I worked alone without getting any help, except asking questions on Piazza and reading the Notes of this course.

1 Polynomial Practice

(a)

(i) At least: **0**. At most: $\mathbf{max(deg(}f\mathbf{),\ deg(}g\mathbf{))}$

Since we can't guarantee that the resulting polynomial $f+g$ always has a real root, so the least number of roots is 0.

WLOG, let $\deg(f) \leq \deg(g)$. Consider $f+g$, since it is non-zero, so it is a polynomial whose degree is at most $\deg(g)$, and since a polynomial with degree d has at most d roots, so $f+g$ has at most $\deg(g)$ roots. Vice versa for the case of $\deg(f) > \deg(g)$, which gives us that the polynomial $f+g$ has at most $\max(\deg(f), \deg(g))$ roots.

(ii) At least: **0**. At most: $\mathbf{deg(}f\mathbf{)\ +\ deg(}g\mathbf{)}$

Consider $f = g = x^2+1$, so f, g have no roots, so $f \cdot g$ also have no roots, which implies that $f \cdot g$ is not guaranteed to have any roots, so the least number of roots is 0.

Let $a = \deg(f)$ and $b = \deg(g)$. Since $x^a \cdot x^b = x^{a+b}$, so the degree of the polynomial $f \cdot g$ is $a+b$, which is $\deg(f) + \deg(g)$. Again, using the property of polynomials, so $f \cdot g$ has at most $\deg(f) + \deg(g)$ roots.

(iii) At least: **0**. At most: $\mathbf{deg(}f\mathbf{)\ -\ deg(}g\mathbf{)}$

Again, we can't guarantee that f/g always has a root. For example, let $f = x^2+1$ and $g = 1$, so $f/g = x^2+1$ does not have a root. So, the least number of roots is 0.

Let $a = \deg(f)$ and $b = \deg(g)$. Since $\frac{x^a}{x^b} = x^{a-b}$ and that we are given that f/g is a polynomial, so the degree of the polynomial f/g is $a-b$, which is $\deg(f) - \deg(g)$. Again, using the property of polynomials, so f/g has at most $\deg(f) - \deg(g)$ roots.

(b)

(i) No, it isn't.

We proceed by providing a counterexample. Consider $p = 2,\ f(x) = x,\ g(x) = x+1$. For all $x \in GF(p)$, namely, $x_1 = 0$ or $x_2 = 1$, we have that when $x_1 = 0$, $f \cdot g(x) = 0 \cdot 1 = 0$; when $x_2 = 1$, $f \cdot g(x) = 1 \cdot 2 \equiv 0 \pmod 2$, which gives us that $f \cdot g = 0$.

Yet, when $x = 1$, $f(x) = 1 \neq 0$; when $x = 0$, $g(x) = 1 \neq 0$, which implies that $f \neq 0$ and $g \neq 0$. Thus, this is a counterexample, so if f, g are polynomials over $GF(p)$ and $f \cdot g = 0$, it isn't necessarily true that either $f = 0$ or $g = 0$.

(ii) Direct Proof.

Proof. We proceed by a direct proof. Suppose f be a polynomial over $GF(p)$, and that $\deg(f) \geq p$.

Now, let $f(i) = y_i\ \forall i \in \{0, 1, ..., p-1\}$, which means that we have p pairs $(0, y_0), (1, y_1), ..., (p-1, y_{p-1}$ with all the x_i distinct. Thus, by Property 2 of polynomials, so there is a unique polynomial $h(x)$ of degree (at most) $p-1$, which means that h is a polynomial with $\deg(h) \leq p-1 < p$ such that $f(x) = h(x)$ for all $x \in \{0, 1, ..., p-1\}$, as desired. Thus, such a polynomial h exists.

Q.E.D.

(iii) p^d

For a polynomial f with degree $d < p$, so $f(x) = a_d x^d + a_{d-1}x^{d-1} + \cdots + a_1 x + a_0$, where

$a_i \in \{0, 1, ..., p-1\} \; \forall \; i \in \mathbb{N}, i \leq d$.

Then, the only constraint on f is that $f(0) = a_0 = a$ where $a \in \{0, 1, ..., p-1\}$. So only one coefficient, a_0, of f is set, which means that $\forall \; 1 \leq i \leq d, i \in \mathbb{N}$, we have that a_i could be any value in the set $\{0, 1, ..., p-1\}$. Thus, there are p possible values for each coefficient a_i of f, and there are d-many coefficients we could assign values arbitrarily (as long as they are in $\{0, 1, ..., p-1\}$), which implies that there are p^d-many such polynomials f.

(c) $f = 4x^2 + 1$. There are **25** such polynomial.

Part (1) Finding one such polynomial f

We first use Lagrange interpolation to find one such f. So, we have that $8^{-1} \equiv 2 \pmod 5$ with $8 \cdot 2 = 16 = 5 \cdot 3 + 1$, and similarly, $(-4)^{-1} \equiv 1 \pmod 5$ with $-4 \cdot 1 = -4 = 5 \cdot (-1) + 1$.

Thus, using Lagrange interpolation, we have:

$\Delta_1(x) = \frac{(x-2)(x-4)}{(0-2)(0-4)} = \frac{x^2-6x+8}{8} = 2(x^2 - 6x + 8) = 2x^2 - 12x + 16$

$\Delta_2(x) = \frac{(x-0)(x-4)}{(2-0)(2-4)} = \frac{x^2-4x}{-4} = 1(x^2 - 4x) = x^2 - 4x$

$\Delta_3(x) = \frac{(x-0)(x-2)}{(4-0)(4-2)} = \frac{x^2-2x}{8} = 2(x^2 - 2x) = 2x^2 - 4x$

Thus, the polynomial $f(x)$ is therefore given by:

$f(x) = 1 \cdot \Delta_1(x) + 2 \cdot \Delta_2(x) + 0 \cdot \Delta_3(x) = 1(2x^2 - 12x + 16) + 2(x^2 - 4x) + 0(2x^2 - 4x) = (2+2)x^2 + (-12-8)x + 16 = 4x^2 - 20x + 16$

Since f is a polynomial over $GF(5)$, and that we have $-20 = 5 \cdot (-4) + 0, 16 = 5 \cdot 3 + 1$, which gives us that $-20 \equiv 0 \pmod 5, 16 \equiv 1 \pmod 5$, so we have that $f(x) = 4x^2 + 1$.

Part (2) Showing that there are 25 such polynomials

Since we define f over $GF(5)$, so the value of x is also confined within $\{0, 1, 2, 3, 4\}$. Now, since $f(0) = 1, f(2) = 2, f(4) = 0$, so we only have $x = 1, x = 3$ that we can arbitrarily assign values to, with $f(x)$ still confined within $\{0, 1, 2, 3, 4\}$. So, we have $5 \cdot 5 = 25$ different combinations of value assignments to $f(x), 0 \leq x \leq 4$. Then, by the property of polynomials, we can construct a unique polynomial based on each of these assignments, which means that there are 25 different such polynomials.

2 The CRT and Lagrange Interpolation

(a) Direct Proof

Proof. We proceed by a direct proof for both statements given $k = 2$. First, we show that: by assumption, n_1, n_2 are coprime, which is equivalent to $\gcd(n_1, n_2) = 1$, so we can write $k_1n_1 + k_2n_2 = 1$ for some $k_1, k_2 \in \mathbb{Z}$.

Part 1: Given that $a_1 = 1, a_2 = 0$, so we have that $x_1 \equiv 1 \pmod{n_1}, x_1 \equiv 0 \pmod{n_2}$. Now, consider $x_1 = k_2n_2$. Since $n_2, k_2 \in \mathbb{Z}$, so $x_1 \in \mathbb{Z}$ and $x_1 \equiv 0 \pmod{n_2}$. Since we also have $x_1 = 1 - k_1n_1 = (-k_1) \cdot (n_1) + 1$, with $-k_1 \in \mathbb{Z}$, so $x_1 \equiv 1 \pmod{n_1}$, which implies that $x_1 = k_2n_2$ is a valid solution to the first situation.

Part 2: Given that $a_1 = 0, a_2 = 1$, so we have that $x_2 \equiv 0 \pmod{n_1}, x_2 \equiv 1 \pmod{n_2}$. Now, consider $x_2 = k_1n_1 \in \mathbb{Z}$. Similarly, we have that $x_2 \equiv 0 \pmod{n_1}$, and with $x_2 = 1 - k_2n_2 = (-k_2) \cdot (n_2) + 1$, so similar to Part 1 above, $x_2 \equiv 1 \pmod{n_2}$, which gives that $x_2 = k_1n_1$ is a valid solution to the first situation.

Q.E.D.

(b) Direct Proof

Proof. We proceed by a direct proof for both statements where we still write $k_1n_1 + k_2n_2 = 1$ for some $k_1, k_2 \in \mathbb{Z}$.

For any given a_1, a_2, consider $x = a_1k_2n_2 + a_2k_1n_1$. So, $x = a_1(1 - k_1n_1) + a_2k_1n_1 = a_1 + (-a_1k_1 + a_2k_1)n_1$. Since $a_1, k_1, a_2, k_2 \in \mathbb{Z}$, so $-a_1k_1 + a_2k_1 \in \mathbb{Z}$, which gives us that $x \equiv a_1 \pmod{n_1}$. Similarly, $x = a_1k_2n_2 + a_2k_1n_1 = a_1k_2n_2 + a_2(1 - k_2n_2) = a_2 + (a_1k_2 - a_2k_2)n_2$ and since $a_1, k_1, a_2, k_2 \in \mathbb{Z}$, so $a_1k_2 - a_2k_2 \in \mathbb{Z}$, so we have $x \equiv a_2 \pmod{n_2}$. Thus, there exists at least one solutoin x to (1) and (2) for any a_1, a_2.

For any two solutions x', x^* to (1) and (2) with given a_1, a_2, we have that $x' \equiv x^* \equiv a_1 \pmod{n_1}$ and $x' \equiv x^* \equiv a_2 \pmod{n_2}$. So, $x' - x^* \equiv 0 \pmod{n_1}$ and $x' - x^* \equiv 0 \pmod{n_2}$. Since given that $\gcd(n_1, n_2) = 1$, using previous homework results, we have that $x' - x^* \equiv 0 \pmod{n_1n_2}$. Thus, $x' \equiv x^* \pmod{n_1n_2}$, which implies that all possible solutions are equivalent $\pmod{n_1n_2}$, as desired.

Q.E.D.

(c) Direct Proof

Proof. We proceed by a direct proof for both statements.

Since for all $i \neq j$, it is given that n_i, n_j are coprime, so we can repeat the process we described and proved in part (b) by solving two equations at a time, which will always yield us a solution x. In other words, $\exists$ a solution x to (1)-(k).

Again, we can show that this solution is unique $\pmod{n_1n_2 \cdots n_k}$ by showing that for any two solutions x', x^* to (1)-(k), we have $x' - x^* \equiv 0 \pmod{n_1n_2 \cdots n_k}$ by repetitively using the strategy presented in part (b), given that for all $i \neq j$, it is given that n_i, n_j are coprime. Thus, $x' \equiv x^* \pmod{n_1n_2 \cdots n_k}$, which implies that the solution x is unique $\pmod{n_1n_2 \cdots n_k}$.

Q.E.D.

(d) Definition see below; $p(1)$

Let $p(x) = k(x) \cdot q(x) + r(x)$ where $k(x), r(x)$ are also polynomials and $0 \leq \deg(r) < \deg(q)$. Mimicing the definition of a mod b for integers, we define polynomial mod, $p(x)$ mod $q(x)$, to be: $q(x) \equiv r(x) \pmod{p(x)}$ where $0 \leq \deg(r) < \deg(q)$.

Then, consider when $x = 1$, we have $p(x) - p(1) = p(1) - p(1) = 0$, which means that $x = 1$ is a root for $p(x) - p(1)$ where $p(1)$ is a constant that can be calculated, which implies that $\deg(p(1)) = 0 < 1 = \deg(x - 1)$. Since we also have that $p(x) - p(1) = 0 \equiv 0 \pmod{x - 1}$, so $p(x) \equiv p(1) \pmod{x - 1}$.

Thus, $p(x) \bmod (x-1)$ is $p(1)$.

(e) Direct Proof; connection to Lagrange interpolation explained below.

Proof. We proceed by a direct proof for both statements. We claim that each of the $x - x_i$ are pairwise coprime given the x_i are pairwise distinct.

We proceed by a proof by contradiction to prove the above claim. Assume that for some two polynomials $x - x_m, x - x_n$ with $x_m \neq x_n$, they have a common divisor of degree 1, $ax + b$ and $a \neq 0$. Let R be the assertion that $x_m \neq x_n$ and let $x - x_m = k_m(ax+b), x - x_n = k_n(ax+b)$ where $k_m, k_n \in \mathbb{R}$. So, $x - x_m = ak_m x + bk_m$ and $x - x_n = ak_n x + bk_n$, which gives us these four equations:

$$\begin{aligned} 1 &= ak_m \\ -x_m &= bk_m \\ 1 &= ak_n \\ -x_n &= bk_n \end{aligned}$$

So, we have $1 = ak_m = ak_n$. With $a \neq 0$, so $k_m = k_n$. Thus, the equations above gives us that $x_m = -bk_m = -bk_n = x_n$, which implies $\neg R$. So, $R \wedge \neg R$ holds, which gives the contradiction.

Thus, our claim is true that each of the $x - x_i$ are pairwise coprime. Then, since we're told that the CRT still holds when replacing x, a_i, n_i with polynomials and using the coprime definition, so the system of congruences given has a unique solution $\pmod{(x-x_1)\cdots(x-x_k)}$ whenever the x_i are pairwise distinct.

Now, this is very similar to our Lagrange interpolation, since the way we write the greatest common divison of the $(x - x_i)$'s corresponds to the first step of finding the polynomials Δ_i, and finding the actual solution $p(x)$ by multiplication corresponds to our step in CRT where we multiply the base solution by y_i for each corresponding factor. Therefore, using the CRT for polynomials to find $p(x)$ is an equivalent method to Lagrange interpolation. Thus, this is also another proof of why Lagrange interpolation works and why there's a unique solution $p(x)$ in the modular setting, correspondingly, $GF(p)$.

3 Old secrets, new secrets

Bob$_1$ can achieve this by giving out $p'(1) = \frac{s'-s}{n+1} + p(1)$ instead of his actual number $p(1)$ when the Bobs gather to jointly discover the secret.

Consider the Lagrange interpolation process the Bobs would use once they gather together:

$$\Delta_1(x) = \frac{(x-2)(x-3)\cdots(x-(n+1))}{(1-2)(1-3)\cdots(1-(n+1))}$$
$$\Delta_2(x) = \frac{(x-1)(x-3)(x-4)\cdots(x-(n+1))}{(2-1)(2-3)(2-4)\cdots(2-(n+1))}$$
$$\ldots \qquad \ldots$$
$$\Delta_n(x) = \frac{(x-1)(x-2)\cdots(x-(n-1))(x-(n+1))}{(n-1)(n-2)\cdots(n-(n-1))(n-(n+1))}$$
$$\Delta_{n+1}(x) = \frac{(x-1)(x-2)\cdots(x-n)}{(n+1-1)(n+1-2)\cdots(n+1-n)}$$

where the $\Delta_i(x) = \frac{(x-1)(x-2)\cdots(x-(i-1))(x-(i+1))\cdots(x-n)(x-(n+1))}{(i-1)(i-2)\cdots(i-(i-1))(i-(i+1))\cdots(i-n)(i-(n+1))}$

Thus, the original polynomial $f(x)$ is:

$$p(x) = p(1) \cdot \Delta_1(x) + p(2) \cdot \Delta_2(x) + p(3) \cdot \Delta_3(x) + \cdots + p(n+1) \cdot \Delta_{n+1}(x)$$

which gives that the secret $s = p(0) = p(1) \cdot \Delta_1(0) + p(2) \cdot \Delta_2(0) + \cdots + p(n+1) \cdot \Delta_{n+1}(0)$

Now, suppose Bob$_1$ wants to trick the other Bobs into believing that the secret is actually some fixed s'. Since the only thing he could lie about is $p(1)$, so let him say that he got the number $p'(1)$. Using Lagrange interpolation again, the Δ's would remain the same, and so new altered polynomial would be calculated as:

$$p'(x) = p'(1) \cdot \Delta_1(x) + p(2) \cdot \Delta_2(x) + p(3) \cdot \Delta_3(x) + \cdots + p(n+1) \cdot \Delta_{n+1}(x)$$

which means that the new secret $s' = p'(0) = p'(1) \cdot \Delta_1(0) + p(2) \cdot \Delta_2(0) + \cdots + p(n+1) \cdot \Delta_{n+1}(0)$

So, we have that:

$$s' - s = \Big(p'(1) \cdot \Delta_1(0) + p(2) \cdot \Delta_2(0) + \cdots + p(n+1) \cdot \Delta_{n+1}(0)\Big) - \Big(p(1) \cdot \Delta_1(0) + p(2) \cdot \Delta_2(0) + \cdots + p(n+1) \cdot \Delta_{n+1}(0)\Big) = p'(1) \cdot \Delta_1(0) - p(1) \cdot \Delta_1(0) = \big(p'(1) - p(1)\big) \cdot \Delta_1(0)$$

because we can cancel out all the other terms. Revisiting our definitions of $p'(1)$ and $p(1)$, so we have:

$$s' - s = \big(p'(1) - p(1)\big) \cdot \Delta_1(0)$$

Then, since $\Delta_1(0) = \frac{(0-2)(0-3)\cdots(0-(n+1))}{(1-2)(1-3)\cdots(1-(n+1))} = \frac{-2\cdot-3\cdots-n\cdot-(n+1)}{-1\cdot-2\cdot-3\cdots-(n-1)\cdot-n}$, which can be canceled out (as all terms are non-zero) into the form $\Delta_1(0) = \frac{-(n+1)}{-1} = n+1$. Thus, since $n+1 \neq 0$ by assumption, so we can divide both sides of the equation by $\Delta_1(0) = n+1$, which gives us that: $p'(1) - p(1) = \frac{s'-s}{n+1}$, which would then allow Bob$_1$ to calculate:

$$p'(1) = \frac{s'-s}{n+1} + p(1)$$

Thus, with s and $(n+1)$ being known by Bob$_1$ and s' being his goal in mind, so he can decide his fake value $p'(1)$ with the equation above and trick the other Bobs into believing that secret is actually some fixed s' instead of the original s.

4 Berlekamp-Welch for General Errors

(a) $\deg(E(x)) = 1$, $\deg(Q(x)) = 3$.

Since there's only one error, so $k = 1$ here, so the degree of $E(x) = k = 1$

Now, since Hector wants to send a length $n = 3$ message, so the degree of $Q(x) = n + k - 1 = 3$

Using the given relation, so we can write:

$E(x) = x - e_1$
$Q(x) = a_3x^3 + a_2x^2 + a_1x + a_0$

Then, since $Q(i) = r_iE(i)$ for $0 \leq i < 5$, and with $E(x) = x - e_i = x + b_0$, so we have that:

$Q(0) = r_0E(0)$, which gives: $a_0 = 3(0 + b_0) = 3b_0$
$Q(1) = r_1E(1)$, which gives: $a_3 + a_2 + a_1 + a_0 = 7(1 + b_0) = 7 + 7b_0$
$Q(2) = r_2E(2)$, which gives: $8a_3 + 4a_2 + 2a_1 + a_0 = 0(2 + b_0) = 0$
$Q(3) = r_3E(3)$, which gives: $27a_3 + 9a_2 + 3a_1 + a_0 = 2(3 + b_0) = 6 + 2b_0$
$Q(4) = r_4E(4)$, which gives: $64a_3 + 16a_2 + 4a_1 + a_0 = 10(4 + b_0) = 40 + 10b_0$

(b) $Q(x) = 3x^3 + 6x^2 + 5x + 8, E(x) = x - 1$; error located at position 1.

Suppose we're working over $GF(11)$, and with the system of equations derived above, we can translate them into:

$a_0 + 8b_0 = 0$
$a_3 + a_2 + a_1 + a_0 + 4b_0 = 7$
$8a_3 + 4a_2 + 2a_1 + a_0 = 0$
$5a_3 + 9a_2 + 3a_1 + a_0 + 9b_0 = 6$
$9a_3 + 5a_2 + 4a_1 + a_0 + b_0 = 7$

Since $2 \cdot 6 = 12 = 11 + 1$, so $2^{-1} \equiv 6 \pmod{11}$. Thus, solving this system of linear equations (with the help of Jupyter Notebook) would yield:

$a_3 = -\frac{5}{2} \equiv -5 \cdot 6 = -3 \cdot 11 + 3 \equiv 3 \pmod{11}$,
$a_2 = \frac{1}{2} \equiv 1 \cdot 6 = 6 \pmod{11}$,
$a_1 = 5 \pmod{11}$,
$a_0 = 8 \pmod{11}$,
$b_0 = -1 \pmod{11}$.

Thus, $Q(x) = 3x^3 + 6x^2 + 5x + 8$ and $E(x) = x - 1$; the location of this error is at position $e_1 = -b_0 = 1$.

(c) $P(x) = 3x^2 + 9x + 3$; message = "DEA".

Then, since we're working over $GF(11)$, with $Q(x) = 3x^3 + 6x^2 + 5x + 8$ and $E(x) = x - 1$, so we can calculate $P(x) = \frac{Q(x)}{E(x)} = \frac{3x^3+6x^2+5x+8}{x-1} = 3x^2 + 9x + 3$.

Since we noticed that the first character was corrupted as $e_1 = 1$, so we calculate $P(1) = 3+9+3 = 4 =$ "E", which means that

5 Error-Detecting Codes

Part (1) Scheme with extra k distinct packets/symbols

Below, I'll first present a brute force explanation of a scheme of sending an extra k packets works (assume a maximum of k errors). This means that Alice would be sending out values corresponding to $x = 1, 2, ..., n, n+1, ..., n+k$ where the first n packets represent her encoded message.

First, if any erasure error occurs, then Bob would easily detect it and throw away the message, so I'll only discuss the case where a maximum of k general errors occur.

Let's think of Bob as a person who loves doing Lagrange interpolation, so he picks out all the combinations of the $n+k$ packets he received from Alice, and uses Lagrange interpolation to calculate all the corresponding polynomials. Since there are a maximum of k errors, so there are at least n non-corrupted packets/symbols, and let's call one set of n non-corrupted symbols as S.

By assumption, Bob must have picked the set of n symbols, S, at some point during his Lagranging process, which would allow him to deduce a unique polynomial $P(x)$, and by the proerty of polynomials, this is the original polynomial that Alice used to extend her message. Now, if there are any corrupted packets, let one such error be at $x = e$. Since Alice used $P(x)$ as her encoding polynomimal, so the original symbol should be $P(e)$, and let the corrupted symbol be r_e such that $P(e) \neq r_e$.

Now, consider the set of n symbols, S', which consists of the first $n-1$ elements of S and the symbol corresponding to $x = e$, which is r_e. Again, by assumption, Bob must have picked the set of n symbols, S', at some point during his Lagranging process, from which he would deduce a unique polynomial $P'(x)$. Here, we have that $P'(e) = r_e \neq P(e)$. Thus, $P(x)$ must be different from $P'(x)$ by the property of polynomials, which implies that if any error exists, Bob would deduce at least two different polynomials (under $GF(p)$) at some time during his Lagrange interpolation process. In this situation, he can infer that the transmitted code contains at least one error, and throw away the message.

On the other hand, if no errors exist in the transmitted message, then every set of n symbols Bob picked would all let him deduce the same polynomial, which is the original $P(x)$ Alice used.

Thus, this scheme of adding k extra, distinct packets allows Bob to detect if the transmitted code contains at least one error.

(Proof of minimality on the next page)

Part (2) Minimality of k extra packets

We proceed to show with a counterexample that adding any lesser number of symbols is not good enough. Consider $k = 2, n = 2$. By my scheme, Alice need to extend her message by at least 2 packets. Now, suppose Alice only sends 1 extra packet.

Consider this situation: suppose Alice wants to send a simple message "bc", which corresponds to the 2 characters "1", "2" over a modem where $a = 0, b = 1, c = 2, d = 3, e = 4$.

Thus, Alice wishes to transmit $m_1 = 1, m_2 = 2$, and we can decide that the unique polynomials of degree $n - 1 = 1$ is $P(x) = x$ by simple observation (or via Lagrange interpolation). By assumption, Alice only sends one extra packet, which is $P(3) = 3$, so the encoded message sent by Alice is $c_1 = 1, c_2 = 2, c_3 = 3$.

Now, since the maximum number of errors is $k = 2$, suppose we have all 2 errors such that the last two symbols of the message got corrupted, and we have $r_1 = 1, r_2 = 3, r_3 = 5$. Then Bob, not knowing how many errors were present, would naturally deduce $P'(x) = 2x - 1$ without noticing any errors, since this is a correct degree 1 polynomial corresponding to the corrupted message Bob received. Thus, he would decode Alice's message as $m'_1 = 1, m'_2 = 3$, which would translate into "bd", which is different from the original message.

Thus, this is a counterexample where adding any lesser number of symbols than proposed my scheme would not necessarily help Bob detect an error.