This is the Title PhL 3.26.05

This stuff is fully handled in Appendix F, just archiving older notes here. It is the CRC detection theorems.

Not sure what this was for.

Little Lemma: $\text{Rem}(c/ab) = 0 \Rightarrow \text{Rem}(c/a) = 0$ and $\text{Rem}(c/b) = 0$

Proof: $[\text{Rem}(c/ab)] = 0 \Rightarrow c/ab = N \Rightarrow c/a = bN = N_1 \Rightarrow \text{Rem}(c/a) = 0$
$\Rightarrow c/b = aN = N_2 \Rightarrow \text{Rem}(c/b) = 0$

Corollary:

$\text{Rem}(c/a) \neq 0 \Rightarrow \text{Rem}(c/ab) \neq 0$

Fact: If g(x) divides $x^n - 1$ then g(x) cannot be written $g(x) = x^i f(x)$ for any $i > 1$

Proof: First a little obvious Lemma

Lemma: $\text{Rem}(c/ab) = 0 \Rightarrow \text{Rem}(c/a) = 0$ and $\text{Rem}(c/b) = 0$
or ab divides c $\Rightarrow$ a divides c and b divides c
or $ab \mid c \Rightarrow a \mid c$ and $b \mid c$

Proof: $[\text{Rem}(c/ab)] = 0 \Rightarrow c/ab = N \Rightarrow c/a = bN = N_1 \Rightarrow \text{Rem}(c/a) = 0$
$\Rightarrow c/b = aN = N_2 \Rightarrow \text{Rem}(c/b) = 0$

Thus, if g(x) divides $x^n - 1$ then both x^i and $g_1(x)$ must divide $x^n - 1$. But x^i is not a factor of $x^n - 1$, QED.

Fact: If g(x) divides $x^n - 1$ then g(x) cannot be written $g(x) = (x^i + x^j)f(x)$ for any $i \geq j > 0$.

Proof: As above, this requires that $x^n - 1$ be divisible by $(x^i + x^j) = x^j(x^{i-j} + 1)$, and that in turn requires that $x^n - 1$ be divisible by x^j but we showed above that is impossible.

Maybe this stuff is Appendix Material. Which one should come first

Appendix E on Existence of g(x) of any order theorem. I would refer to this right where cyclic codes are defined.

Appendix F on Cyclic Code Error Detection. I would refer to this from my little CRC section

---

Fact 1: If $g(0) \neq 0$, a cyclic code generated by g(x) over GF(p) detects all single-symbol errors.

Proof. Suppose C'(x) and C(x) differ in one symbol position. Then $E(x) \equiv C'(x)-C(x) = \alpha x^i$ where i = 0,1..n-1 and α is some non-zero element of GF(p). We need to show that in this case the syndrome will be non-zero so the error will then be detected. But :

$$s(x) = \text{Rem}[C'(x)/g(x)] = \text{Rem}[\{C(x) + E(x)\}/g(x)] = \text{Rem}[E(x)/g(x)] = \text{Rem}[\alpha\, x^i/g(x)]$$

where we used $\text{Rem}[C(x)/g(x)] = 0$. Now:

- For i = 0, $\text{Rem}[\alpha x^i/g(x)] = \text{Rem}[\alpha/g(x)] = \alpha \neq 0$. Note that $g(0) \neq 0$ rules out g(x) = constant ≠ 0.
- For i > 1, $\text{Rem}[\alpha x^i/g(x)] \neq 0$ unless $g(x) = \alpha x^n$ for $n \leq i$, but $g(0) \neq 0$ rules out such a g(x).

QED. Notice that it was not required that g(x) divide evenly into $x^n - 1$.

Fact 2: If g(x) = (x-1)f(x),a cyclic code generated by g(x) over GF(2) detects all odd-num-of-bit errors.

Proof: In this case, E(x) = C'(x) - C(x) is a polynomial with an odd number of terms. We must show that in this case the syndrome does not vanish so the error will be detected. But:

$$s(x) = \text{Rem}[E(x)/g(x)] = \text{Rem}[(\text{poly with odd of terms})/g(x)]$$

Assume that this remainder is 0. Then we would have

$$\text{poly with odd of terms}\ (x) = g(x)\, s(x) = (x-1)f(x)\, s(x)$$

Evaluate at x = 1 to get

$$\text{odd sum of 1's} = 1 = (1-1)f(1)s(1) = 0$$

Since $1 \neq 0$ we have a contradiction, so it must be that $s(x) \neq 0$ and the error is detected.

Notice that it was not required that g(x) divide evenly into $x^n - 1$.

Fact 3: If $g(0) \neq 0$ and the period of g(x) = n, a cyclic code generated by g(x) over GF(2) detects all double-bit errors and all single-bit errors.

Proof: We already know from Fact 1 that single-bit errors are detected. For double bit errors we have $E(x) = x^i + x^j$ for some $0 \leq i \leq j \leq n-1$ We need to show that the syndrome does not vanish in this case.

$$s(x) = \text{Rem}[E(x)/g(x)] = \text{Rem}[(x^i+x^j)/g(x)] = \text{Rem}[x^i(x^{j-i}+1)/g(x)] .$$

Since $g(0) \neq 0$, we know that g(x) has no factors of x so the x^i cancels nothing in g(x), so the only way for s(x) to be 0 is if

$$\text{Rem}[(x^{j-i}+1)/g(x)] = 0$$

which we rewrite in GF(2) as

$$\text{Rem}[(x^{j-i} - 1)/g(x)] = 0 \qquad (*)$$

The maximum value of j-i is (n-1)-(0) = n-1. If the period of g(x) is n, then (*) cannot be true, since the period s of g(x) is the smallest s such that (x^s-1) is divisible by g(x). Then we get $s(x) \neq 0$ and the double-bit error will be detected.

This proof requires that g(x) divide evenly into $x^n - 1$ *and* that n be the period of g(x).

Fact 4: If $g(0) \neq 0$ *and* $g(x) = (x-1)f(x)$ *and* the period of g(x) = n, a cyclic code generated by g(x) over GF(2) detects all single-bit, double-bit, triple-bit and other odd_number_of_bit errors.

Proof: Such a g(x) meets the criteria for Facts 1,2 and 3 so we can take the union of the detections of each.

Definition: A burst error of length (extent) r means any number of symbol errors occurring within a sequential set of powers x^s to x^{s+r-1}. For example, if a,b,c,d are non-zero elements of GF(p), the error pattern $E(x) = ax^3 + bx^5 + cx^9$ would be the error pattern for a burst error of length r = 7. Other E(x) of length 7: $ax^4 + bx^{10}$, $ax^2 + bx^4 + cx^6 + dx^8$. For GF(2) all coefficients are of course 1.

Fact 5: If $g(0) \neq 0$, a cyclic code generated by g(x) over GF(p) detects all burst errors of length n-k or smaller.

Proof: The error pattern can be written $E(x) = x^s f(x)$ where f(x) has degree r-1 = n-k-1. The syndrome is

$$s(x) = \text{Rem}[E(x)/g(x)] = \text{Rem}[x^s f(x)/g(x)] .$$

As before, x^s cancels nothing in g(x) if $g(0) \neq 0$, so the only way to get s(x) = 0 is if

$$\text{Rem}[f(x)/g(x)] = 0 .$$

But since g(x) has degree n-k and f(x) has degree n-k-1, this remainder is just $f(x) \neq 0$. QED.

Notice that it was not required that g(x) divide evenly into $x^n - 1$.

Fact 6: The cyclic code of Fact 5 detects all burst errors of length $r \leq n-k$. It does not detect all burst errors for r > n-k, but statistically it can detect a lot of them. For GF(2), here are the conclusions:

fraction of burst errors NOT detected for r = (n-k) + 1: $1/2^{(n-k-1)}$

fraction of burst errors NOT detected for r > (n-k) + 1: $1/2^{(n-k)}$

For example, if n-k = 32 as in CRC-32, then $1/2^{(n-k-1)} = 1/2^{31} \sim 10^{-9}$. For such a code, only one burst error out of a billion goes undetected even if the burst extent is the entire packet! And *all* bursts of extent 32 or less are detected.

Proof: See Theorem 6 of the original Peterson and Brown CRC paper appearing in the References.