 \\ 6a+2b+3c+d &= 3-e \\ a+2b+4c+d &= 0 \end{aligned}$$

Solving yields:

$$Q(x) = x^3+5x^2+5x+4, E(x) = x-3$$

To find $P(x)$ we divide $Q(x)$ by $E(x)$ and get $P(x) = x^2+x+1$. So Alice's message is $m_1 = 1, m_2 = 1, m_3 = 1$. The x-value of the packet Eve changed is 3.

Alternative solution: Since we have 5 points, we have to find a polynomial of degree 2 that goes through 4 of those points. The point that the polynomial does not go through will be the packet that Eve changed. Since 3 points uniquely determine a polynomial of degree 2, we can pick 3 points and check if it goes through a 4th point. (It may be the case that we need to try all sets of 3 points.)

We pick the points $(1,3),(2,0),(4,0)$. Lagrange interpolation can be used to create the polynomial but we can see that for the polynomial that goes through these 3 points, it has 0s at $x = 2$ and $x = 4$. Thus the polynomial is $k(x-2)(x-4) = k(x^2-6x+8) \pmod 7 \equiv k(x^2+x+1) \pmod 7$. We find $k \equiv 1$ by plugging in the point $(1,3)$, so our polynomial is x^2+x+1. We then check to see if this polynomial goes through one of the 2 points that we didn't use. Plugging in 0 for x, we get 1. The packet that Eve changed is the point that our polynomial does not go through which has x-value 3. Alice's original message was $m_1 = 1, m_2 = 1, m_3 = 1$.

(b) Since Bob knows that Eve changed 2 of the points, the 3 remaining points will still be on the degree 1 polynomial that Alice encoded her message on. Thus if Bob can find a degree 1 polynomial that passes through at least 3 of the points that he receives, he will be able to uniquely recover Eve's message. The only time that Bob cannot uniquely determine Alice's message is if there are 2 polynomials with degree 1 that pass through 3 of the 5 points that he receives. Since we are working with degree 1 polynomials, we can plot the points that Bob receives and then see which values of x will cause 2 sets of 3 points to fall on a line.

$(0,5),(1,7),(4,0)$ already fall on a line. If $x = 6$, $(1,7),(2,6),(3,5)$ also falls on a line. If $x = 5$, $(0,5),(2,5),(3,5)$ also falls on a line. If $x = 9$, $(0,5),(2,9),(4,0)$ falls on the original line, so here Bob can decode the message. If $x = 10$, $(2,10),(3,5),(4,0)$ also falls on a line. So if $x = 6,5,10$, Bob will not be able to uniquely determine Alice's message.

(c) Channel X can send 6 packets, so the first 6 characters of the message can be send through Channel X. Channel Y can send 6 packets, but 1 will be corrupted, thus only a message of length 4 can be sent. Thus, a total of $m = 6+4 = 10$ characters can effectively sent.

5 Error-Correcting Codes

Note 9

(a) Recall from class the error-correcting code for erasure errors, which protects against up to k lost packets by sending a total of $n+k$ packets (where n is the number of packets in the original message). Often the number of packets lost is not some fixed number k, but rather a *fraction* of the number of packets sent. Suppose we wish to protect against a fraction α of lost packets (where $0 < \alpha < 1$). At least how many packets do we need to send (as a function of n and α)?

(b) Repeat part (a) for the case of general errors.

Solution:

(a) Suppose we send a total of m packets (where m is to be determined). Since at most a fraction α of these are lost, the number of packets received is at least $(1-\alpha)m$. But in order to reconstruct the polynomial used in transmission, we need at least n packets. Hence it is sufficient to have $(1-\alpha)m \geq n$, which can be rearranged to give $m \geq n/(1-\alpha)$.

(b) Suppose we send a total of $m = n+2k$ packets, where k is the number of errors we can guard against. The number of corrupted packets is at most αm, so we need $k \geq \alpha m$. Hence $m \geq n+2\alpha m$. Rearranging gives $m \geq n/(1-2\alpha)$.

Note: Recovery in this case is impossible if $\alpha \geq 1/2$.