Mathematics 55– Spring 2005 – M. Christ
Counting[1]

This handout catalogues the main counting techniques, problems, and examples we've discussed (as of 4/8/2005) in the course. (A second part, covering material in Chapter 6, will be provided later in the semester.)

My intent is neither to review details nor to summarize, but to help you to better organize what we've studied. I hope that this will help you to take a problem, recognize how it fits into the framework of the subject, and select the relevant technique and draw analogies to examples. For details on how to analyze each item, see the relevant sections of text, lecture notes, and problem set solutions. I also hope that this outline will help you to organize your studying.

The most basic counting techniques and formulas:

• **Sum rule**. Split a counting task into mutually exclusive subtasks. Count each, and sum.

• **Product rule**. Often applies to problems where alternative possibilities are considered in sequence; see various examples about poker hands, e.g. counting 5 card poker hands with two distinct pairs but not three of any one kind. The product rule is often useful in calculating probabilities related to independent events.

The sum and product rules are often used together. We've discussed several examples involving card games, especially counting the number of poker hands of various types. A good test of your skill for this type of problem is the question on Martian poker on the second midterm exam.

• **Inclusion-exclusion principle** for two sets. (We'll learn a more sophisticated version in Chapter 6, and some applications.)

• **Permutations**: n distinct objects can be placed in order in exactly $n!$ different ways. Equivalently, this is the number of ways of choosing n objects in order, with no repetitions.

More generally, the number of ways of choosing r objects in order from a set of n objects, without repetitions, is $n(n-1)\cdots(n-r+1) = \frac{n!}{(n-r)!}$.

• **Combinations**: The number of ways $C(n,r)$ of choosing r elements from a set of n distinct elements, *without regard to the order in which the r elements are chosen*, is the *binomial coefficient* $\binom{n}{r} = \frac{n!}{r!(n-r)!}$. Here no repetitions are allowed.

This is the same as the number bit strings of length n having r ones; to set up a one-to-one correspondence linking the two problems, think of a 1 in the k-th place as encoding the fact that the k-th object (out of n) is among those chosen.

Our derivation of the formula $C(n,r) = \binom{n}{r}$ established a link between combinations and permutations.

Next come two more sophisticated topics, both involving selections with repetitions allowed:

• **Permutations with repetition**: (i) The number of ways to choose r elements *in order*, with repetitions allowed, from a set of n elements is n^r. This is the same as: (ii) The number of ways of putting n distinguishable objects into r boxes. (Number the objects $\{1, 2, \cdots, n\}$, and likewise encode the boxes with letters from some alphabet. There's a one-to-one correspondence between strings of n symbols from this alphabet, and ways of putting the objects into boxes. The string *acbbaa* indicates that object 1 goes into box a, objects 2 goes into box c, objects $3, 4$ go into box b, and objects $5, 6$ go into box a.) (iii) The number of strings of length r from an alphabet of n symbols. (The k-th symbol in the string tells us which box to put the k-th object into. Clearly there are n^r such strings, by the product rule, since there are n possibilities for the first symbol, n for the second, n for the third, et cetera.)

Beware: This counting problem sounds similar to others that are actually quite different, e.g. putting *in*distinguishable objects into distinguishable boxes. How to keep it all straight? I picture basic situations in my mind, and remember the pictures, not a list of formulas. Bit strings, including stars/bars, are helpful for me.

• **Combinations of r objects taken from a set of n elements**, with any number of repetitions allowed. There are $C(n + r - 1, r)$ such combinations. We derived this formula by placing these combinations into one-to-one correspondence with (ii) sequences of $n - 1$ bars and r stars. (The stars represent the r objects, and the bars can be thought of as walls between the boxes; it takes only $n - 1$ walls to separate n boxes arranged in a row.) Thus the number of combinations of r objects taken from a set of n elements, with repetitions allowed, equals the number of bit strings with $n-1$ ones and r zeros (replace bars by 1s and zeros by stars). (iii) It also equals the number of ways of distributing r *indistinguishable* objects into n *distinguishable* boxes (think of the boxes as being in one-to-one correspondence with the n objects, and think of the number of objects put into a given box as encoding the number of times the corresponding object is chosen). (iv) And it equals the number of ways that r can be expressed as $x_1 + x_2 + x_3 + \cdots + x_n$ where each x_j is a nonnegative integer. (Here order matters; the expression $20 = 5+7+8$ is considered to be different from $20 = 8 + 7 + 5$. In terms of objects in boxes, think of x_1 as the number of objects placed into box number 1, of x_2 as the number of objects placed into box number 2, and so forth.))

Two important variants:

• Number of ways to **distribute n distinguishable objects into k distinguishable boxes, putting exactly n_i objects into box i**, equals $\frac{n!}{n_1!n_2!\cdots n_k!}$. Example: The strings that can be formed out of MISSISSIPPI, using each letter exactly as many times as it appears, can be placed into one-to-one correspondence with the number of ways of placing 11 distinguishable objects into 4 distinguishable boxes, with the constraint that 1 object goes into the first box, 4 go into the second box, 4 go into the third box, and 2 go into the fourth box. (Name the boxes M, I, S, P in that order. Number the objects $1, 2, 3, \cdots, 11$. Given a string, the letter in the k-th position tells us which box to put the k-th object into.) This is same as number of *permutations of n distinguishable objects, where there are n_i objects of each type i*, with $\sum_i n_i = n$.

• **Bit strings**. Several problems involving bit strings have already been mentioned. A variant: Number of bit strings of length n, having k ones, with no consecutive ones. Such bit strings can be placed into one-to-one correspondence with bit strings of length $n-(k-1)$ having k ones; given a string of length n of this type, delete the zero immediately to the right of each of the 1s, *except* the last. (Of course there are many variants of this problem involving strings of symbols from alphabets with more than two symbols, and involving exclusion of other configurations besides two consecutive occurrences of one particular symbol.)

Two miscellaneous important topics:

• **Identities for binomial coefficients**. We learned some particular identities. Some of those were special cases of the *binomial theorem*. Others had direct connections with counting problems, and this sometimes led to *combinatorial proofs* of identities; e.g. Pascal's and Vandermonde's identities, Theorem 4 page 332, Corollary 4 page 332, and the chairperson identity. (Because binomial coefficients arise so frequently, identities involving them are often useful in solving counting problems. Example: The expected number of successes in n Bernoulli trials is naturally expressed as $\sum_{k=0}^{n} k\binom{n}{k}p^k(1-p)^{n-k}$; the chairperson identity can be used to evaluate this sum.)

• **Pigeonhole principle**. This arises primarily in problems where we are not aiming to count something exactly, but instead to establish a *lower bound* for the number of something. In particular, it's often used to prove that there is *at least one* of something. Examples: (i) Among a group of n people, at least two have the same number of acquaintances. (ii) Hooking computers up to share servers; example 9 page 316. (iii) Every sequence of n^2+1 distinct real numbers contains a subsequence of length at least $n+1$ that is either increasing or decreasing.

• **One-to-one correspondences**. We've learned basic counting techniques, not merely how to solve a short (or long) list of particular problems. Often a new counting problem can be solved by placing the objects to be counted into one-to-one correspondence with some other class of objects which we've already learned how to count.

• **Synthesis**. Often a new counting problem can be attacked by splitting it into subproblems, where each subproblem is one we've learned to deal with. Then use the sum and product rules to combine the resulting sub-counts into a total count. Putting *indistinguishable objects into indistinguishable boxes* is an example; we did one homework problem of this type with particular numbers, by *ad hoc* reasoning. (We've learned no general formula for this problem.)

Not every general type of counting problem has a clean, simple formula, but the techniques we've learned are nonetheless often helpful in working out particular cases.

Last comment: We've talked a lot about various versions of "distributing objects into boxes". Many problems involve splitting some set into subsets with various restrictions on how this can be done, and can be rephrased as "objects into boxes" problems. One of our goals is to learn, through experience, to recognize when a counting problem can be viewed (by setting up a one-to-one correspondence) as a "objects into boxes" problem of one of the types we've studied.