Chapter 1

6.897 Algorithmic Introduction to Coding Theory September 5, 2001

Lecture 1

Lecturer: Madhu Sudan *Scribe: Ryan O'Donnell*

1.1 Error correcting codes and this class

This course teaches the mathematics underlying objects called error correcting codes (ECCs). We won't define what they are today - for reasons to be clarified later. For now it will suffice to know that they are combinatorial objects that possess certain extremal properties which are very useful for the communication and storage of information.

Modulo the definition of ECCs, we can still discuss the big outline of this course. The contents of this course can be divided into four roughly equal parts. Our first part will be explain some of the basic constructions of error-correcting codes. Next we will show "negative results" showing limitations, or bounds, on the performance of codes. The two parts above are essentially combinatorial in nature, with few computational themes. We will get into the computational aspects in the third and fourth parts of this course. The third part of the course is where we focus on algorithmic tasks associated with ECCs and some solutions to these tasks. Finally, in the fourth part we will discuss some consequences one can derive in computational complexity theory as a result of the existence of ECCs and from the algorithmic capabilities surrounding them.

1.1.1 Standard references

Since the subject is quite old, there are plenty of texts. Yet we won't follow any one of them. Here are comments on some of them.

1. The text by MacWilliams and Sloane [74].

(a) This is possibly the most referenced text in CS?

(b) Unfortunately, it is getting outdated pretty fast.

(c) Has detailed coverage of (too) many families of codes.

(d) Coverage of algorithms is not so good.

2. The text by van Lint [117].

   (a) The book is much more concise and handy than MacWilliams and Sloane. Easier to get to some of the results here, if they are available.

   (b) Still, the emphasis is more combinatorial than algorithmic.

3. A text by Blahut [19].

   (a) Book was targetted at engineers rather than mathematics, or so the author claims.

   (b) Yet the coverage is excellent, especially for insight, motivations, definitions, even algorithms

   (c) The main drawback is that it is out of print and not easily available (not in MIT library, e.g.).

4. The Handbook of Coding Theory [88].

   (a) Offers extensive coverage of many recent themes.

   (b) Sometimes excessive. (E.g. 130 pages of table of best known codes.)

   (c) But contains many interesting chapters. E.g., chapter on algebraic-geometry codes, and the chapter on deep-space applications.

1.2 Information Theory

Even though this course is on coding theory, we start with a brief coverage of information theory. Part of the reason is historic. Coding theory was initiated by two seminal papers:

1. In 1948, Shannon wrote a detailed treatise on the mathematics behind communication [100].

2. In 1950, Hamming, motivated by the task of correcting small number of errors on magnetic storage media, wrote the first paper introducing error-correcting codes [48].

In this lecture we will discuss the main results from the Shannon paper, which founded the theory of information, while also co-founding (with Hamming) the theory of error-correcting codes. The theory of information provides the motivation for many questions in coding theory and so we will study this first.

Shannon considered the problem of two parties, say Alice and Bob, who wish to communicate digitally. In particular, Alice wishes to send a message to Bob over a certain digital channel. Shannon considered both "noiseless" and "noisy" channels. We start by considering the "noiseless" case.

1.2.1 Noiseless coding

In this case, the channels perfectly transmits the bits Alice wants to send to Bob. However, we might imagine that the channel is slow, so our goal is to compress the information we want to send down to as few bits as possible. Then we send these bits across the channel for Bob to receive and decompress. We start with an example which shows how to effect such a compression scheme.

1.2.2 An example

Consider transmitting the contents of a piece of paper, which contains some handwritten material on it, by fax. If the paper has just a small amount of black text on a white background, when it is digitized it might consist of 99% 0's and 1% 1's. Let's say for the sake of argument that we want to transmit a message in which each bit is independently 0 with probability .99 and 1 with probability .01.

Consider the following encoding scheme: We split the message up into blocks of 10 bits. If the block is 0000000000, we send the bit 0. If the block is anything else, say x, we send the string $1x$. Now the expected length of the encoding of a block is:

$$1 \cdot \Pr[\text{block is } 0000000000] + 11 \cdot \Pr[\text{block is not } 0000000000] \quad = \quad 11 - 10q$$

where $q := \Pr[\text{block is } 0000000000] = .99^{10} \geq .9$. Hence the expected length of the encoding of one 10 bit block is at most 2 bits. Thus the original message has been compressed to within 20% of its length! (Food for thought: Do we really need the probabilistic model to acheive this compression?)

Can we do better? This is exactly the question Shannon raised and answered with his theorem on noiseless coding.

Entropy

To analyze how much a distribution could be compressed, Shannon introduced some mathematical definitions associated with information. The source of information is modeled as a probability distribution, and a message is a random variable drawn from this distribution. The goal of compression is to encode the support of the probability space (using, say binary strings) so as to minimize the expected length of the message. To any source (or more correctly, to any distribution), Shannon assigned a non-negative real number, that he termed "entropy", that measures the information content of the distribution. He then showed that this quantity (to within an additive term of *one bit*) measures the compressibility of a bit.

The simplest possible distribution is a coin flip, in which heads has probability p and tails has probability $1 - p$. The entropy assigned to this is:

$$H(p) \stackrel{\text{def}}{=} p \log_2 \frac{1}{p} + (1-p) \log_2 \frac{1}{1-p}.$$

We can generalize this notion to arbitrary distributions.

Definition 1.1 *Let U be any finite set, and let D be a probability distribution on U, i.e., $D : U \to [0,1]$ with $\sum_{x \in U} D(x) = 1$. Let X be a random variable distributed according to D. Then the entropy*

of D is[1]*:*

$$H(D) \stackrel{\text{def}}{=} \sum_{x \in U} D(x) \log_2 \frac{1}{D(x)}.$$

The noiseless coding theorem of Shannon essentially says the following:

Theorem 1.2 *For every finite set U and for every distribution $D : U \to [0,1]$, There exists an encoding function $\mathsf{Enc} : U \to \{0,1\}^*$ and a decoding function $\mathsf{Dec} : \{0,1\}^* \to U$ such that for every $x \in U$, $\mathsf{Dec}(\mathsf{Enc}(x)) = x$, and*

$$\mathrm{Exp}_{x \leftarrow D}[|\mathsf{Enc}(x)|] \in [H(D), H(D)+1],$$

where $|s|$ denotes the length of a string s. Conversely, no error-free encoding can do better.

Shannon's actual theorem is often stated differently, but we will state the theorem in the above since it is clearer. We don't prove the theorem here, but the main idea for the existence result is the following: (1) Round all probabilities down so that they become powers of 2 (i.e., construct D' such that for every x, $D'(x) = 2^{-i}$ for some integer i, and $D(x)/2 < D'(x) \leq D(x)$. (2) Show that there exists an encoding Enc that encodes an element x, whose new probability $D'(x)$ equals 2^{-i}, with i bits (so that no strings have the same encoding). (3) Conclude that this encoding has an expected length in the desired range! The converse is harder to prove - we won't get into it.

An elegant algebraical identity

The entropy function exhibits some very nice mathematical properties. For example if a distribution D decomposes into two independent distributions D_1 and D_2 (i.e., $U = U_1 \times U_2$, $D_1 : U_1 \to [0,1]$ and $D_2 : U_2 \to [0,1]$ and $D(x,y) = D_1(x)D_2(y)$), then $H(D) = H(D_1) + H(D_2)$. This fact can be use to prove an interesting algebraic identity, which is otherwise quite tricky to prove.

From the above property and the entropy of a bit, we find that the entropy in n independent p-biased coin flips is $nH(p)$. On the other hand, this should be equal to the entropy of the distribution D given by $D(x) := p^{\#1's \text{ in } x}(1-p)^{\#0's \text{ in } x}$. Using the second formula, we calculate this as:

$$\sum_{t=0}^{n} \binom{n}{t} D_t \log_2 \frac{1}{D_t}$$

where $D_t := p^t(1-p)^{n-t}$. Thus we get

$$\sum_{t=0}^{n} \binom{n}{t} D_t \log_2 \frac{1}{D_t} = nH(p).$$

Entropy and its variants play an important role in combinatorics and probability. Some very useful notions to keep track of in this area are those of mutual information, conditional entropy, and relative entropy!

To turn back to our example, what is the optimal compression factor for the case of our message to be faxed, whose bits were 1 with probability 1%? The answer is $H(.01) \approx 8\%$ — message can be compressed to within $H(.01)$ of their original length, if we know ones occur with probability about 1%.

[1] In lecture, we used a random variable X drawn according to D and defined this to be the entropy of X, rather than D. In these notes we will switch to the more appropriate notation, which defines it to be a function of the distribution and not the variable.

1.2.3 Noisy channels

For the purposes of this course, the more interesting notion of a channel is the class of "noisy" channels considered by Shannon — i.e., channels that flip some of the bits sent across them. The problem here is for Alice to encode the message she wishes to send in such a way that even if the channel corrupts some of the bits in the encoding, Bob will be able to decode the result into Alice's original message, with high probability. Shannon considered a large class of probabilistic models for noisy channels, and for each proved the surprising theorem that you could always overcome a constant error rate by sending an encoded message that was longer by a constant factor.

The general model Shannon gave for channels is as follows. There is an input alphabet Σ and an output alphabet Γ (both usually finite). Then we have a bipartite graph with Σ on the left and Γ on the right; each edge (σ, γ) is labeled with a probability of the channel converting a σ to a γ. (The channel operates independently on each character.) Of course, for each σ on the left, the sum of the labels on the edges touching it must be 1.

We will illustrate his results for such channels only in cases where $\Sigma \subseteq \Gamma$. (In such cases, it is clear what an error is.)

Two commonly considered channels:

1. Binary Symmetric Channel — the channel considered in the theorem. $\Sigma = \Gamma = \{0, 1\}$, $(0, 0)$ and $(1, 1)$ get probability $1 - p$, and $(0, 1)$ and $(1, 0)$ get probability p.
2. Binary Erasure Channel — in this channel, bits don't get flipped — rather they get erased. Specfically, $\Sigma = \{0, 1\}$, $\Gamma = \{0, 1, ?\}$, $(0, 0)$ and $(1, 1)$ get probability $1 - p$, and $(0, ?)$ and $(1, ?)$ get probability p.

Noisy coding theorem

The noisy coding theorem of Shannon is a powerful and general one. When specialized to the case of a binary symmetric channel with error probability p, we get the following result.

Theorem 1.3 *For every $p < 1/2$, there exists a constant $c < \infty$ and a pair of functions $E : \{0, 1\}^k \to \{0, 1\}^{ck}$, and $D : \{0, 1\}^{ck} \to \{0, 1\}^k$ with the following property: If we pick a message uniformly at random from $\{0, 1\}^k$, encode with E and then send the result across the noisy channel, and decode the result, then we recover the original message with probability $1 - o(1)$.*[2]

Theorem 1.3 applies — with different constants — to the binary erasure channel as well.

Hamming notations

We need just a few definitions before proceeding with the proof of the theorem.

Definition 1.4 *If x and y are in Σ^n, then the* Hamming distance between them is $\Delta(x, y) :=$ # of coordinates on which x and y differ.

[2] The $o(1)$ term depends only on k.

Definition 1.5 *The* Hamming ball of radius r centered at y is $B(y,r) := \{x \in \Sigma^n : \Delta(x,y) \leq r\}$.

Definition 1.6 $\mathrm{Vol}(r,n)$ *denotes the volume of (any) radius-r ball in $\{0,1\}^n$; i.e., $|B(y,r)|$. Exactly, this quantity is $\sum_{i=0}^{r} \binom{n}{i}$. If we fix some $p > 0$ and let $n \to \infty$ then we get* $\mathrm{Vol}(pn,n) = 2^{(H(p)+o(1))n}$.

Proof of Theorem 1.3

Proof The proof is highly non-constructive; it uses the probabilistic method.

Let $n > k$ be decided upon later. Pick the encoding function $E : \{0,1\}^k \to \{0,1\}^n$ at random, i.e., for every $m \in \{0,1\}^k$, $E(m)$ is chosen uniformly at random from $\{0,1\}^n$, independently of all other choices.

The decoding function $D : \{0,1\}^n \to \{0,1\}^k$ works as follows. Given a string $y \in \{0,1\}^n$, we find (non-constructively) the $m \in \{0,1\}^k$ such that $\Delta(y, E(m))$ is minimized. This m is the value of $D(y)$.

We now prove that D and E have the required property, with high probability, over the random choice of E.

Fix $m \in \{0,1\}^k$ as also $E(m)$. (The rest of the proof will use the fact that $E(m')$ for $m' \neq m$ is still random.) the message being sent. Denote by y the corrupted version of $E(m)$ that is received by Bob. Let η denote the error vector $y - E(m)$. Note that the η is a random variable with each of its coordinates being 1 w.p. p and 0 w.p. $1-p$, independent of other coordinates.

Fix $\epsilon > 0$. Let $r = (p+\epsilon)n$. In order for $D(y) \neq m$ at least one of the following two events must occur:

1. $y \notin B(E(m), r)$ (i.e., too many errors occurred in transmission.)
2. There exists some $m' \neq m$ such that $E(m') \in B(y.r)$. (The errors take the received word too close to the encoding of some other message.)

In particular, if neither event occurs, then m is the unique message such that $E(m)$ is within a distance of r from y and so $D(y) = m$.

We show that for an appropriate choice of n, the events above happen with low probability. For the first event to happen, it must be that η has more than $p + \epsilon$ fraction of 1s. We can apply Chernoff bounds (see appendix at the end of this lecture) to see that:

$$\Pr_{\eta}[y \notin B(E(m), r))] \leq 2^{-(\epsilon^2/2)n}.$$

For any $\epsilon > 0$, we can pick n to be large enough that the above quantity is as small as we want.

We now move onto the second event. First fix y and an $m' \neq m$ and consider the event that $E(m') \notin B(y,r)$. The probability of this event, taken over the random variable $E(m')$, is exactly $\mathrm{Vol}(B(y,r))/2^n$. Using the approximation $\mathrm{Vol}(B(y,pn) \approx 2^{H(p)n}$ and ignoring ϵ (which is arbitrarily small), we find that for every $m' \neq m$.

$$\Pr_{E}[E(m') \in B(y,r)] \approx 2^{H(p)n-n}.$$

Using the union bound, we get that

$$\Pr_E[\exists m' \neq m \text{ s.t. } E(m') \in B(y,r)] \approx 2^{k-n+H(p)n}.$$

Thus we find that if $c > \frac{1}{1-H(p)}$, then $k/n < 1 - H(p)$ and thus the quantity above is less than one. This gives the choice of c that we need.

Our proof is not yet complete! Why? Well, we have argued that a fixed m is likely to be decoded correctly, but what about other messages? To complete the argument, we will actually need to lose a little bit.

Let $\delta = 2^{-(\epsilon^2/2)n} + 2^{k-n+H(p)n}$, denote the total probability of error in either of the two steps above. We have shown that for any fixed m, for a random choice of E and the associated D, the expected decoding error probability is at most δ. Thus we can now conclude that this expectation continues to hold if we make m a random variable chosen uniformly from $\{0,1\}^k$. We get

$$\text{Exp}_{m,E,\eta}[D(E(m)+\eta) \neq m] \leq \delta.$$

In particular this implies that there exists an E such that for the corresponding D, we have

$$\text{Exp}_{m,\eta}[D(E(m)+\eta) \neq m] \leq \delta.$$

This concludes the proof of Shannon's theorem. ■

Capacity of the channel

Shannon's theorem above shows that if we are willing to slow down the effective transmission rate of the channel to $1/c < 1 - H(p)$, then we can acheive error-free communication with vanishingly small probability. Furthermore this quantity $1 - H(p)$ is only a function of the "noisy channel" and not of the number of bits we wish to transmit over it. I.e., the effective rate of transmission can be an absolute constant independent of n. Shannon called this quantity (the limiting rate of k/n, as $n \to \infty$) the *capacity* of the noisy channel.

For the Binary Symmetric Channel, how large can its capacity be? The proof above shows that the capacity, denoted $C_{\text{BSC}}(p)$, is at least $1 - H(p)$. It is natural to ask if this quantity is an artifact of the proof, or is it the correct capacity for the channel. Shannon proved that the latter was the case. We will prove this in the next lecture, but first let us see why this is the case intuitively.

Suppose that we are actually is a setup there is a noiseless channel between Alice's location and Bob's, but this channel has been "hijacked" by Eve and Fred. Say Alice and Eve are in the same physical location while Fred and Bob are at the other. To send a message over, Alice must hand it over to Eve who then sends it through the channel (after some potential corruption). Similarly at the receiving end, Fred receives the message and hands it over to Bob. Suppose Eve and Fred want to use this channel to exchange some messages of their own (at Alice & Bob's expense). They do so by informing Alice and Bob that the channel is noisy with bits being flipped with probability p. They advise Alice and Bob to use some encoding/decoding schemes. Alice and Bob agree on an encoding scheme E and a decoding scheme D and in their naivette share these functions with Eve and Fred as well.

In truth it may be that Eve wishes to use the "noise" to send some messages of her own to Fred. Say she has a message η which is a plain paper image, where each pixel of the page is 1 independently

with probability p. The way she sends η to Fred (at Alice's expense) is that when Alice gives her a encoded message $E(m)$ to transmit, Eve sends over $E(m) + \eta$. Fred receives $y = E(m) + \eta$ (the channel does not introduce any noise) at the receiving end and passes it on, untampered, to Bob, but also retains a copy. As far as Alice and Bob are concerned, nothing mailicious is occuring - with high probability $D(y) = m$ and so they are exchanging messages at capacity of the "noisy channel". But note the situation w.r.t. Eve and Fred. Fred also knows E and D and can compute $\eta = y - E(D(y)) = E(m) + \eta - E(m)$, with high probability. So Eve and Fred are also exchanging messages among themselves (with some small probability of exchanging incorrect messages). But now if we consider Alice & Eve together at one end of the noiseless channel, and Bob & Fred together at the other end, the parties are exchanging bits at a rate of at least $C(p) + H(p)$ across the noiseless channel (assuming we believe the tightness of the noiseless coding theorem). Since we are normalizing so that the rate the noiseless channel is 1, we get $C(p) + H(p) < 1$! This is the link between the noisy case and the noiseless case of the Shannon theorems.

Appendix

Notation used in the lecture. We mention some notation that we used earlier or may use in later lectures. $\mathbb{Z}$ denotes the set of integers, $\mathbb{Z}^{\geq 0}$ denotes the set of non-negative integers, and $\mathbb{Z}^+$ the set of positive integers. $\mathbb{R}$ denote the set of reals, and $\mathcal{Q}$ the rationals. For real numbers a and b, the notation $[a, b]$ stands for the closed interval from a to b, i.e., the set $\{x \in \mathbb{R} | a \leq x \leq b\}$, while (a, b) is the open interval between a and b. For an integer k, we will use $[k]$ to denote the set $\{1, \ldots, k\}$. If D is a distribution on the universe U, then $X \leftarrow D$ denotes a random variable X drawn from U according to the distribution D. For an event $\mathcal{E} \subseteq U$, the quantity $\Pr_{X \leftarrow D}[X \in \mathcal{E}]$ denotes the probability of the event $\mathcal{E}$ when X is chosen according to D. For a real-valued function $f : U \to \mathbb{R}$, the quantity $\mathrm{Exp}_{X \leftarrow D}[f(X)]$ denotes the expected value of $f(X)$ when X is chosen according to D. When the distribution D is clear from context, we may abbreviate these quantities to $\Pr[\mathcal{E}]$ and $\mathrm{Exp}_X[f(X)]$. Similarly $\mathrm{Var}_{X \leftarrow D}[f(X)]$ denotes the variance of $f(X)$ (i.e., $\mathrm{Var}(f(X)) = \mathrm{Exp}[(f(X))^2] - \mathrm{Exp}[f(X)]^2$).

Some basic probability facts Here is a quick recap of basic facts on probability and expectations.

Probability One of the most used facts on probability is the union bound: $\Pr[\mathcal{E}_1 \cup \mathcal{E}_2] \leq \Pr[\mathcal{E}_1] + \Pr[\mathcal{E}_2]$. Note that the bound makes no "independence" assumptions.

Expectations Analogous to the above we have: $\mathrm{Exp}[X_1 + X_2] = \mathrm{Exp}[X_1] + \mathrm{Exp}[X_2]$. Note that this is an equality! If random variables are independent, then we get a product relationship $\mathrm{Exp}[X_1 X_2] = \mathrm{Exp}[X_1]\mathrm{Exp}[X_2]$.

Converting probabilities to expectations: Since expectations are more amenable to algebraic manipulations, it is often useful to convert statements on probability to statements of events. The standard way to do this is to use "indicator variables". For an event $\mathcal{E}$, let $I_{\mathcal{E}}$ be the 0/1-valued variable given by $I_{\mathcal{E}}(X) = 1$ if $X \in \mathcal{E}$ and $I_{\mathcal{E}}(X) = 0$ otherwise. Then we have $\mathrm{Exp}_X[I_{\mathcal{E}}(X)] = \Pr[\mathcal{E}]$.

Converting expectations to probabilities: Since probabilities are the quantities that have more intuitive meaning, these are the more standard targets of our investigation. Since expectations figure in proofs, we would like to find ways to convert statements on expectations back into probability statements. This conversion is not standard. Several "tail inequalities" are used to acheive this conversion, e.g., Markov's, Chebychev's, and Chernoff's: We state them below:

Markov's inequality For a non-negative random variable X and positive real α, then $\Pr[X \geq \alpha] \leq \frac{E[X]}{\alpha}$.

Chebychev's inequality In its general form, this inequality is just an application of Markov's inequality to the random variable Y^2, for arbitrary Y. In the general form, it is quite hard to see its strength, so we give a special form.

Let $Y = \sum_{i=1}^{n} Y_i$, where the Y_i's identically distributed random variables that are pairwise (but not fully) independent. Let $\mathrm{Exp}[Y_i] = \mu$ and $\mathrm{Var}[Y_i] = \sigma^2$. Then for $\lambda > 0$, we have $\Pr[Y \geq (\mu + \lambda)n] \leq \frac{\sigma^2}{n\lambda^2}$.

Chernoff bounds If $Y_1, \ldots, Y_n$ are completely independent it is possible to get stronger bounds on the probability that their sum deviates much from their expectation. We consider the special case of variables taking values in the interval $[0, 1]$

Let $Y_1, \ldots, Y_n$ be independent and identically distributed random variables taking values in the interval $[0, 1]$ with mean μ. Let $Y = \sum_i Y_i$. Then $\Pr[Y \geq (\mu + \lambda)n] \leq e^{-(\lambda^2/2)n}$, where e is the base of the natural logarithm.

For further elaboration on probabilities and expectations one may consult the text on Randomized Algorithms [82].

Chapter 2

6.897 Algorithmic Introduction to Coding Theory September 10, 2001

Lecture 2

Lecturer: Madhu Sudan *Scribe: Omprakash Gnawali*

Today we will cover the following topics:

- Converse to Shannon's coding theorem.
- Some remarks on Shannon's coding theorem.
- Error correcting codes.
- Linear codes.

2.1 Converse to Shannon's coding theorem

Recall that the Binary Symmetric Channel (BSC) with parameter p is the channel that transmits bits, flipping each transmitted bit with probability p independent of all other events. Lets start by recalling Shannon's coding theorem informally.

> Over the BSC with parameter p, it is possible to transmit information at any rate less than $1 - H(p)$.

(To give a sense of how to make the above formal, here is the formal version in all its quantified glory: "For every $p < \frac{1}{2}$ and $\epsilon, \delta > 0$, there exists an $n_0 < \infty$ such that for every $n \geq n_0$ and $k \leq (1 - H(p) - \epsilon)n$, there exist functions $E : \{0,1\}^k \to \{0,1\}^n$ and $D : \{0,1\}^n \to \{0,1\}^k$ such that

$$\Pr_{\eta \leftarrow D_{p,n}, m \leftarrow U_k}[D(E(m) + \eta) = m] \geq 1 - \delta,$$

where U_k is the uniform distribution on $\{0,1\}^k$ and $D_{p,n}$ is the distribution on n bits chosen independently with each bit being 1 with probability p.")

We will now proof a converse to this theorem.

Theorem 2.1 *For every $p < \frac{1}{2}$ and $\epsilon, \delta > 0$, there exists an $n_0 < \infty$ such that for every $n \geq n_0$, $k \geq (1 - H(p) + \epsilon)n$, and functions $E : \{0,1\}^k \to \{0,1\}^n$ and $D : \{0,1\}^n \to \{0,1\}^k$, it is the case that*

$$\Pr_{\eta \leftarrow D_{p,n}, m \leftarrow U_k}[D(E(m) + \eta) \neq m] \geq 1 - \delta,$$

where U_k is the uniform distribution on $\{0,1\}^k$ and $D_{p,n}$ is the distribution on n bits chosen independently with each bit being 1 with probability p.

Ignoring all quantifiers above, the essense is that if we are trying to send information at a rate of $1 - H(p) + \epsilon$, then the decoding is erroneous with probability almost 1, *no matter* which encoding and decoding function we use.

Proof The hard part of this proof is deciding how to deal with the encoding and decoding functions which are completely arbitrary! Turns out, we will ignore the encoding function entirely, and ignore the decoding function almost entirely! The main focus is on the error, and the fact that the error distributes the transmitted word over a large space of possibilities (and so any decoding function should be helpless). Specifically, we note the following:

- The number of errors is unlikely to be too small or too large: Formally, for every m and E,

$$\Pr_{\eta}[E(m) + \eta \in B(E(m), (p - \epsilon)n)] = \Pr_{\eta}[\eta \in B(\mathbf{0}, (p - \epsilon)n)] \leq e^{-\epsilon^2 n}. \tag{2.1}$$

  The equality above is obtained by simply translating the center of the ball from $E(m)$ to the origin $\mathbf{0}$, the string consisting of all 0's. The inequality above is a straightforward application of Chernoff bounds - however, this time we are using the fact that it proves that a random variable is not likely to take on a value much less than its expectation. Similarly, we get that

$$\Pr_{\eta}[E(m) + \eta \notin B(E(m), (p + \epsilon)n)] = \Pr_{\eta}[\eta \in B(\mathbf{0}, (p - \epsilon)n)] \leq e^{-\epsilon^2 n}.$$

- Given that the error is large (but not too large), no single point in the space has a high probability of being the received word. Specifically, for every m, E and $y \in B(E(m), (p + \epsilon)n) - B(E(m), (p - \epsilon)n)$,

$$\Pr_{\eta}[E(m) + \eta = y] \leq \frac{n}{2^{H(p-\epsilon)n}}. \tag{2.2}$$

  To see why the above is true, let $R = \Delta(y, E(m))$ be the Hamming distance between y and $E(m)$. Note we have $R \in [(p - \epsilon)n, (p + \epsilon)n]$. Let N_R be the number of binary vectors with R ones and $n - R$ zeroes. Since all error patterns η with the same number of errors are equally likely, we note we the probability of having any fixed vector containing exactly R ones as the error vector is at most $\frac{1}{N_R}$. Thus to upper bound the probability of the event in question it suffices to lower bound N_R for $R \in [(p - \epsilon)n, (p + \epsilon)n]$. Using the fact that $N_R = \mathrm{Vol}(R, n) - \mathrm{Vol}(R-1, n)$, and the fact that N_R is increasing for R in the range $[0, \frac{n}{2}]$, we get than $N_R \geq \frac{\mathrm{Vol}(R,n)}{n} \geq \frac{\mathrm{Vol}((p-\epsilon)n,n)}{n}$. Now using the fact that $\mathrm{Vol}((p - \epsilon)n, n) \approx 2^{-H(p-\epsilon)n}$, we get that the probability $\eta = y - E(m)$ is at most $\frac{n}{2^{H(p-\epsilon)n}}$.

To continue the proof, we finally look at the decoding function (though even this look will be very superficial). Let $K = 2^k$ and let $\{m_1, \ldots, m_K\}$ denote the K possible messages. Let $S_i = \{y | D(y) = m_i\}$ be the set of received words that are decoded to the ith message. The only property we use about the decoding is that $\sum_{i=1}^{K} |S_i| = 2^n$, i.e., the decoding is a *function*! (On the other hand,

there is little else to use!) We now use the observations in the previous paragraph to prove that the decoding function is not very likely to succeed.

Let ρ be the probability of decoding successfully. In order for the decoding to succeed, we must pick some message m_i to encode and transmit, and the error vector η must be such that the received vector $y = E(m_i) + \eta$ must lie in S_i. This gives:

$$\rho = \sum_{i=1}^{K} \sum_{y \in S_i} \Pr_{m,\eta}[m = m_i \text{ and } \eta = y - E(m_i)] = \Pr_{m}[m = m_i] \Pr_{\eta}[\eta = y - E(m_i)],$$

where the second equality follows from the fact that the events considered are independent. Fixed m_i and let us bound the inner summation above. The probability that m equals m_i is exactly $1/K = 2^{-k}$. The event that $\eta = y - E(m_i)$ is independent of m and so we can estimate this quantity separately. Fix m_i. Let $U = B(E(m_i), (p-\epsilon)n)$ and $V = \{0,1\}^n - B(E(m_i), (p+\epsilon)n)$ (i.e., U is the points too close to $E(m)$ and V the points too far from $E(m)$. Then

$$\begin{aligned}
&\sum_{y \in S_i} \Pr_{\eta}[\eta = y - E(m_i)] \\
&\quad \leq \sum_{y \in U} \Pr_{\eta}[\eta = y - E(m_i)] + \sum_{y \in V} \Pr_{\eta}[\eta = y - E(m_i)] + \sum_{y \in S_i - U - V} \Pr_{\eta}[\eta = y - E(m_i)] \\
&\quad \leq 2e^{-\epsilon^2 n} + |S_i| \frac{n}{2^{H(p-\epsilon)n}},
\end{aligned}$$

where the second inequality above follows from Equations (2.1) and (2.2). Combining the above, we have:

$$\begin{aligned}
\rho &\leq \sum_{i=1}^{K} 2^{-k} \left(2e^{-\epsilon^2 n} + \frac{n|S_i|}{2^{H(p-\epsilon)n}} \right) \\
&= 2e^{-\epsilon^2 n} + \frac{n 2^{-k}}{2^{H(p-\epsilon)n}} \left(\sum_{i=1}^{K} |S_i| \right) \\
&= 2e^{-\epsilon^2 n} + n 2^{-k - H(p-\epsilon)n + n}
\end{aligned}$$

The theorem follows from the fact that for every $\epsilon, \delta > 0$ we can pick n_0 large enough so that for every $n \geq n_0$, it is the case that $2e^{-\epsilon^2 n} \leq \delta/2$ and $n 2^{-k - H(p-\epsilon)n + n} \leq \delta/2$ (assuming $k \geq (1 - H(p) + \epsilon)n$). ■

Recall that at the end of the previous lecture we showed that if we assumed the noiseless coding theorem is tight (i.e, has a converse) then the converse to the noisy coding theorem follows. While the theorem above does not imply a converse to the noiseless coding theorem, the proof technique is general enough to capture the noiseless coding theorem as well. This motivates the following exercise.

Exercise: Prove converse of the noiseless coding theorem (from Lecture 1).

2.2 Remarks on Shannon's Theorem

2.2.1 Discrete Memoryless Channels

Shannon's coding theorem is, of course, much more general that what we have presented. We only presented the result for the case of the Binary Symmetric Channel. For starters, the result can be generalized to the case of all "Discrete Memoryless Channels (DMCs)". Such channels are characterized by two finite sets — Σ representing the input alphabet of the channel and Γ representing the output alphabet of the channel — and a transition probability matrix $P = \{p_{\sigma\gamma}\}$, where $p_{\sigma\gamma}$ denotes that probability that the output alphabet is γ given than the input alphabet is σ. We require that $\sum_{\gamma\in\Gamma} p_{\sigma\gamma} = 1$ for every $\sigma \in \Sigma$, so that this definition makes sense. When we attempt to transmit a sequence of symbols from Σ over this channel, it behaves on each element of the channel independently and produces a sequence of elements from Γ, according to the transition probability matrix P.

Given such a channel, characterized by P, Shannon gave a procedure to compute the capacity of the channel. This capacity relates to the mutual information between two random variables.

Given a distribution D_Σ over Σ, let σ be a random variable chosen according to D_Σ. Pick γ at random from Γ with probability $p_{\sigma\gamma}$. Denote by $D_{\Sigma,\Gamma}$ the joint distribution on the pairs (σ, γ) so generated, and by D_Γ the marginal distribution of γ. Since D_Σ, D_Γ and $D_{\Sigma,\Gamma}$ are all distributions on finite sets, their entropy is well defined. Define the "mutual information" between variables σ and γ (or more correctly of the transition matrix P with initial distribution D_Σ) to be $H(D_\Sigma) + H(D_\Gamma) - H(D_{\Sigma,\Gamma})$.

Shannon's theorem showed that the capacity of the channel characterized by P, is the maximum over all distributions D_Σ, of the mutual information between σ and γ. He also gave a linear system whose solution gave the distribution that maximizes this information.

2.2.2 Markovian Channels

Shannon's theory extends even further. Natural scenarios of error may actually flips bits with some correlation rather than doing so independently. A subclass of such correlations is given by "Markovian channels", where the channel can be in one of several (finitely many) states. Depending on which state the channel is in, the probability with which it makes errors may be different.

Such models are useful in capturing, say, "burst error" scenarios. In this situation the channel makes sporadic sequences of many errors. One can model this source by a channel with two states (noisy/normal) with two different channel characteristics for the two states. (see Figure 2.1). When in the "noisy" state the channel flips every bit, say, with probability $\frac{1}{2}$ (and so a channel that is perpetually in a noisy state can transmit no information). When in the normal state, however, the channel flips bits with only a small probability, say p. Further, if the channel is in a given state at time t it tends to stay in the same state at time $t+1$ with probability $1-q$ and tends to flip its state with a small probability q. Is it possible to transmit information on such a channel? If so, at what rate? Working this out would be a good exercise!

Shannon's theory actually gives the capacity of such a channel as well. Deciphering what the theory says and unravelling the proofs would be a good topic for a term paper.

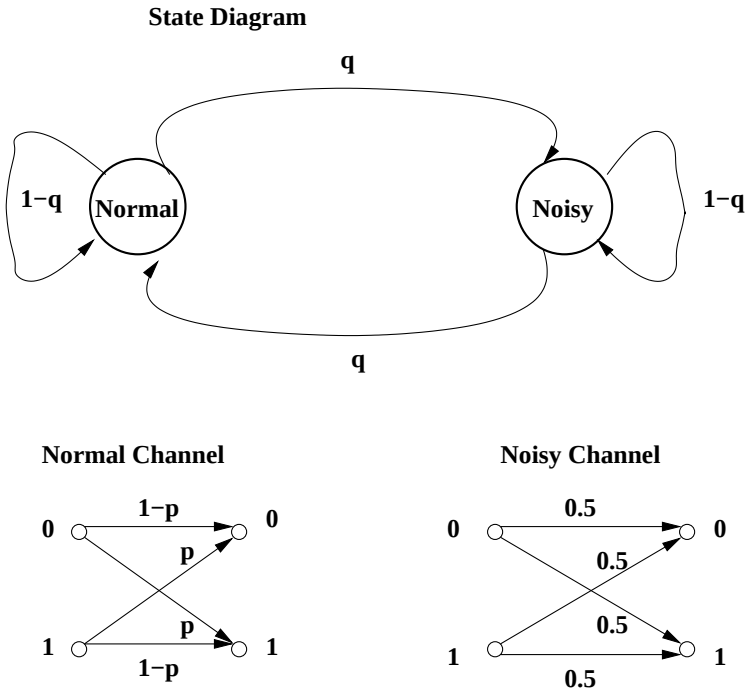

Figure 2.1: State diagram of a simple burst error channel

2.2.3 Zero Error Capacity of a channel

An interesting by product of the Shannon theory is the so called "Zero error capacity" of a channel. To motivate this notion, let us consider a "stuck typewriter". Suppose the keys in the typewriter are sticky and likely to produce the wrong symbol when you hit it. Further suppose that the error pattern is very simple. If you hit a letter of the keyboard, you get as output either the correct letter or the next letter of the alphabet. and that such an error happens with probability $\frac{1}{2}$ for every keystroke independently. In other words, typing A results in A or B, typing B results in B or C, etc. and typing Z results in Z or A (so we have a wrap around). (See Figure 2.2.)

Some analysis of this channel reveals that it has a channel capacity of $\log_2 13$. I.e., each keystroke is capable, on the average of conveying one of 13 possibilities. If the typewriter had not been stuck, it would have had a capacity of $\log_2 26$. Thus the error cuts down the number of possibilities per stroke by a factor of 2.

If we think hard (actually may be not even so hard) we can see a way of achieving this capacity. Simply don't use the even-numbered letters of the alphabet (B,D,F, etc.) and work only with the odd-numbered ones (A, C, E etc.). Since there is no possibility of confusion within the odd-numbered letters, there is no ambiguity in the message (if an A or B is received, A must have been the keystroke typed, if C or D is received, C is the keystroke etc.). The interesting aspect of this way of using the channel is that we acheive the capacity, *with zero error*! This motivates a general concept: The *Zero Error Capacity* of a channel is informally defined as the optimal rate of transmission that can be achieved while maintaining a zero probability of decoding error.

In the case of the stuck typewriter the zero error capacity equals the Shannon capacity. This is

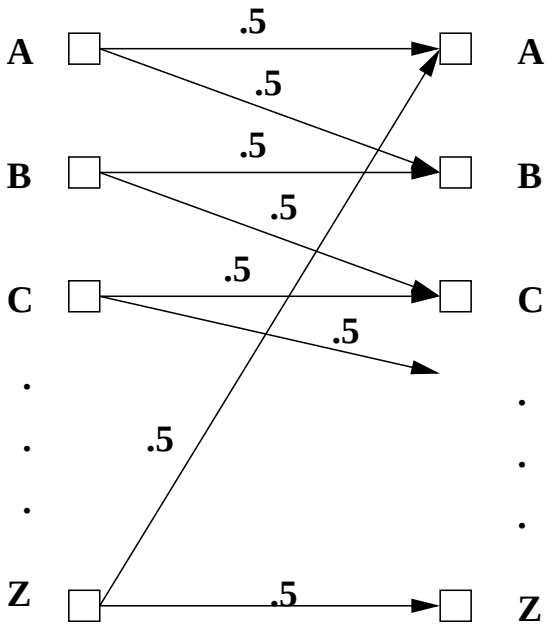

Figure 2.2: Channel for a stuck typewriter

not always the case. For example, the zero error capacity of the binary symmetric channel for any $p \neq 0, 1$ is zero! (Any string has positive probability of being corrupted into to any other string.) It is true that the zero error capacity is less than or equal to the Shannon capacity. Computing the Shannon capacity of even simple channels is non-trivial, and in general this function is not known to be computable.

A simple illustrative example is the zero error capacity of a stuck keyboard with only five keys (or in general an odd number of keys)! Then it is no longer clear what the zero error capacity of this channel is. In 1979, L. Lovász [71] wrote a brilliant paper that showed how to compute the Shannon capacity of several graphs (and in general give a lower bound on the Shannon capacity). In particular he shows that the Shannon capacity of a stuck typewriter with 5 keys is $\sqrt{5}$ - an irrational number! This paper has played a pioneering role in computer science leading to the notion of semi-definite programming and its consequences on combinatorial optimization.

Term paper topic: Study the zero error capacity of arbitrary channels and survey the work of Lovász and successors.

This will terminate our discussion of the Shannon based theory. As mentioned in the first lecture, Shannon's original paper [100] and the text by Cover and Thomas [26] are excellent sources for further reading.

2.3 Error correcting codes

Shannon's theory, while providing exact results for the rate at which one can communicate on a noisy channel, are unfortunately highly non-constructive. Specifically the two key ingredients: the

encoding and *decoding* functions are totally non-constructive. In order to get some sense of how to make these results constructive, one has to examine the encoding function and see what properties about it are useful. Shannon noticed that a result of Hamming indeed does so, and that this may be a step in making his results more constructive. In retrospect it seems this was crucial in making Shannon's results constructive. This is the theory of error-correcting codes, as initiated by the work of Hamming [48]. Hamming focussed on the set of strings in the image of the encoding map, and called them "error-correcting codes". He identified the distance property that would be desirable among the codes and initiated a systematic study. We develop some notation to study these notions.

2.3.1 Notation

We consider codes over some alphabet Σ and reserve the letter q to denote the cardinality of Σ. It is often helpful to think of $\Sigma = \{0, 1\}$ and then the codes are termed binary codes.

We consider transmissions of sequences of n symbols from Σ from sender to receiver. Recall that for two strings $x, y \in \Sigma^n$, the Hamming distance between x and y, denoted $\Delta(x, y)$, is the number of coordinates where x differs from y. We note that the Hamming distance is indeed a metric: i.e., $\Delta(x, z) = \Delta(z, x) \leq \Delta(x, y) + \Delta(y, z)$ and $\Delta(x, y) = 0$ if and only if $x = y$.

A code C is simply a subset of Σ^n for some positive integer n. The minimum distance of a code C, denoted $\Delta(C)$, is given by $\Delta(C) = \min_{x,y \in C, x \neq y}\{\Delta(x, y)\}$. The Hamming theory focusses on the task of constructing (or showing the existence of) codes with large minimum distance and large cardinality.

There are four fundamental parameters associated with a code C:

- Its *block length*: n, where $C \subseteq \Sigma^n$.
- Its *message length*: $k = \log_q |C|$. (To make sense of this parameter, recall that we are thinking of the code as the image of an encoding map $E : \Sigma^k \to \Sigma^n$ and in this case $\log_q |C| = k$ is the length of the messages.)
- Its *minimum distance* $d = \Delta(C)$.
- Its *alphabet size*: $q = |\Sigma|$.

It is often customary to characterize a code by just the four parameters it achieves and refer to such a code as an $(n, k, d)_q$-code.

2.3.2 Broad Goals of Coding Theory

In a nutshell the broad goal of coding theory can be stated in one of the four ways below, where we fix three of the four parameters and try to optimize the fourth. The correct optimizations are:

- Given k, d, q find an $(n, k, d)_q$ code that minimizes n.
- Given n, d, q find an $(n, k, d)_q$ code that maximizes k.
- Given n, k, q find an $(n, k, d)_q$ code that maximizes d.
- Given n, k, d find an $(n, k, d)_q$ code that minimizes q.

The first three choices are self-explanatory. It is always desirable to have a small block length, large message length, and large distance. However it is not so immediate that minimizing q is the right thing to do (in particular, we don't have a monotonicity result). However empirically (and almost certainly) it seems to be the case that one can get get values of the parameters for larger values of q and getting good parameters for small values of q is the challenging part. Furthermore, building codes with large q and then trying to reduce q is a very clever way of getting good codes. So we will keep this version in mind explicitly.

2.3.3 Error Correcting Codes

Why are we interested in codes of large minimum distance? It may be worth our while to revisit Hamming's paper and see what he had to say about this. Hamming actually defined three related properties of a code C.

1. The minimum distance of C, which we have already seen.
2. The error detection capacity of C: A code C is e-error detecting if under the promise that no more than e errors occur during transmission, it is always possible to detect whether errors have occured or not, and e is the largest integer with this property. Hamming notes that a e-error correcting code has minimum distance $e+1$.
3. The error correction capacity of C: A code C is t-error correcting if under the promise that no more than t errors occur during transmission, it is always possible to determine which locations are in error (information-theoretically, but not necessarily efficiently) and correct them, and if t is the largest integer with this property. Hamming notes that a t-error correcting code has minimum distance $2t+1$ or $2t+2$.

Thus the minimum distance of a code is directly relevant to the task of correcting errors and we will focus on this parameter for now. Later (in the second part of the course) we will turn our attention to the question - how can we make these error-detection and error-correction capabilities algorithmic.

2.3.4 Some simple codes

The famed Hamming codes are codes of minimum distance three. Even though Hamming's name is associated with distance-3 codes, he also gave, in the same paper. codes of distance two and four! Let us start even slower and describe the distance one codes first!

- $d = 1$: This is trivial. All we want is that the encoding function be injective. So the identity function works and gives the best possible $(n, n, 1)_q$ code.
- $d = 2$, This is already interesting. A simple way to achieve distance 2 is to append the parity of all the message bits to message and thus get a code with $n = k + 1$, i.e., an $(n, n-1, 2)_2$ code. For general q, one identifies Σ with $\mathbb{Z}_q$, the additive group of integers modulo q and uses (instead of the parity check bit) the check symbol that is the sum of all message symbols over $\mathbb{Z}_q$. This gives, for every q, a $(n, n-1, 2)_q$ code.
- $d = 3$, non-trivial interesting case.

Interpolating from the first two examples, one may conjecture that a $(n, n-d+1, d)_q$ code is always possible. This turns out not to be the case and in fact $d = 3$ already gives a counterexample to this conjecture. Hamming gave codes for this case and proved their optimality. We will describe the codes in the next lecture.

6.897 Algorithmic Introduction to Coding Theory September 12, 2001

Lecture Notes on Algebra

Lecturer: Madhu Sudan *Scribe: Madhu Sudan*

These notes describe some basic algebraic structures that we will encounter during this course, including:

- Finite fields of all sizes (and shapes).
- (Univariate and multivariate) polynomials over finite fields in one or more variables.
- Vector spaces over finite fields (or Linear algebra).

Unfortunately, there is no simple order in which one can present all these objects — their presentation is interleaved for essential reasons. Polynomials are typically defined with coefficients from fields. Fields are constructed by constructing polynomial rings and then reducing them modulo irreducible polynomials. Linear algebra needs to be based on fields. But it also provides convenient ways of looking at fields. We will try to describe all these connections below. Mostly we are interested in computational and combinatorial consequences. We would like to see how to represent fields so as to perform elementary manipulations efficiently. We would also like to know if some computational problems from linear algebra can be solved efficiently. We are also interested in combinatorial questions such as: How often can a polynomial evaluate to zero? How does one prove that this can not happen too often? The notes below present answers to such questions.

2.4 Main definition

Since we are interested in polynomials over fields, it would be nice to know the basic algebraic structures which unify both fields and polynomials. Commutative rings are such structures and we define them below.

Definition 2.2 (Commutative Rings and Fields) *A* commutative ring *is given by a triple* $(R, +, \cdot)$*, where* R *is an arbitrary set containing two special elements* 0 *and* 1 *and* $+, \cdot$ *are functions mapping* $R \times R$ *to* R *satisfying the following properties for every triple* $a, b, c \in R$*:*

Associativity: *Both* $+$ *and* $\cdot$ *are associative, i.e.,* $a+(b+c) = (a+b)+c$ *and* $a \cdot (b \cdot c) = (a \cdot b) \cdot c$*.*

Commutativity *Both* $+$ *and* $\cdot$ *are commutative, i.e.,* $a + b = b + a$ *and* $a \cdot b = b \cdot a$*.*

Distributivity: $\cdot$ *distributes over* $+$*, i.e.,* $a \cdot (b + c) = a \cdot b + a \cdot c$*.*

Identities: $a + 0 = a$ *and* $a \cdot 1 = a$*.*

Additive Inverses: *For every* $a \in R$*, there exists an additive inverse* $-a \in R$ *such that* $a + (-a) = 0$*.*

If in addition, every non-zero element has a multiplicative inverse, then R *is a* field*. (I.e., for every* $a \in R - \{0\}$*, there exists an* $a^{-1} \in R$ *such that* $a \cdot a^{-1} = 1$*.)*

Often we will skip the operators + and · and simply refer to the set R as the ring (with addition and multiplication being specified implicitly). Commutative rings form the foundation for much of the elegant results of algebra and algebraic geometry. Within the class of commutative rings, one can get nicer and nicer domains (rings with nicer and nicer properties) and this culminates with the notion of a field. Informally, rings allow the operations of addition, subtraction and multiplication, while a field also allows division. We will see some of the intermediate notions later. Right now we turn to polynomials.

2.5 Polynomial rings

Given any ring R, and a symbol t (usually referred to as an indeterminate, one can create a ring $R[t]$ of polynomials over R. Such a ring inherits most of the nice properties of the underlying ring. Below is a formal definition of the ring of polynomials.

Definition 2.3 *Given a commutative ring R and indeterminate t, the set $R[t]$ has as its elements finite sequences of R, with the sequence $f = \langle f_0, \ldots, f_l \rangle$ being interpreted as the formal sum $\sum_{i=0}^{l} f_i t^i$. Addition and multiplication over $R[t]$ are defined accordingly, i.e., if $f = \langle f_0, \ldots, f_l \rangle$ and $g = \langle g_0, \ldots, g_k \rangle$ with $l \leq k$ then $f + g = \langle f_0 + g_0, \ldots, f_l + g_l, g_{l+1}, \ldots, g_k \rangle$ and $f \cdot g = \langle h_0, \ldots, h_{l+k}$ where $h_i = \sum_{j=0}^{i} f_j g_{i-j}$. For a polynomial $f \in R[t]$, given by $f = \sum_{i=0}^{d} f_i t^i$, we define its* degree, *denoted* $\deg(f)$ *to be the largest index d' such that $f_{d'}$ is non-zero.*

Proposition 2.4 *For every commutative ring R and indeterminate t, $R[t]$ is a commutative ring.*

The most natural ring of polynomials that we will encounter are the ring of polynomials over some (finite) field F, say $F[x]$. Now we can adjoin a new indeterminate y to this ring to another ring $F[x][y]$. We will use the notation $F[x, y]$ to denote such a ring whose elements are simply polynomials in two variables x and y. In particular $F[y][x] = F[x][y] = F[x, y]$. Continuing this way, adjoining m variables $x_1, \ldots, x_m$ to F for some integer m, we get the space of m-variate polynomials $F[x_1, \ldots, x_m]$. It is also possible to define this ring directly and we do so in order to define various notions of degree associated with it.

Definition 2.5 (Multivariate polynomial rings) *Given a ring R and indeterminates $x_1, \ldots, x_m$ the m-variate polynomial ring over R, denoted $R[x_1, \ldots, x_m]$ has as its elements finite sequences indexed by d-tuple of non-negative integers $f = \langle f_{i_1,\ldots,i_m} \rangle_{0 \leq i_j \leq d_j}$. The element represents the formal sum $\sum_{i_1,\ldots,i_m} f_{i_1,\ldots,i_m} x_1^{i_1} \cdots x_m^{i_m}$. Addition and multiplication are interpreted appropriately. The x_j-*degree *of f, denoted* $\deg_{x_j}(f)$, *is the largest index d'_j such that there exist indices $i_1, \ldots, i_m$ such that $f_{i_1,\ldots,i_{j-1},d_j,i_{j+1},\ldots,i_m}$ is non-zero. The* total degree *of f is the largest sum $\sum_{j=1}^{m} i_j$, among tuples $i_1, \ldots, i_m$ for which $f_{i_1,\ldots,i_m}$ is non-zero.*

We will come back to multivariate polynomials later. Right now we move on to descriptions of fields and this will need univariate polynomials.

2.6 Finite Fields

Fields are the nicest of algebraic structures. that allow all sorts of manipulations efficiently. In particular we can not only define addition and multiplication, but also subtraction ($a - b = a + (-b)$)