MATH 22

Lecture M: 10/14/2003

MORE ABOUT FUNCTIONS

Form follows function.
—Louis Henri Sullivan

This frightful word, function, was born under other skies than those I have loved. —Le Corbusier

D’ora innanzi ogni cosa deve camminare alla perfezione.
—Benito Mussolini

Administrivia

- http://denenberg.com/LectureM.pdf

- Project 5 handed out today, due 10/21 in class.

- Exam 2: Monday 10/27, time & place TBA.

Today: Much more about functions.

From now on everything must function to perfection.

Review

A *binary relation* on set S is a subset of $S \times S$, that is, it's just a set of ordered pairs. Every ordered pair in the relation consists of two things that stand in that relation to each other. Examples: <, "is within 100 miles of".

Binary relations can be *reflexive*, *symmetric*, *transitive*, or any combination of these (or none of these).

A *binary relation from set S to set T* is a subset of $S \times T$. Example: "is a city in"; S is {cities} and T is {states}.

There are $2^{|S||T|}$ binary relations from S to T.

==

A *function* (informally) is a rule that takes a value (the "input", or *preimage*) from a set called the *domain*, and produces a new value (the "output", or *image*) from a set called the *codomain*. We write this $f : D \rightarrow C$.

If A is a subset of the domain of f, then the *image of A*, written $f(A)$, is the subset of the codomain consisting of those elements that are the image of some element of A. The image of the entire domain is called the *range* of f.

$\lfloor$Floor$\rfloor$, $\lceil$ceiling$\rceil$, and truncate are functions on numbers.

Review, cont.

A function that never collapses two inputs to the same output is called *injective*, or *one-to-one*. Formally, f is injective if $f(a) = f(b)$ implies $a = b$.

If $f : A \to B$ is injective, then $|A| \leq |B|$ (for finite sets)

A function that hits each value in the codomain is called *surjective*, or *onto*. Formally, f is surjective if for each $b \in B$ there is an $a \in A$ such that $f(a) = b$. A function is surjective if and only if its range equals its codomain. (Onto-ness is sensitive to what we choose as codomain.)

If $f : A \to B$ is surjective, then $|A| \geq |B|$ (for finite sets)

A function that is both injective and surjective is called *bijective*, or a *bijection*, or *one-one onto*, or a *one-to-one correspondence*. It assigns each element of its domain to a distinct element of its codomain (since it's one-one) and hits the entire codomain (since it's onto). It perfectly matches up the domain and codomain.

If $f : A \to B$ is bijective, then $|A| = |B|$ (for finite sets)

Two Minor Points

Theorem: Suppose $f : S \to T$ and $A, B \subseteq S$. Then

$$f(A \cup B) = f(A) \cup f(B)$$

$$f(A \cap B) \subseteq f(A) \cap f(B)$$

(Why this asymmetry? We'll go through the proof and see. Then we'll fix it by pointing out that the second expression becomes an equality if f is injective.)

Suppose $f : A \to B$ and $A_1 \subseteq A$. Then we can make a new function $f_1 : A_1 \to B$ in the obvious way: we define $f_1(a)$ to be $f(a)$! We call f_1 *the restriction of f to A_1*.

For example, think of the +1 function which takes real numbers to real numbers. The restriction of this function to the integers is +1 function that takes integers as inputs.

In the opposite direction, let A_2 be a set such that $A \subseteq A_2$ and let $f_2 : A_2 \to B$ be such that $f_2(a) = f(a)$ for every $a \in A$. Then f_2 is said to be *an extension of f to A_2*.
[As before: Why is f_1 *the* restriction of f to A_1, but f_2 is *an* extension of f to A_2?]

Functions (formally)

We’ve tried to get an intuitive grasp of functions. But how shall we define them formally? It turns out that a function is a special kind of binary relation.

A *function* f *from A to B*, written f : A □ B, is a binary relation from A to B with the following special property: for each a □ A there is *exactly one* b □ B such that (a,b) □ f.

That is, a function—like a binary relation—is a set of ordered pairs. But *any* set of ordered pairs is a binary relation; for a set of ordered pairs to be a function it must satisfy the special property above. We can think of it as two properties:

– For every x in the domain, there must be an ordered pair (x,y) in the function, and

– For every ordered pair (x,y) in the function, there can’t be any other ordered pair (x,z) in the function.

The first of these properties guarantees that f produces *some* output for every input. The second guarantees that f produces a *unique* output for every input. All functions must have this property: for every input there must be exactly one output.

Terminology Reprise

Now let's look at all the function terminology and see what it boils down to in the language of ordered pairs.

A function f : X □ Y, that is, a function with *domain* X and *codomain* Y, is a subset of X □ Y with the property that for each x □ X there is exactly one y □ Y such that (x,y) □ f. [We just said this.]

If (x,y) □ f, then we say that y is the *image* of x under f and x is a preimage of y under f, and we write f(x) = y.

If A □ X, then the *image* f(A) of A under f is the set of all y □ Y such that there is some (x,y) □ f with x □ A.

The *range* of f is the set of all y □ Y such that (x,y) □ f for some x □ X.

Function f is *injective* if for every y □ Y there is *at most* one pair (x,y) □ f.

Function f is *surjective* if for every y □ Y there is *at least* one pair (x,y) □ f.

Function f is *bijective* if for every y □ Y there is *exactly* one pair (x,y) □ f.

Multiple Arguments

So far we've considered only functions of a single argument (like +1, or squaring, or father-of). How do we handle functions of multiple arguments, like the function + which takes two numbers in and produces one out? [E.g., +(3,7) = 10, more commonly written 3+7=10.]

Answer: We force such functions to be functions of one argument by making them functions of ordered n-tuples! For example, we think of + as a function that takes a (single) ordered pair of (say) integers as input, and produces an integer as output. We write $+ : \underline{\boldsymbol{Z}} \square \underline{\boldsymbol{Z}} \square \underline{\boldsymbol{Z}}$. That is, it's not technically +(3,7), but +((3,7)). The same applies to functions of even more arguments.

DON'T CONFUSE THIS with the use of ordered pairs in the formal definition of a function, which is a distinct use of ordered pairs. As an example, let's look at + more closely. It's a subset of $(\underline{\boldsymbol{Z}} \square \underline{\boldsymbol{Z}}) \square \underline{\boldsymbol{Z}}$:

$$+ \;=\; \{ ((1,1), 2),\;\; ((3, 7),\; 10),\;\; ((4, 1),\; 5),\;\; . \;\; . \;\; . \}$$

The first □ here packages up two arguments into one, so that we can think of + as a function of one argument. (There could be more of these for a function of many arguments.) The second □ is the one connecting the domain with the codomain, that is, pairing the argument of the function with the value of the function.

Multiple Args, cont.

Example: Consider the function that takes a student ID, a course ID, and an exam ID into a grade. E.g.,

Gr(502-33-1234, Math 22, Test1) = 96

If S is the set of students, C the set of courses, and E the set of exams, it's Gr : $S \times C \times E \to \{0, 1, 2, . . ., 100\}$ and we should write Gr((502-33-1234, Math 22, Test1)).

A very important example: Suppose A and B are any two sets and $D \subseteq A \times B$. Then the function $\mathrm{pr}_1 : D \to A$ is defined as follows:

For any $a \in A$ and $b \in B$, $\mathrm{pr}_1(a, b) = a$

This function is called the *first projection of D*. Think of pr_1 as the function that ignores its second argument and outputs its first argument. Similarly, the *second projection of D* is the function pr_2 with $\mathrm{pr}_2(a,b) = b$.

Generalization: The r^{th} *projection* of any subset of $A_1 \times A_2 \times . . . \times A_n$ is the function that takes the ordered n-tuple $(a_1,a_2, . . ., a_n)$ into its r^{th} component a_r.

[Why the word "projection"? Think of a two-dimensional blob projected onto an axis. (Picture.)]

How Many Functions?

How many functions are there from S to T (finite sets!)?
Rather than counting sets of ordered pairs (as we did for relations), it's easier to think of creating a function as a sequence of choices. For each $s \in S$ we must pick some $t \in T$ to be the value of the function, and there are $|T|$ possibilities. So we pick a t for $s1$, then a t for $s2$, etc., with $|T|$ ways to choose each. And of course we make $|S|$ such choices in all. By Rule of Product, there are $|T|^{|S|}$ ways to build a function from S to T. (Indeed, we use T^S to denote *the set of all functions from S to T*.)

How many injective functions are there from S to T?
Almost the same problem. For the first element of S we have $|T|$ choices, for the second we have $|T|–1$ choices (because we can't reuse the first choice) for the third we have $|T|–2$ choices, etc. The answer is $|T|! \,/\, (|T|–|S|)!$

How many bijective functions are there from S to T?
The trick here is to see that, for a bijection, we must have $|S| = |T|$, and any injective function is bijective if $|S| = |T|$. So the answer is $|T|! = |S|!$

How many surjective functions are there from S to T?
Turns out we need a new tool for this: Stirling numbers.

Binary Operations

The domain of a multi-arg function can be the cross product of different sets, and the codomain can be yet another set, as we saw with Gr : $S \times C \times T \to \underline{\mathbf{Z}}$. When a function has two arguments both from the same set, that is f : $S \times S \to T$, we call f a *binary operation*.

When the codomain is also the same set, f : $S \times S \to S$, we call f a *closed binary operation*. (We'll have very little to do with binary operations that aren't closed.)

If f is a binary operation, we write x f y to mean f(x,y).

Example: + : $\underline{\mathbf{Z}} \times \underline{\mathbf{Z}} \to \underline{\mathbf{Z}}$ is a closed binary operation on the integers. We write $a+b$ instead of +(a,b). Similarly *, –, etc. But /, considered as a binary operation on the integers, is not closed, nor is – on the *positive* integers.

Example: □ : $S \times S \to S$ is a closed binary operation on sets. We write A □ B instead of □(A,B).

Example: □ : $P \times P \to P$ is a closed binary operation on propositions: it takes in two propositions and outputs a third. We write p □ q. Sometimes we consider □ as a binary operation on the set of truth values {t, f}.

Properties of Operations

Now let * be some arbitrary closed binary operation on S (not multiplication). We use the following terminology:

Operation * is *commutative* if $a*b = b*a$ for all a,b □ S.

Operation * is *associative* if $(a*b)*c = a*(b*c)$ for all a,b,c □ S.

(Comment: For associative operations we can write $a*b*c$, which doesn't make sense otherwise. We can also write $a*b*c*...*z$ because any way of putting in the parentheses gives the same answer. It turns out that associativity is more important than commutativity; many operations aren't commutative, but few non-associative operations are very interesting.)

Operation * is *idempotent* if $a*a = a$ for all a □ S. (For example, □ and □ are idempotent. Arithmetics +, –, and so forth aren't, but *max* and *min* are: $max(x, x) = x$.

(Actually, some of this terminology applies even to binary operations that aren't closed. Which of these terms *requires* the binary operation to be closed?)

A Critical Property

Let * be a binary operation on S. We say that $e \in S$ is an *identity* for * if for all $x \in S$ we have $x*e = e*x = x$.

We know many examples: The operation + on numbers has identity 0. Multiplication has identity 1. The empty set ∅ is the identity for the union operation on sets. (What is the identity for the intersection operation? For symmetric difference?) Regarded as operations on truth values, has identity false and ∧ has identity true.

On the other hand, consider the binary operations *min* and *max* on numbers. These operations are commutative, associative, and idempotent. But neither has an identity element.

Can a binary operation have more than one identity?

Theorem: No.

Proof: Suppose * is a binary operation on S with two identities e_1 and e_2. Since e_1 is an identity, $e_1 * e_2 = e_2$. But since e_2 is an identity, $e_1 * e_2 = e_1$. Since $e_1 * e_2$ can only have one value, we must have $e_1 = e_2$. So all identity elements are the same. (Question: Is this a proof by contradiction?)

Final Miscellany

A function from *S* into *S* is called a unary relation on *S*. For example, – is a unary operation on the integers (or the rationals, or the reals). Floor, ceiling, and set complement are other unary relations that we've seen.

Here's a very important example of a function: We know about sequences of numbers (or of anything) like

<3, 7, 9, 0, 1, 3, 5, 5, –3, 0>

Which is finite, and infinite sequences like

<0, 1, 1, 2, 3, 5, 8, 13, 21, 34, . . . >

(the so-called *Fibonacci sequence*). What are these, exactly? We can think of a sequence of objects from a set *X* as a function whose domain is some subset of the integers and whose codomain is *X*. For example, the Fibonacci sequence is really a function f : $\underline{\boldsymbol{Z}}^*$ □ $\underline{\boldsymbol{Z}}^*$ where f(0) = 1, f(1) = f(2) = 1, f(3) = 2, f(8) = 21, etc. So it's no mystery that sequences can have duplicates; it's the same as saying that not all functions are injective!

Grimaldi shows how to count the number of binary operations on *S*, the number of commutative binary operations on *S*, and the number of binary operations on *S* with given identity element. Read if interested.