Discrete Math
CS 70

Contents

Discrete Math and Probability **Fall 2016**

RSA & Bijections: September 23

Table of Contents Local *Scribe: Brandon McKinzie*

RSA Alice communicates to Bob. Eve wants to figure it out. The message is

$$m = D\Big(E(m, s), s\Big) \tag{1}$$

Bijections. A **bijective** function $f : S \to T$ is defined as

- One-to-one: $f(x) \neq f(x')\forall x, x' \neq x \in S$.
- Onto: $\forall y \in T \exists x \in S$ where $f(x) = y$.

Theorem: *Two sets have same size iff there is a bijection between them.* Relation to modular arithmetic:

→ Can reverse mapping from S to T with inverse function $g : T \to S$ that maps outputs of f back to their input.

→ Consider $f(x) = x + 1 \mod m$. Is it 1-1?

→ Well, consider $g(x) = x - 1 \mod m$. It is the inverse of f, and so the function $\underline{\text{is}}$ one-to-one. **TIP:** To show a function is one-to-one, trying finding its inverse.

→ **Theorem:** If $\gcd(a, m) = 1,\ ax \neq ax'(mod m)$ for $x \neq x' \in \{0, \ldots, m-1\}$

→ Consider output space $T = \{0a \mod m, \ldots, (m-1)a \mod m\}$ and input $S = \{0, 1, \ldots, (m-1)\}$. Want to show that $S = T$.

- $T \subseteq S$, obvi.
- one-to-one mapping from S to T, so $|T| \geq |S|$ and T is superset of S.
- $\therefore S = T$.

→ Result: Since $S = T$, inverse of $a \mod m$ must exist because $1 \mod m \in T$.

Discrete Math and Probability **Fall 2016**

More RSA: September 26

Table of Contents Local

Scribe: Brandon McKinzie

Example: RSA

- Public key: $(N = 77, e = 7)$ and $d = 43$ and $p \times q = 11 \times 7$.
- $E(2) = 2^e \mod 77 = 51 \mod 77 \longrightarrow D(51) = 51^{43} \mod 77$
- 51^{43} is big. **Repeated squaring** to the rescue.
- $51^{43} = 51^{2^5+2^3+2^1+2^0} \mod 77$. Calculate each factor alone $\mod 77$ and use results from lower powers to calculate higher powers.
- How to actually do it[1]: To compute $n^e \mod p$, divide exponent e repeatedly by 2, flooring each time [Save sequence of numbers this produces]. Starting from smallest number (probably 1), successively take n raised to that power mod 7. Use past results to help future ones. The last number in the sequence is e and you'll have $n^e \mod p$.

Properties of e, d, and exponents in modular arithmetic.

- **Theorem:**

$$m^{ed} = m \mod pq \text{ if } ed = 1 \mod (p-1)(q-1) \tag{2}$$

- **Corollary:**

$$a^{k(p-1)+1} = a \mod p \tag{3}$$

- **Lemma 1**: For any prime p and any a, b:[2] $a^{1+b(p-1)} \equiv a \mod p$
- **Lemma 2**: $\forall$ primes $p, q \neq p$ and $\forall x, k$: $x^{1+k(p-1)(q-1)} \equiv x \mod pq$
- **Prime Number Theorem:** Let $\pi(N)$ denote the number of primes less than or equal to N. For all $N \geq 17$

$$\pi(N) \geq N / \ln N \tag{4}$$

[1] See Discussion 5B

[2] Think Fermat's little theorem.

Important Notes on FLT[3]

- $gcd(a, pq) = 1 \Leftarrow gcd(a, p) = gcd(a, q) = 1$
- Before expanding the exponent in $a^{(p-1)(q-1)}$, realize that it's the same as $(a^{(p-1)})^{q-1}$

[3] Ctr-f: Fermat's Little Theorem fermat Fermats little theorem

Discrete Math and Probability **Fall 2016**

Polynomials: September 28

Table of Contents Local *Scribe: Brandon McKinzie*

Polynomials in modular arithmetic $P(x) \mod p$ consist only of points in the domain $\{0, 1, \ldots, p-1\}$.

Solve intersection of polynomials by equating and solving for x, use multiplicative inverses rather than dividing. "Whole world is $\mod p$.

Theorem: There is exactly one polynomial of degree $\leq d$ ([optionally] with arithmetic modulo prime p) that **contains** $d+1$ (particular/given) points.

Secret: I'm going to give you $2+1$ points of a parabola, and the *secret* is that parabola's y-intercept.

Shamir's **k out of n scheme:**

1. Choose secret $s = a_0 \in \{0, \ldots, p-1\}$ and randomly $a_1, \ldots, a_{k-1}$.
2. Let $P(x) = a_{k-1}x^{k-1} + \cdots + a_0$.

3 . The ith shared point is $(i, P(i) \mod p$.

- **Robustness**: Any k shares gives secret.
- **Secrecy:** Knowing $\leq k-1$ points $\Rightarrow$ any $P(0)$ is possible.

Solving polynomial given enough points $\equiv$ **General linear system:**

- Given points: $(x_1, y_1), \ldots, (x_k, y_k)$, Solve...

$$a_{k-1}x_1^{k-1} + \cdots + a_0 \equiv y_1 \mod p \tag{5}$$

$$\vdots \tag{6}$$

$$a_{k-1}x_k^{k-1} + \cdots + a_0 \equiv y_k \mod p \tag{7}$$

Interpolation

- **Goal**: Want to find $P(x) = a_2x^2 + a_1x + a_0 \mod 5$ that contains $(1,3), (2,4), (3,0)$.

1. Find $\Delta_1(x)$ defined such that, for all x above except $x = 1$, $\Delta_1(x) = 0 \mod 5$ and evaluates to 1 at $x = 1$. Solution, as shown below, is to factor all $x - x_i$ together, evaluate at $x = 1$, and multiply the inverse of that to force/normalize $\Delta_1(x = 1) = 1 \mod 5$.

$$\Delta_1(x) = 3(x-2)(x-3) \mod 5 \tag{8}$$

   where 3 is inverse of $(1-3)(1-2) \mod 5$.

2. Repeat, constructing $\Delta_i(x) \forall x \in$ given points.

3. Now we have 3 polynomials that each evaluate to 1 only and 0 else for each given point. To make the $y - values$ align and get desired polynomial, compute result:

$$P(x) = y_1\Delta_1(x) + 4\Delta_2(x) + 0\Delta_3(x) \mod 5 \tag{9}$$

- **General interpolation:**

$$\Delta_i(x) = \frac{\prod_{j \neq i}(x - x_j)}{\prod_{j \neq i}(x_i - x_j)} \tag{10}$$

  where, if in modular field, you don't technically "divide" the lower product; rather, you should read that as a multiplication by denom$^{-1} \mod p$ (the multiplicative inverse).

- Construction via interpolation proves existence of unique solution.

Theorem: Any degree d polynomial has at most d roots.

Polynomial division

- Problem: Divide $4x^2 - 3x + 2$ by $(x-3) \mod 5$.
- One approach is calculating while ignoring mod, then modding at end

$$
\begin{array}{rrrr}
 & & 4x & +9 \\
\cline{2-4}
x-3 \big) & 4x^2 & -3x & +2 \\
 & -4x^2 & +12x & \\
\cline{2-3}
 & & 9x & +2 \\
 & & -9x & +27 \\
\cline{3-4}
 & & & 29
\end{array}
$$

  and answer is then $29 \mod 5 = 4$. You can also just mod 5 everything as you go, too.

- In general, dividing $P(x)$ by $(x-a)$ gives $Q(x)$ and remainder r. i.e.

$$P(x) = (x-a)\ Q(x) + r \tag{11}$$

Lemma 1: $P(x)$ has root a iff $P(x)/(x-a)$ has remainder 0.[4]

Lemma 2: $P(x)$ has d roots; $r_1, \ldots, r_d$ then[5]

$$P(x) = c(x-r_1)(x-r_2)\cdots(x-r_d) \tag{12}$$

Polynomials Discussion

1. **How many polynomials?** (I'll express my degree of certainty for each of my answers as a footnote)

   (a) Strictly speaking, $P(2)$ can only have 5 values since $GF(5)$. The number of distinct polynomials is $5 \times 5 \times 5 = 125$.[6]

   (b) The number of different pairs are $5^2 = 25$. The number of polynomials here is the number of distinct pairs of $P(i \neq 0), P(j \neq 0, i)$. This is $(5 \times 4) \times (5 \times 3) = 300$.[7]

   (c) ~~If we know k values, then we need $(d+1) - k = (d-k) + 1$ more points to uniquely determine any polynomial. The next point can have $p - k$ possible values for x, and each of those can have p possible y values, for a total of $(p-k) \times p$ unique choices for the next point alone. For subsequent choices, the number of possibles decreases by a factor of p. Therefore, the number of different polynomials we could obtain, given that we are in $GF(p)$, is~~[8]

   Error: The main error in your line of thought is that many of those polynomials would be the same one. Although polynomials are indeed definable by a set of points, many such sets can define a single polynomial. If you're going to take this approach, you need to say more like: We have (d-k)+1 points, each of which could take on p different values, so the number of *distinct* polynomials is $p^{(d-k)+1}$. Ta-da.

2. **Lagrange Interpolation**. I have an issue with their wording: Should just say "of degree 3" since it says unique. Whatever[9]

   (a) $\Delta_{-1}(x) = \frac{(x-0)(x-1)(x-2)}{(-1-0)(-1-1)(-1-2)}$

   (b) $\Delta_0(x) = \frac{(x+1)(x-1)(x-2)}{(1)(-1)(-2)}$

   (c) $\Delta_1(x) = \frac{(x+1)(x-0)(x-2)}{(2)(1)(-1)}$

   (d) $\Delta_2(x) = \frac{(x+1)(x-0)(x-1)}{(3)(2)(1)}$

[4] To prove: use 11

[5] To prove: induction on number of roots. Take advantage of Lemma 1.

[6] Certainty: 95 percent.

[7] Certainty: 90 percent

[8] Certainty: ~~95 percent~~ ~~More like 40 percent~~ 0 Percent because I know I was wrong now.

[9] Certainty:90 percent only because algrebra errors.

(e) $p(x) = 3\Delta_{-1}(x) + 1\Delta_0(x) + 2\Delta_1(x) + 0\Delta_2(x)$

3. **Secret sharing** Generate a degree 2 polynomial. Give each TA two points of it. Give each reader 1 point of it.[10]

Polynomials Note

- ***General Definitions***
  - **Polynomial division**: If we have a polynomial p(x) of degree d, we can divide by a polynomial q(x) of degree *le* by using long division. The result will be: $p(x) = q(x)q'(x) + r(x)$ where[11] $\deg(r) < \deg(p)$. Subtlety: When you rewrite p in quotient/remainder form like this, where you've explictly said what you're dividing by (q), then $\deg(r) < \deg(q)$ by definition.
- ***Property 1***: A non-zero polynomial of degree d has at most d roots.
  - **Claim 1** $\Big[p(a) = 0\Big] \Rightarrow \Big[p(x) = (x-a)q(x)\Big]$ where $\deg(p) = d$ and $\deg(q) = d - 1$.
  - **Claim 2**:[12] If $p(x)$ has d distinct roots a_i, then $p(x)$ can be written as $p(x) = c(x-a_1)(x-a_2)\cdots(x-a_d)$.
- ***Property 2***: Given $d+1$ pairs with all x_i distinct $\exists$ unique $p(x)$ of degree (at most) d such that $p(x_i) = y_i \forall i \in \{1, \ldots, d+1\}$.
- ***Counting***
  - Can specify any $d+1$ polynomial with either (a) it's coefficients (coefficient representation) a_i, or (2) a set of $d+1$ points (value representation) contained by the polynomial. Can convert rep (a) to rep (b) by evaluating at the points. Can convert (b) to (a) with lagrange interpolation.
  - IMPORTANT: When they say "how many distinct polynomials go through these.." and whatever, they apparently always assume that the x points are ordered, and you're only interested in the value of $p(x)$ at the next, as of yet unspecified, x point. Wtf?
- ***Exhaustive List of PROOF TECHNIQUES:***
  - Rewriting $p(x)$ in quotient + remainder form and exploiting properties of roots,degree of the quotient, etc.
  - Induction on the degree d of a polynomial.

[10] Certainty: 70 percent. Question seems open-ended and the wording is shit

[11] Check Piazza for followup on my question regarding this

[12] Claim 2 $\implies$ Property 1

- When thinking about number of polynomials in [. . .], remember that a polynomial can be uniquely defined by its *coefficients*. Equivalently, can think of as defined by $d + 1$ points; Note that there can be *many* such sets of $d + 1$ points that define the same polynomial.

Discrete Math and Probability **Fall 2016**

Erasure Coding: September 30

Table of Contents Local *Scribe: Brandon McKinzie*

- Lecture outline:
  - Finish polynomials and secret sharing
  - Finite fields: Abstract Algebra
  - Erasure Coding
- Note: the $d+1$ points needed to specify any polynomial must have different x values (obvi).
- ***Finite Fields***
  - Proofs of uniqueness haven't depended on whether x is reals, rationals, complex numbers. . . but not integers since no multiplicative inverses. Only works if modulo a prime p and finite element sets.
  - Can still generalize all to **finite fields**. Denote arithmetic mod p as field F_p or $GF(p)$.
  - Field def (informal): set with operations corresponding to addition/mult/div.
  - **Fact:** The number of degree d polynomials over $GF(m)$ is m^{d+1}.
- Revisit **efficiency** of polynomial secret sharing (k of n).
  - Need $p > n$ to hand out n shares.
  - For b-bit secret, need[13] $p > 2^b$.
  - **Theorem:** There is always a prime between n and $2n$.
- ***Erasure Codes*** (error correcting codes)
  - **Problem:** Want to send message with n packets. Lossy channel: loses k packets.
  - **Question:** Can you send $n+k$ packets and recover message?[14]
  - Solution Idea: Use polynomials. "*Any* n packets (out of the $n+k$) should allow reconstruction of original n packet message."[15]

[13] so you can share any secret you want. Good to choose p = $2^b + 1$.

[14] $n+k$ because, since we know k packets out of the n will be lost, we should send $n+k$ packets if we want a total of n packets to be received.

[15] Think polynomial secret sharing.

- Restated: Any n **point values** allow reconstruction of degree $n-1$ polynomial.
- **Erasure coding scheme:** Message consists of n packets denoted $m_0, m_1, \ldots, m_{n-1}$. Each m_i is packet.
    1. Choose prime $p > 2^b$ for packet size b (num bits).
    2. $P(x) = m_{n-1}x^{n-1} + \cdots + m_0 \mod p$.
    3. Send $P(1), P(2), \ldots, P(n+k)$.
- Any n of the $n+k$ gives polynomial, and thus the message.

- Comparison: Erasure codes vs. secret sharing.
    - Secret sharing: each share is size of whole secret.
    - Erasure: each share (a packet) is size $1/n$ of whole secret.

- **Example**: Erasure codes
    - Send message $1, 4, 4$ containing $n = 3$ numbers, up to $k = 3$ of which can be lost.
    - Make $P(1) = 1$, $P(2) = 4$, and $P(3) = 4$.
    - Work modulo 7 to accommodate at least $n + k = 6$ packets.
    - Can construct via linear system:[16]

$$P(1) = a_2 + a_1 + a_0 \equiv 1 \mod 7 \tag{13}$$
$$P(2) = 4a_2 + 2a_1 + a_0 \equiv 4 \mod 7 \tag{14}$$
$$P(3) = 2a_2 + 3a_1 + a_0 \equiv 4 \mod 7 \tag{15}$$
$$\tag{16}$$

so $P(x) = 2x^2 + 4x + 2$. Send packets $(1,1), (2,4), (3,4), (4, P(4)), (5, P(5)), (6, P(6))$.

Don't forget to take mods

Error Correcting Codes

- ***Erasure Errors***: (missing packets)
    - Note: I'm only writing info here that I didn't write in the previous section.
    - If each packet is a b-bit string, choose prime p to be any prime larger than 2^b.
    - Be careful to ensure that $n + k \leq p$, which is usually pretty easy.
    - If receiver only gets $n-1$ of the packets, there are exactly p polynomials of degree at most $n-1$ that agree with the received packets.

[16] Form is always the same: Plug in values for x into $a_{k-1}x^{k-1} + \cdots + a_1 x + a_0 \mod p$. Don't forget to take mod on all coefficients!

  - "This error-correcting scheme is therefore **optimal**: it can recover the n characters of the transmitted message from any n received characters, but recovery from any fewer characters is impossible."
  - To prove that the linear system always has a solution and that it is unique (which is true), hint is to show that a certain determinant is non-zero.

- ***General Errors*** (individual packets may be corrupted, but all are transmitted)
  - **DISTINCTION BETWEEN ERASURE**: Rather than the message being the coefficients of the polynomial, now want to encode as what polynomial evaluates to. fml.
  - One can still guard against k general errors by transmitting only 2k additional packets or characters[17].
  - Encoded message: $c_1, c_2, \ldots, c_{n+2k}$ where $c_j = P(j)$ for $1 \leq j \leq n+2k$. At least $n+k$ of these are received uncorrupted[18]
  - Receiver has to find $P(x)$. Know that $P(i) = r_i$ on at least $n+k$ points, where r_i denotes the ith *received* value. There are k points where $P(i) \neq r_i$ because they have been corrupted (changed) during the transmission process.
  - If $e_1, \ldots, e_k$ packets corrupted, define degree k polynomial $E(x)$ as follows, and with relationship to $P(x)$:

$$E(x) = (x - e_1)(x - e_2)\cdots(x - e_k) \tag{17}$$

$$P(i)\ E(i) = r_i\ E(i) \tag{18}$$

    for $1 \leq i \leq n+k$ where received points are of form (i, r_i). For any $i = e_i$, $E(i) = 0$. This is true because: (1) out of the $n+2k$ received, $n+k$ match the desired $P(x)$ correctly, i.e. $P(i) = r_i$ for $n+k$ points and eq 18 is obviously true. For the other points (the ones that got corrupted), $P(i)$ will be some (as of yet unknown) value that is not r_i. However, eq 18 is still true because $E(x) = 0$ for any x that was corrupted.
  - Eq 18 is really $n+2k$ linear equations with $n+2k$ unknowns.
    - Unknowns are the coefficients of $E(x)$ and $Q(x) := P(x)E(x)$.

$$Q(x) = a_{n+k-1}x^{n+k-1} + \cdots + a_1 x + a_0 \tag{19}$$

$$E(x) = (1)x^k + b_{k-1}x^{k-1} + \cdots + b_1 x + b_0 \tag{20}$$

  - Convention seems to be that, if we want to send a message of size n, we encode that message directly ***in order*** as $P(1), \cdots, P(n)$, starting for some reason at 1. We then encode the extra k parts as ordered eval of $P(n+1), \cdots, P(n+k)$.

[17] only twice as many as in the erasure case

[18] Goal is still for receiver to determine the unique polynomial $P(j)$.

- The **degree of P(x) is** $deg(P) = n - 1$. In other words, we map the desired n-point message to $(n-1)+1$ points defining the degree $n-1$ polynomial.
- **Exhaustive procedure/example**:
  - Setup: Working over $GF(7)$. Message has $n = 3$ characters.
  - **UNKNOWN TO RECEIVER:** Desired message: $3, 0, 6$. Then we need $P(x)$ uniquely defined by the points $(1,3), (2,0), (3,6)$. Therefore, $P(x)$ is degree $n - 1 = 2$ with $P(x) = x^2 + x + 1 \pmod 7$.
  - **KNOWN TO RECEIVER**: Know that $n = 3$, $k = 1$, and therefore they know that the received message of size $n + 2k = 5$ has 1 corrupted letter. They know that the following polynomials take the respective forms[19]

$$E(x) = x + e_0 \tag{21}$$

$$\begin{aligned} Q(x) &= q_3x^3 + q_2x^2 + q_1x + q_0 \qquad (22) \\ &= r_xE(x) \qquad (23) \end{aligned}$$

  - Don't forget to take mods of coefficients along the way.
  - **Q**: Given that we know $k = 1$ points will be corrupted, why is it *exactly* that we need to send $n + 2k = 5$ points? **A**: See below. Basically, it is so we can guarantee that the recovered polynomial P is unique (and the one we sent).

[19] Fact: For any polynomials P and Q, it is true that deg(PQ) = deg(P) + deg(Q).

Discrete Math and Probability **Fall 2016**

General Errors: October 3

Table of Contents Local *Scribe: Brandon McKinzie*

- Only going to write new information here.
- **Problem**:Communicate n packets $m_1 \dots m_n$ on noisy channel that corrupts $\leq k$ packets. Notice that it is $\leq k$ now.
- **Reed Solomon Code**: Make $P(x)$ of degree $n-1$.

$$P(1) = m_1; \dots ; P(n) = m_n \tag{24}$$

- Send $P(1), \dots, P(n+2k)$.
- **Why n + 2k?**
- [20]. Okay I think I know why we need $n+2k$ points. It is related to the fact that we need to guarantee the receiver will reconstruct the *unique* polynomial $P(x)$ as opposed to some other polynomial.
- Claim: If two polynomials $P(x)$ and $P'(x)$ satisfy $P(i) = r_i$ and $P'(i') = r'_i$ for their own (separate) sets of $\geq n+k$ points in the received message of size $n+2k$, then $P(x) = P'(x)$.
- Proof: We know that $\leq k$ (so at most k) packets are corrupted. This means that $P(x)$ and $P'(x)$ share *at least* n points in common (out of their respective $n+k$ point sets), i.e. where for any of these points r_j, it is true that $P(j) = r_j = P'(r_j)$. Since they are degree $n-1$ polynomials that are uniquely defined by n points, it must be that $P(x) = P'(x)$.
- Lec then goes over example of $3, 0, 6$ from the note and works through it.
- jargon: calls $E(x)$ the **error locator polynomial**.
- kind of annoyed that he keeps saying things like $P(x)$ is degree $\leq n-1$, when the note seems to just say ”equals”. Come back later and explain whether or not I should care.
- However, says deg(E) = k.

[20] Paused lec at 24:20

Discrete Math and Probability **Fall 2016**

Error Review & Infinity: October 5

Table of Contents Local *Scribe: Brandon McKinzie*

- Continues on general-error encoding example from note.
- Technique is called **Berlekamp-Welch**.[21]
- Wants to answer existence and uniqueness of $P(x)$ and $Q(x)$. Existence is easy. n+2k in n+2k unknowns can be solved so yes it exists.
- uniqueness requires proof by contradiction assuming two different solutions exist. I don't see how this is any different from my claim/proof in the previous lecture. Time: 17:00. Identical proof as in note though regarding EQ = Q'E'.
- **Infinity an Uncountability**. Proof techniques are enumeration and constructing bijections.
- **Countably infinite**: A set is countably infinite if its elements can be put in one-to-one correspondence with the set of natural numbers.
- Determining if two sets are **same size**.
  - Make function $f : A \to B$.
  - Show f is one-to-one, defined as $\forall x, y \in A, x \neq y \implies f(x) \neq f(y)$. Show f is onto, i.e. $\forall s \in B, \exists c \in A, \ s = f(c)$.
  - **Isomorphism principle:** If there exists bijection $f : A \to B$, then $|A| = |B|$ (the cardinality of A is the same as cardinality of B).
- ***Number of subsets of $S = \{a_1, \ldots, a_n\}$.***
  - Equal to number of binary n-bit strings. In other words, there exists a bijection f : subsets $\to$ n-bit strings.
  - **Proof**: For some subset x of $\{a_1, \ldots, a_n\}$, define

$$f(x) = \Big(g(x, a_1), \ldots, g(x, a_n) \Big) \tag{25}$$

$$g(x, a) = \begin{cases} 1 & a \in x \\ 0 & \text{otherwise} \end{cases} \tag{26}$$

[21] This technique, i guess, *uses* reed-solomon code. Whatever.

  - Example: $S = \{1, 2, 3, 4, 5\}, x = \{1, 3, 4\}$. Then $f(x) = (1, 0, 1, 1, 0)$.
  - The cardinality of the **Power set** of S is

$$|\mathcal{P}(S)| = |\{0, 1\}^n| = 2^n \tag{27}$$

    which is the number of n-bit binary strings, and *therefore* the number of subsets is also 2^n since f is a bijection.

- ***Infinity*** [38:00]
  - Natural numbers = "the counting numbers".
  - Any set S is **countable** if there exists a bijection between S and *some subset of* $\mathbb{N}$.
  - If the subset of $\mathbb{N}$ is finite, then S has **finite cardinality**. If infinite subset then countably infinite and say it has "the same cardinality as $\mathbb{N}$".
  - Note, if a bijection exists from A to B, then we automatically know one exists from B to A because function inverse guaranteed.
  - Comparing cardinality of $\mathbb{Z}$ to that of $\mathbb{N}$: Define $f : \mathbb{N} \to \mathbb{Z}$ where

$$f(n) = \begin{cases} n/2 & \text{if n even} \\ -(n+1)/2 & \text{odd} \end{cases} \tag{28}$$

    and check (1) one-to-one by proof by cases on $x, y \in \mathbb{N}$ and combinations of one/both being even/odd, and (2) onto by for $z \in \mathbb{Z}$, cases where its negative/nonnegative and showing that it's pre-image would be $\in \mathbb{N}$.

Discrete Math and Probability **Fall 2016**

Countability & Computability: October 7

Table of Contents Local *Scribe: Brandon McKinzie*

- **Lists** have natural ordering property where position of item in list is a natural number. One way of showing if list is countable is by **enumeration** of elements in that set. Enumerability $\equiv$ countability.
- When enumerating, need to be careful that each element has a *finite* specified position in the list.
- **Lemma:** Any subset T of a countable set S is countable.
- All countably infinite sets have the same cardinality.
- For finite sets S_1 and S_2, cardinality of $S_1 \times S_2$ is $|S_1| \times |S_2|$.[22]
- **Cantor's diagonalization** for analyzing the cardinality of $\mathbb{R}$.
  - Try enumerating. View as a table. Construct a number along the diagonal: digit i is 7 if row i's ith digit is not 7, 6 otherwise. Implies that the diagonal number is not in the list[23], but it is somehow in $\mathbb{R}$, which is a **contradiction**.
  - Note: We can say that, *since* the numbers in the range $[0, 1]$ are uncountable, and since they are a subset of $\mathbb{R}$, that $\mathbb{R}$ is uncountable.
- Can show a bijection between two uncountable sets, e.g. $f : \mathbb{R}^+ \to [0, 1]$.

  Computability:

  - **Barber Paradox**. Why is this supposed to be interesting? Proof by cases leads to contradiction.
  - Any definable collection is a set. Example:

$$\exists Y \forall x (x \in Y \iff P(x)) \tag{29}$$

    and "y is the set of elements that satisfies P(x)." Can apply to barber paradox.
  - Key notion here is **self-reference**.

[22] Note: seems to suggest that $\mathbb{N} \times \mathbb{N}$ is undefined. But countable... Check.

[23] If it were, say, the jth element of the list, then by definition its jth element could not be its jth element. Don't hurt yourself, it's simple.

- The **halting problem**: write program that checks if other program halts: $HALT(P, I)$ where P is a program, I is input. Determines if $P(I)$ [P run on I] halts or loops forever. Program itself is some text string, which is why it (a program) can be fed as input to a program. *This enables self-reference in computation. One program executing on itself is possible.*

- HALT does **not** exist. Proof: Assume there is a program called HALT and a program TURING(P).

  1. If HALT(P, P) = "**halts**". then define Turing such that it goes into an infinite loop.
  2. Otherwise, Turing halts immediately. It basically does the opposite.
  3. Assumptions: there is a program HALT and text that are both the programs TURING and HALT.
  4. Question: Does Turing(Turing) halt? Proof by cases.
     - Assume it does halt. Then HALT(Turing, Turing) = halts. Then we TURING(turing) loops forever. Contradiction.
     - Assume it loops forever. Then HALT(turing, turing) $\neq$ halts. Then Turing(turing) halts. Contradiction.

  and so program HALT does not exist.

Discrete Math and Probability **Fall 2016**

Counting: October 10

Table of Contents Local *Scribe: Brandon McKinzie*

Computability Wrap-up:

- Goes over Turing machine. Infinite tape with characters. Can be in a state, read a character. More left/right and read/write charcter.
- Universal turing machine: tape could be a description of a ... turing machine.
- Church proved equivalent theorem about **Lambda calculus**.
- Godel proved his **incompleteness theorem**: any formal system is either inconsistent [false statement can be proven] or incomplete [the is no proof for some sentence in the system]. Godel also showed every statement corresponds to a natural number. wtf.

Counting:

- Related to questions of the form "How many ... given [condition]?"
- **Product Rule**: Objects made by choosing from n_1 then n_2, ..., then n_k, then the number of objects is

$$n_1 \times n_2 \times \cdots \times n_k \tag{30}$$

- **Permutations:** General case is "how many different samples of size k from n numbers **without replacement**." Answer:

$$n \times (n-1) \times \cdots \times (n-k+1) = \frac{n!}{(n-k)!} \quad [{}^nP_k] \tag{31}$$

- If order doesn't matter, count ordered objects and then divide by number of orderings[24]. Have n objects and want to choose k?

$$\frac{n!}{(n-k)! \times k!} = \binom{n}{k} \tag{32}$$

[24] Calls this "second rule of counting." The first rule is the produce rule.

- Suppose sampling with replacement but order doesn't matter. Famous example is **Stars and bars**: *How many ways can Bob and Alice split 5 dollar bills?* For each of 5 dollars pick Bob or Alice (2^5), "then divide out order??" Let a denote number of dollars for Alice, similarly for bob such that $a + b = 5$, or in more general case $a + b = k$. There are apparently $k + 1$ ways.
- General case[48:00]: If want to split up between, say, $k = 3$, can split with **stars and bars**: $* * \mid * \mid * *$. Each sequence of stars and bars $\implies$ split.
- **Counting rule:** If there is a 1-to-1 mapping between two sets, they have the same size.
- **Sum rule**: For disjoint S and T, $|S \cup T| = |S| + |T|$.
- **Inclusion/Exclusion**: $\forall S,\ T,\ \ |S \cup T| = |S| + |T| - |S \cap T|$.

General stars and bars: k stars $n - 1$ bars. There are

$$\binom{n+k-1}{n-1} = \binom{(n-1)+k}{n-1} = \binom{n+k-1}{k} \tag{33}$$

...in other words, $n + k - 1$ positions from which to choose $n - 1$ bar positions. WIKIPEDIA VERSION:

Theorem one

$\forall n, k \in \mathbb{Z}^+$: the number of k-tuples of **positive** integers, whose sum = n, is $\binom{n-1}{k-1}$ Translation: If each person must get something, there are $\binom{n-1}{k-1}$ ways to split n stars up among $k + 1$ people.

Theorem two

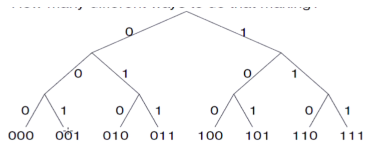
$\forall n, k \in \mathbb{Z}^+$: the number of k-tuples of **non-negative** integers, whose sum = n, is $\binom{n+k-1}{k-1} = \binom{n+k-1}{n}$. Translation: In general case, there are $\binom{n+k-1}{k-1} = \binom{n+k-1}{n}$ ways to split n stars up among $k + 1$ people.

Since the above is confusing, here is the clearest possible way I can state it: If asked, how many ways to split up n [things] among k [people]? The answer is always

$$\binom{n+k-1}{k-1} \tag{34}$$

Examples

- How many 3-bit strings?



8 leaves which is 2 × 2 × 2.

- How many outcomes for k coin tosses? 2^k.

- How many 10 digit numbers? 10^k.

- How many n digit base m numbers? m^n.

- How many **functions** f mapping S to T? $|T|^{|S|}$, because $\forall s_i \in S$ have $|T|$ choices for $f(s_i)$.

- How many **polynomials** of degree d modulo p? p^{d+1} coefficient choices and/or choices of the unique $d+1$ points (both lead to same answer).

- How many 10 digit numbers *without repeating a digit?*. $10 \times 9 \times \cdots \times 1 = 10!$.

- How many 1-to-1 functions from $|S|$ to $|S|$? $|S|!$.

- How many poker hands? Number of orderings for a given poker hand is 5!, so answer is $52!/(5!47!)$.

- How many different 5 star and 2 bar diagrams? 7 positions in which to place the 2 bars. $\binom{7}{2}$ ways splitting 5 dollars among 3 people.

Combinatorial Proofs

Let $|A| = n$. Prove $\binom{n}{k+1} = \binom{n-1}{k} + \binom{n-2}{k} + \cdots + \binom{k}{k}$.

- LHS. Number of subsets of size $k+1$ from set of size n.

- RHS. Ask yourself: What's another way I could find all subsets of size $k+1$?
  - Well, I could count the number of subsets that include the element $\min(A)$. This means I have k elements out of the remaining $n-1$ to choose from, i.e. $\binom{n-1}{k}$. That takes care of all subsets including $\min(A)$.
  - What about subsets where the smallest element is the *second-smallest* element in A?[25] Now we have k elements out of the remaining $n-2$ to choose from, i.e. $\binom{n-2}{k}$, and the pattern emerges.

[25] Notice that all such subsets do not include *any* of the subsets counted in the previous bullet point.

- Therefore, the jth term on the RHS represents the number of subsets of size k where the smallest item in the (jth) subset is the jth smallest element in A.

Textbook (Rosen) Notes

- If $A_1, \ldots, A_m$ are finite sets, then number of elements in the Cartesian product of these sets is

  Equation

  $$|A_1 \times \cdots \times A_m| = |A_1| \cdots |A_m| \tag{35}$$

- An **r-combination** of elements of a set is an unordered selection of r elements from the set. Thus, an r-combination is simply a subset of the set with r elements. The number of r-combinations from a set of n elements is often denoted as $\binom{n}{r}$.

Binomial theorem and related stuff.

Binomial Theorem

$$(x+y)^n = \sum_{j=0}^{n} \binom{n}{j} x^{n-j} y^j \tag{36}$$

which can be proved by counting the number of $x^{n-j}y^j$ terms. Since we have n products of sums $x+y$, we would need to *choose* $n-j$ x's from the n sums. But this is just $\binom{n}{n-j} = \binom{n}{j}$. Damn.

Corollaries to the Binomial Theorem

$$\sum_{k=0}^{n} \binom{n}{k} = 2^n \tag{37}$$

$$\sum_{k=0}^{n} (-1)^k \binom{n}{k} = 0 \tag{38}$$

$$\sum_{k=0}^{n} (2)^k \binom{n}{k} = 3^n \tag{39}$$

where all of these can be proven very easily using the Binomial Theorem (Hint: Think about what each implies about the values of x and y).

***Other useful Identities*.**

Pascal's Identity and Vandermonde's Identity

$$\binom{n+1}{k} = \binom{n}{k-1} + \binom{n}{k} \quad \mathit{PASCAL} \tag{40}$$

$$\binom{m+n}{r} = \sum_{k=0}^{r} \binom{m}{r-k}\binom{n}{k} \quad \mathit{VAND.} \tag{41}$$

Note: It seems pretty popular to think about $\binom{n}{k}$ as "the number of bit strings of length n containing k ones."

Discrete Math and Probability **Fall 2016**

Midterm 2 Review: October 22

Table of Contents Local *Scribe: Brandon McKinzie*

Bijections/Sets

→ [**FA15.4.a**] If need bijection $f : (1, \infty) \rightarrow (0, 1)$, don't get too caught up with how any particular number should be mapped. Instead, think about what functions *over the given domain* map a positive real number above 1 to the interval 0, 1. The function they use is $1/x$. Then show it's one-to-one and onto in order to prove bijection.

→ To check if two sets A, B are *equal* (not just same size), check both that $A \subseteq B$ and $B \subseteq A$.

RSA/Modular Arithmetic

→ **Q** [**FA15.1.d**]: Given just N and e, how to quickly find d? **A:** You can't unless you know the factors of N.

→ **Q** [**FA15.1.e**]: What is the general meaning of 'signature of x'?

→ Write everything here about meaning of *relatively prime to [a number]* and what it implies/how to think about it.

- ⋆ Definition: a rel prime to b iff $\gcd(a, b) = 1$
- ⋆ Means that the two numbers share no common factor.
- ⋆ Multiplicative inverse of a exists mod b and vice versa.[26]
- ⋆ If inverse exists, then it is *also* relatively prime with the other number. This should be obvious because the inverse of the inverse exists (it is the original number) which means it must be rel prime.
- ⋆ **GENERAL FLT**: For any modulus n and any integer a coprime to n,

$$a^{\varphi(n)} \equiv 1 \pmod{n} \tag{42}$$

where $\varphi(n)$ denotes **Euler's totient function** which counts the number of integers

[26] Does it matter if one number is bigger than the other? **A: No it does not matter.**

between 1 and n *that are coprime with* n.

$$\varphi(n) = n \prod_{p|n} \Big(1 - \frac{1}{p}\Big) \tag{43}$$

$$\gcd(m, n) = 1 \implies \varphi(mn) = \varphi(m)\varphi(n) \tag{44}$$

$$\varphi(p^k) = p^k \Big(1 - \frac{1}{p}\Big) \tag{45}$$

→ **Chinese Remainder Theorem**: a theorem of number theory, which states that, if one knows the remainders of the division of an integer n by several integers, then one can determine uniquely the remainder of the division of n by the product of these integers, under the condition that the divisors are pairwise coprime.

→ Any RSA scheme is considered broken/breakable if knowing N allows one to deduce the value of $(p-1)(q-1)$, where you're only given N, not its factors. This is because, equivalently, breaking RSA means figuring out the value of $d = e^{-1} \pmod{(p-1)(q-1)}$.

- Also, unbreakable means at least as difficult as ordinary RSA. So, if you can make a bridge between the problem you're doing and the problem of ordinary RSA (given just N, e, find d), that suffices.
- **Q**: How to prove correctness of RSA?

Polynomials/Modular Arithmetic

→ Walkthrough of how smart person would approach "What is $3^{240} \pmod{77}$"

1. *Oh, 77 is 11 × 7, so I could think of as* $\pmod{77} = \pmod{pq}$.
2. *From things theorems like 2, I know that*

$$x^y \pmod{pq} \equiv_{pq} (x^y)^{1 \pmod{(p-1)(q-1)}} \equiv_{pq} x^{y \pmod{(p-1)(q-1)}}$$

3. *So I can rewrite and solve as*

$$3^{240} \equiv_{pq} 3^{240 \pmod{(10-1)(7-1)}} \equiv_{pq} 3^{240 \pmod{6)0}} \equiv_{pq} 3^0 \equiv_{pq} 1$$

→ [**FA15.2.b**] Write about polynomial intersections here. $P(x) - Q(x) = 0$ is max deg 4, so it has 4 roots, answer is 4.

→ Note: $n + x \equiv_n x \pmod{n}$.

→ Note: Modulo over polynomials should be *prime*.

→ General errors. Remember that for $E(x) = \prod_i (x - err_i)$, the err_i is an x value (!!!) and NOT a y value. It is an index.

Counting

→ **Stars and Bars**. If k bars and n stars, $\binom{n+k}{k} = \binom{n+k}{n}$ ways. I promise.

→ **Bins**. Convert to stars and bars problem with (numBins - 1) bars.

→ Don't forget the general sum rule: $\forall S,\ T,\quad |S \cup T| = |S| + |T| - |S \cap T|$.

Computability

→ **Q** [**FA15.5.a**] Meaning of "undecidable"? **A:** an undecidable problem is a decision problem for which it is known to be impossible to construct a single algorithm that always leads to a correct yes-or-no answer.

→ [**FA15.5.a**] Master: halting problem, programs that return themselves.

→ **Quine**: A program that prints itself.

```
Print out the following sentence twice, the second time in quotes:
‘‘Print out the following sentence twice, the second time in quotes:’’
```

⇝ We can always write quines in any programming language.
⇝ Another example:

(Quine "s") (s "s")

which, if passed in $s = Quine$, will output (Quine "s"), which means we run the string s (now interpreted as a program) on itself.

→ **Theorem:** *Given any program $P(x,y)$, we can always "convert it" to another program $Q(x)$ such that $Q(x) = P(x,Q)$, i.e. Q behaves exactly as P would if its second input is the description of the program Q.*

→ **Halting Problem**.

- Proof relies on (1) self-reference, and (2) fact that we can't separate programs from data.
- Problem: Given the **description P of a program** and its input, write a program `TestHalt` that behaves as:

$$TestHalt(P,x) = \begin{cases} \text{“yes”} & \text{if P halts on input x} \\ \text{“no”} & \text{if P loops on input x} \end{cases} \tag{46}$$

- Proof: Try feeding program P the input P (itself as bitstring). Define

```
def Turing(P):
  if TestHalt(P, P) == "yes":
    loop forever
  else:
    halt
```

and consider behavior of Turing(Turing). It leads to proof by contradiction that TestHalt(P, P) cannot exist, since that was our main assumption this whole time.

→ **Reduction/TestEasyHalt** [**HARD**]

- ☞ General pattern to recognize for problem-solving: Try **reducing** (changing) the problem into the general form of the halting problem.

General Tips

- ⋆ Repeated squaring: It's easier if you write within the equation as you go. Example:

$$x^{16} \pmod y = (x^2)^8 \pmod y = ((x^2)^2)^4 \pmod y = \cdots$$

- ⋆ Write down cardinality of as many sets as possible and whether or not they are countable.
- ⋆ Rational numbers have decimal expansions that are either finite or periodic.

Discrete Math and Probability **Fall 2016**

Bayes' Rule, Independence, Mutual Independence: October 19

Table of Contents Local *Scribe: Brandon McKinzie*

Note: This lecture (23) corresponds to ***Note 14*** *(Combinations of Events).*

Conditional Probability Review.

- A and B positively correlated: $Pr(A|B) > Pr(A)$; Negatively correlated if $Pr(A|B) < Pr(A)$
- $B \subset A \implies$ A and B positively correlated.
- $A \cap B = \emptyset \implies$ A and B negatively correlated.
- Total probability rule: $Pr(B) = Pr(A \cap B) + Pr(\bar{A} \cap B)$.
- **True**: If $Pr(A|B) > Pr(A)$, then $Pr(B|A) > Pr(B)$.
- **False**: If $Pr(C|A) > Pr(C|B)$, then $Pr(A|C) > Pr(B|C)$.
- See lec at [18:00] for square-space probability illustration.

Independence. Two events A and B are independent if any of the (equivalent) statements hold:

$$Pr(A \cap B) = Pr(A)Pr(B) \tag{47}$$
$$Pr(A|B) = Pr(A) \tag{48}$$
$$Pr(B|A) = Pr(B) \tag{49}$$

Examples:

→ When rolling two dice, one blue and one red, define events A = sum is 7 and B = red die is 1. **Q**: Are these independent events?[27] **A**: Yes.

→ Now define events A = sum is 3 and B = red die is 1. **Q**: Are these independent events? **A**: no.

[27] I'm predicting yes they are, ~~because having the sum be seven doesn't tell us any information about which colored die was what.~~ You were right but *for the wrong reason.* The sum does actually give us some info in general, but the only reason it doesn't here is because it is 7, which is a possibility regardless of what the first die says. See the next example, which shows a case where they are not independent.

Mutual Independence. Events $\{A_j,\ j \in J\}$ are mutually independent if

$$Pr(\cap_{k \in K}) = \prod_{k \in K} Pr(A_k) \tag{50}$$

for all finite $K \subseteq J$.

- **Theorem:** *If all K_n are pairwise disjoint finite subsets of J, then events V_n defined by $\{A_j, j \in K_n\}$ are mutually independent.* Proof is in Note 25 example 2.7.
- **Fact:** $(A, B, C, \ldots, G, H$ mutually indep. $) \implies (A, B^C, C, \ldots, G^C, H$ mutually indep. $)$. **Inductive Proof**. Need to show eq 50 holds regardless of which events we take complement of or not. Proceed by induction on n, *the number of complements*. Base case For $n = 0$, this is the normal definition of mutual independence. Hypothesis: Assume true for n. Step. For $n + 1$, need[28]

  $$A \cap B^c \cap C \cap \cdots \cap G^c \cap H = X \cap H \ \setminus \ X \cap G \cap H \tag{51}$$

  where $X := A \cap B^c \cap C \cap \cdots \cap F$. Recognize that $X \cap G \cap H \ \subset \ X \cap H$.

[28] Note: The **relative complement** of A with respect to B, denoted as $A \setminus B$, is defined as all objects that belong to A and not to B.

Discrete Math and Probability **Fall 2016**

Balls, Coupons, and Random Variables: October 26

Table of Contents Local *Scribe: Brandon McKinzie*

Note: This lecture (25) corresponds to ***Note 16*** *(Random Variables, Distribution, Expectation).*

Balls in bins. Have n bins and $m < n$ balls. Randomly (uniformly) throw balls, one by one, into bins. **Q**: What is the probability that after some m balls, that we don't have any collisions? (no two balls in same bin)[29]. Result:

$$Pr(\text{no collision}) \approx e^{-\frac{m^2}{2n}} \tag{52}$$

Coupons. Say there are large $n >> 1$ number of unique possible baseball cards. Each cereal box has a random card. You buy m boxes. The probability that you don't get a particular card (approx), and also a bound on the probability that you miss at least one card is shown below.

$$Pr(\text{miss a specific card}) \approx e^{-\frac{m}{n}} \tag{53}$$

$$Pr(\text{miss at least one card}) \leq ne^{-\frac{m}{n}} \tag{54}$$

Random Variables. Define random variable X to be the function $X : \Omega \to \mathbb{R}$ that assigns the value $X(\omega)$ to outcome ω. For more, see portion of section **??** on random variables. The **expected value** of a (discrete) random variable X is

$$\begin{aligned} \mathbb{E}[X] &= \sum_a a \; Pr(X = a) && (55) \\ &= \sum_\omega X(\omega) \; Pr(\omega) && (56) \end{aligned}$$

where subscript a denotes all possible values of X, and ω denotes all possible outcomes in the sample space.

[29] Similar to having m people in room and wanting probability that no two people have same birthday ($n = 365$)

This suggests that if we repeat an experiement a large number N of times and denote $X_1, \ldots, X_n$ as the successive values we get, then

$$\mathbb{E}[X] \approx \frac{\sum_i X_i}{N} \tag{57}$$

Summary. If asked on final the definition of random variable X, write the following:

> X is a real-valued function of the outcome of a random experiment.

and some useful properties:

- $Pr(X = a) := Pr(X^{-1}(a)) = Pr(\{\omega | X(\omega) = a\})$ "The probability that X takes on the value a = The probability that random outcome of experiment happens to map into a"
- $Pr(X \in A) := Pr(X^{-1}(A))$.
- The **distribution** of X is the list of possible values and their probability:

  $$\{(a, Pr(X = a)),\ a \in \mathcal{A}\}$$

  where A is the range of X.

Discrete Math and Probability **Fall 2016**

Expectation; Geometric and Poisson: October 28

Table of Contents Local *Scribe: Brandon McKinzie*

Lecture Overview:

→ Review Random Variables.
→ Expectation.
→ Linearity of Expectation.
→ Geometric Distribution.
→ Poisson Distribution.

Review of Random Variables. Note that definition of the inverse of a random variable is defined as

$$\forall a \in \mathbb{R} \quad X^{-1}(a) := \{\omega \in \Omega | X(\omega) = a\} \tag{58}$$

and the probability the $X = a$ is defined as $Pr(X = a) = Pr(X^{-1}(a))$. Functions of random variables: Let X, Y, Z be random variables on Ω and $g : \mathbb{R}^3 \to \mathbb{R}$. Then $g(X, Y, Z)$ is the random variable that assigns the value $g\Big(X(\omega), Y(\omega), Z(\omega)\Big)$ to ω.

Expectation. The expectation of a random variable X is

$$\mathbb{E}[X] = \sum_a Pr(X = a)a \tag{59}$$

Example of an **indicator**: Let A be an event. The random variable X defined by

$$X(\omega) = \begin{cases} 1 & \text{if } \omega \in A \\ 0 & \text{if } \omega \notin A \end{cases} \tag{60}$$

is called the **indicator of the event A**. Note that $Pr(X = 1) = Pr(A)$ and $Pr(X = 0) = 1 - Pr(A)$. Hence $\mathbb{E}[X] = Pr(A)$. **Equivalent Notation:** Sometimes also denote indicators like

$$1\{\omega \in A\} \text{ or } 1_A(\omega) \tag{61}$$

Linearity of Expectation. The mean value of a linear combination of random variables is a linear combination of the mean values:

$$\mathbb{E}[a_1 X_1 + \cdots + a_n X_n] = a_1 \mathbb{E}[X_1] + \cdots + a_n \mathbb{E}[X_n] \tag{62}$$

The common pattern I'm seeing for using linearity is the following: Some generic situation where we might be tempted to let X denote the number of [blank], but the distribution of $Pr(X = \text{[blank]})$ is complicated for taking expectations. Try instead: Let $X = X_1 + \cdots + X_n$, where each X_i represents ith occurrence of [blank] (in which case it is 1) or it is zero if ith occurrence doesn't happen. Useful: Assume A and B are disjoint events. Then

$$1_{A \cup B}(\omega) = 1_A(\omega) + 1_B(\omega) \tag{63}$$
$$1_{A \cup B}(\omega) = 1_A(\omega) + 1_B(\omega) - 1_{A \cap B}(\omega) \tag{64}$$

where the second equation is the more general case where we don't know A and B are disjoint.

- Recall that the expectation of a function of X is given by the following (equivalent) formulas:

$$\mathbb{E}[g(X)] = \sum_x g(x) Pr(X = x) \tag{65}$$
$$= \sum_\omega g\big(X(\omega)\big) Pr(\omega) \tag{66}$$

- **Monotonicity**. Let X, Y be two random variables on Ω. We write $X \leq Y$ if $X(\omega) \leq Y(\omega)$ for all $\omega \in \Omega$. (a) If $X \geq 0$ then $\mathbb{E}[X] \geq 0$; and (b) If $X \leq Y$, then $\mathbb{E}[X] \leq \mathbb{E}[Y]$.

Geometric Distribution. Example: flip a coin *until* we get heads (H), where $Pr(H) = p$. Our sample space is then $\Omega = \{\omega_n, \;\; n = 1, 2, \ldots\}$ where $\omega_i = T_1, T_2, \ldots, T_{i-1}, H$. Let $X(\omega_n) = n$ be the number of flips required to get the first H. This random variable has a **geometric distribution**, defined as

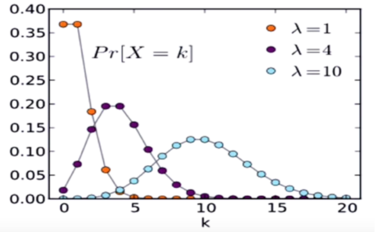
$$Pr(X = n) = p(1-p)^{n-1} \qquad n \geq 1 \tag{67}$$

where $\mathbb{E}[X] = 1/p$. Distribution is *memoryless*:

Theorem: *Let X be $G(p)$. Then, for $n \geq 0$, $Pr(X > n) = (1-p)^n$, and*

$$Pr(X > n + m \mid X > n) = Pr(X > m) \qquad m, n \geq 0 \tag{68}$$

Poisson Distribution. Experiment: Flip a coin n times. Told that coin is such that $Pr(H) = \lambda/n$. Let random variable $X = Binom(n, \lambda/n)$ be number of heads. The **Poisson distribution** of X is the distribution of X for "very large n".



We expect $X << n$. For $m << n$, one has

$$Pr(X = m) = \binom{n}{m} p^m (1-p)^{n-m} \quad (69)$$

$$\approx \frac{\lambda^m}{m!}\left(1 - \frac{\lambda}{n}\right)^n \quad (70)$$

$$= \frac{\lambda^m}{m!} e^{-\lambda} \quad (71)$$

where $\lambda > 0$ and $m \geq 0$. The mean value is $\mathbb{E}[X] = \lambda$.

> If you count the number of times something rare happens, it tends to have a Poisson distribution.

Coupon Collector's Problem. There are n coupons to collect, each equally likely, and we sample with replacement. What is the probability that more than t sample trials are needed to collect all n coupons?

- Asymptotic. The expected number of trials grows as $\mathcal{O}(n \log n)$.
- Key ideas: It takes very little time to collect the first few coupons, and much longer to collect the the last few. Idea: split the total time into n intervals (for problem of n coupons) where the expected time *can* be calculated.