MAT 312/AMS 351 Spring 2012 Review for Final

(use the reviews for Midterms 1 and 2 plus the following)

§5.4 and "Notes on binary codes." Understand that a binary code of length n is a subset C of the abelian group $\mathbf{Z}_2 \times \ldots \mathbf{Z}_2$ (n times), also written as $\mathbf{B}^n$ (sequences of length n of 0s and 1s; addition is vector addition: componentwise, mod 2). The code C is *linear* if C is a subgroup of $\mathbf{B}^n$. (Since mod 2 every word is its own inverse, the condition amounts to C being closed under addition). Understand the (Hamming) distance p.234 between codewords and for a linear code know how to reckon the usefulness of a code (for error detection &/or correction), i.e. the minimum distance netween codewords, by inspection of the set of code-words p.237. [After p.237 switch to material in the "Notes."]

Interpreting codewords as vectors, understand how a homomorphism (a linear transformation) $h : \mathbf{B}^n \to \mathbf{B}^m$ defines a linear code C_h by $C_h = \{\mathbf{x} \in \mathbf{B}^n | h(\mathbf{x}) = \mathbf{0}\}$, the set of all n-tuples which h sends to the zero m-tuple in $\mathbf{B}^m$. (This is called the *kernel* of h). Understand how h can be represented by right-multiplication by a matrix H: the m-vector $h(\mathbf{x})$ is the matrix product $\mathbf{x}H$; the first row of H is the m-vector $h(1, 0, 0, \ldots, 0)$, the second row is $h(0, 1, 0, \ldots, 0)$, etc.); we can call the code C_H. Know how to deduce the error-detecting or error-correcting properties of C_H by inspection of the matrix H.

§6.1, 6.2 Understand the similarity between the divisibility of polynomials $s(x), t(x), \ldots$ with coefficients in a field (the real numbers, $\mathbf{Z}_2$, etc.) and the divisibility of integers. Be able to carry out the division algorithm ("long division") for polynomials, giving a quotient and a remainder. Be able to carry out the Euclidean Algorithm to calculate a greatest common divisor $d(x)$ of $s(x), t(x)$ and to write $d(x)$ as a polynomial linear combination of $s(x)$ and $t(x)$. Note one difference: a polynomial $p(x)$ has a linear factor $(x - \alpha)$ if and only if $p(\alpha) = 0$. This is very useful in finite fields, since there are only finitely many possible α.

§6.3 Understand the definition of *irreducible* p.273; the distinction between irreducible and *prime* is not important in this context. Understand the proof of Theorem 6.3.4 (every polynomial can be written as a product of irreducibles) and the difference from Theorem 1.3.3 (unique factorization for integers): an irreducible factor is only determined up to a nonzero multiplicative constant. When the coefficient field is $\mathbf{Z}_2$ this difference does not manifest itself since the only nonzero constant is 1. Understand Examples 1 and 2 on p.277 completely.

§6.4 Understand that polynomial congruence classes are defined and have many properties like congruence classes of integers *mod m*. In particular, understand that when a polynomial $p(x)$ is irreducible, every nonzero congruence class *mod* $p(x)$ has a multiplicative inverse: Proposition 6.4.3, Example 2 p.281 and continuing in the Example on p.282. Understand the Example p.283 (important for §6.5): here $p(x) = x^n - 1$, every equivalence class has a unique representative $a_0 + a_1x + \cdots + a_{n-1}x^{n-1}$; multiplying this class by the equivalence class of the polynomial x gives $a_{n-1} + a_0x + a_1x^2 + \cdots a_{n-2}x^{n-1}$: the coefficients have cycled.

§6.5 Understand the definition of *cyclic code* of length n: it's a linear binary code where a cyclic permutation of the bits in a word leads to another word of the code. Understand the equivalence between the set of codewords and a set of polynomials *mod* $(x^n - 1)$:

$$(a_0, a_1, \ldots a_{n-1}) \leftrightarrow p(x) = a_0 + a_1x + \cdots a_{n-1}x^{n-1}$$

and that a cyclic permutation of the bits in the codeword corresponds to multiplication of the corresponding $p(x)$ by a power of the linear polynomial x (*mod* $(x^n - 1)$).

Understand how then linearity of the code forces the product $s(x)p(x)$ *mod* $(x^n - 1)$, for any polynomial $s(x)$ to also correspond to a codeword. [Remember, coefficients are $\mathbf{Z}_2$, so $s(x)p(x)$ is a sum of polynomials of the form $x^ip(x)$]. Understand the proof of Proposition 6.5.2: every cyclic code of length n, interpreted as a set of polynomials *mod* $(x^n - 1)$, has a *generator* $g(x)$: every polynomial in the code is a multiple of $g(x)$ *mod* $(x^n - 1)$. Be able to apply Corollary 6.5.3: such a $g(x)$ must be a divisor of $(x^n - 1)$. Factoring $(x^n - 1)$ gives all the possible $g(x)$, and consequently all the possible cyclic codes of length n.

Example. As in the text, consider cyclic codes of length 6. The generator $g(x)$ must be a divisor of $(x^6 - 1) = (x^3 - 1)^2 = (x - 1)^2(x^2 + x + 1)^2$. Take $g(x) = x^2 + x + 1$. The codeword polynomials are multiples of $g(x)$. Switching now to the methods explained in the "Notes on binary codes," we can generate the code using the linear transformation $h : \mathbf{B}^6 \to \mathbf{B}^2$ which assigns to each 6-tuple (polynomial of degree 5) its *remainder* after division by $1 + x + x^2$, which will be a (polynomial of degree 1) 2-tuple. The corresponding matrix H will have 6 rows; the rows will consist of the remainders of $1, x, x^2, x^3, x^4, x^5$, written as 2-tuples. Since

$$1 = 0(1 + x + x^2) + 1$$

$$x = 0(1 + x + x^2) + x$$

$$x^2 = 1(1 + x + x^2) + 1 + x$$

$$x^3 = (1 + x)(1 + x + x^2) + 1$$

$$x^4 = (x + x^2)(1 + x + x^2) + x$$

$$x^5 = (1 + x^2 + x^3)(1 + x + x^2) + 1 + x$$

the matrix is

$$H = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix}.$$

Since the matrix has only nonzero rows, but some of the rows are equal, this code is suitable for single error detection but not for double error detection or single error correction.

Taking instead $g(x) = (x-1)(x^3-1) = 1+x+x^3+x^4$, the remainders now have four coefficients:

$$\begin{aligned} 1 &= 0(1+x+x^3+x^4)+1 \\ x &= 0(1+x+x^3+x^4)+x \\ x^2 &= 0(1+x+x^3+x^4)+x^2 \\ x^3 &= 0(1+x+x^3+x^4)+x^3 \\ x^4 &= 1(1+x+x^3+x^4)+1+x+x^3 \\ x^5 &= x(1+x+x^3+x^4)+1+x^2+x^3 \end{aligned}$$

giving the matrix

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}.$$

In this matrix there is no zero row, no two rows are equal, and no three rows add up to zero. So the code C_H is suitable for triple error detection or double error detection and single error correction.