Mathematics 69
Winter 2009
Some Mathematical Structures

So far we have learned the syntax of first order logic; that is, we know what the symbols are and how to combine them into formulas. We have also talked about semantics (meaning) in an informal way.

We are about to talk about semantics in a more formal way. That is, we will interpret sentences of first order logic as being *true* or *false* (or, more formally, *satisfied* or *not satisfied*) in a particular mathematical structure. In preparation, we will learn about a few mathematical structures that will be useful to us later.

Basically, a mathematical structure consists of a set of elements with various functions and relations defined on the set. For example, there are many mathematical structures that consist of a set with a binary relation defined on it.

Definition: An *equivalence relation* consists of a set X with a binary relation $\approx$ on X that is

1. *reflexive*: For every x in X, we have $(x \approx x)$,
2. *symmetric*: For every x and y in X, if $x \approx y$ then $y \approx x$,
3. *transitive*: For every x, y and z in X, if $x \approx y$ and $y \approx z$ then $x \approx z$.

Sometimes we say "$\approx$ is an equivalence relation on the set X."

For example, congruence modulo 3 is an equivalence relation on $\mathbb{Z}$, defined by

$$m \approx n \iff (n - m) \text{ is a multiple of } 3.$$

Exercise 1: Show that tautological equivalence is an equivalence relation on the set of wffs of sentential logic; that is, if we define

$$\alpha \approx \beta \iff \alpha \models\!\models \beta,$$

then $\approx$ is an equivalence relation on the set of wffs.

If $\approx$ is an equivalence relation on the set X, then for any $x \in X$ we can define the *equivalence class* of x, denoted $[x]$, by

$$[x] = \{y \mid y \approx x\};$$

that is, $[x]$ is the set of all elements of X that are equivalent to x. It is not hard to show, using the properties of equivalence relations, that any two equivalence classes are either identical or disjoint; specifically, for all x and y in X

$$x \approx y \implies [x] = [y]$$
$$x \not\approx y \implies [x] \cap [y] = \emptyset.$$

Another way to say this is that the equivalence classes form a *partition* of X.

Exercise 2: The first-order language of equivalence relations contains the equality symbol and a single binary predicate symbol $\approx$, which you may write in infix notation ($x \approx y$ rather than $\approx xy$). Suppose we interpret $\forall x$ to mean "for every x in the set X," and interpret $\approx$ to be an equivalence relation on X. Write a sentence of our language (feeling free to use all our abbreviations and so forth) that says every two equivalence classes are either identical or disjoint.

Note, we *cannot* say directly "for every equivalence class," because in our first-order language we cannot talk about subsets of X, only about elements of X. The textbook makes this point: We cannot say in the language of elementary number theory, "Every nonempty set of natural numbers has a least element." However, we can talk indirectly about some sets. From the textbook's example, again, we can say, "The set of primes has a least element," by saying, "There is a smallest prime."

It might help to note that two sets are the same iff they have exactly the same elements.

For example, if $\approx$ is the equivalence relation of congruence modulo 3 on $\mathbb{Z}$, then there are three equivalence classes (in this case often called *congruence classes*), and every integer belongs to exactly one of them:

$$[0] = \{n \mid n \text{ is a multiple of } 3\}$$

$$[1] = \{n \mid (n-1) \text{ is a multiple of } 3\}$$

$$[2] = \{n \mid (n-2) \text{ is a multiple of } 3\}.$$

The collection of equivalence classes is sometimes denoted by $X/\approx$, so

$$\mathbb{Z}/\approx \ = \{[0], [1], [2]\}.$$

A common and important mathematical construction, which we will need to prove the Completeness Theorem for first-order logic, is the following:

1. Start with some mathematical structure, that is, a set X with some functions and relations defined on it.

2. Define an equivalence relation $\approx$ on X.

3. Use the functions and relations defined on X to define functions and relations on $X/\approx$.

There is a natural way to do this, but it doesn't always work.

To see this, let's look at the example of congruence modulo 3 on $\mathbb{Z}$. One natural function on $\mathbb{Z}$ is addition. We use addition on the integers to define addition of congruence classes. That is, for all m and n, we define

$$[m] + [n] = [m + n].$$

It's not enough to make this definition; we need to know that the definition is reasonable, that is, that addition of congruence classes is *well-defined*. Before saying exactly what this means, let's look an example of a proposed function that is not well-defined. We will try to define exponentiation of congruence classes by, for all m and n,

$$[m]^{[n]} = [m^n].$$

This is not a reasonable definition; to see this, let's set $a = [2]$ and $b = [1]$. Then according to this definition, we must have

$$a^b = [2]^{[1]} = [2^1] = [2].$$

But we also have that $a = [2] = [-1]$ and $b = [1] = [4]$. So, according to our definition, we must have

$$a^b = [-1]^{[4]} = [(-1)^4] = [1].$$

That is, our definition tells us that $a^b = [2]$ and $a^b = [1]$. Since $[2] \neq [1]$, this is not a reasonable definition; this (proposed) function is **not** well-defined.

Definition: A proposed function f is *well-defined* if the definition gives a unique value for $f(x_1, \ldots, x_n)$ for all possible values of the arguments x_1, $\ldots$, x_n.

In our case, we have an equivalence relation $\approx$ on a set X, we have a function f on X, and we want to define a function F on equivalence classes by

$$F([x_1], \ldots, [x_n]) = [f(x_1, \ldots, x_n)].$$

To show that this is well-defined, we need to show that if we represent the same equivalence classes $[x_1]$, $\ldots [x_n]$ in a different way, we still get the same answer for $F([x_1], \ldots, [x_n])$ (possibly represented in a different way.) That is, we need to show that for all x_1, $\ldots$, x_n and y_1, $\ldots$, y_n in X,

If $[x_1] = [y_1]$, $\ldots$, $[x_n] = [y_n]$,
then $[f(x_1, \ldots, x_n)] = [f(y_1, \ldots, y_n)]$.

We can rephrase this as:

If $x_1 \approx y_1$, $\ldots$, $x_n \approx y_n$,
then$f(x_1, \ldots, x_n) \approx f(y_1, \ldots, y_n)$.

Our proposed exponentiation function on congruence classes was not well-defined because $2 \approx (-1)$ and $1 \approx 4$ but $2^1 \not\approx (-1)^4$.

Let us prove that addition of congruence classes is well-defined. We need to show that, for all integers x_1, y_1, x_2, y_2, we have

If $x_1 \approx y_1$ and $x_2 \approx y_2$
then $(x_1 + x_2) \approx (y_1 + y_2)$.

So let's suppose that $x_1 \approx y_1$ and $x_2 \approx y_2$. By the definition of $\approx$, this means that $(x_1 - y_1)$ and $(x_2 - y_2)$ are both multiples of 3:

$$(x_1 - y_1) = 3n \qquad\qquad (x_2 - y_2) = 3m.$$

To show $(x_1 + x_2) \approx (y_1 + y_2)$ we must show that $((x_1 + x_2) - (y_1 + y_2))$ is also a multiple of 3. This is easy to show:

$$(x_1 + x_2) - (y_1 + y_2) = (x_1 - y_1) + (x_2 - y_2) = 3n - 3m = 3(n - m).$$

This completes the proof.

Exercise 3: Let X be the set of all wffs of sentential logic and $\approx$ be tautological equivalence. Define a binary (2-place) function on equivalence classes, which we could call conjunction, by

$$[\alpha] \bigwedge [\beta] = [\alpha \wedge \beta].$$

Prove that this function is well-defined.

As you do this, at some point you are going to have to prove that two wffs are tautologically equivalent. For this exercise, please do this by showing *explicitly* that any truth assignment that satisfies one of the formulas also satisfies the other, and conversely.

You may think it's obvious that these wffs are tautologically equivalent. I agree, and after this proof, you can get away with saying so, or giving a more informal explanation, in similar circumstances.

The idea behind using a relation on X to define a relation on $X/\approx$ is quite similar to that behind using a function on X to define a function on $X/\approx$, and again we need to worry about whether our relation on equivalence classes is well-defined.

For example, let's look again at $\mathbb{Z}$ with the equivalence relation of congruence modulo 3. Less than, $<$, is a binary relation on $\mathbb{Z}$. We can try to define a binary relation on equivalence classes by

$$[n] < [m] \iff n < m.$$

This relation turns out not to be well-defined. To see this, let's again set $a = [2]$ and $b = [1]$. Then according to this definition, we must have

$$a < b \iff [2] < [1] \iff 2 < 1,$$

so that $a \not< b$. But we also have that $a = [2] = [-1]$ and $b = [1] = [4]$. So, according to our definition, we must have

$$a < b \iff [-1] < [4] \iff -1 < 4,$$

so that $a < b$. That is, our definition tells us that $a \not< b$ and $a < b$. This is not a reasonable definition; this (proposed) relation is **not** well-defined.

Exercise 4: Let X be the set of all wffs of sentential logic and $\approx$ be tautological equivalence. Define a binary (2-place) relation on equivalence classes by

$$[\alpha] <_{sub} [\beta] \iff \alpha \text{ is a subformula of } \beta.$$

Prove that this relation is not well-defined.

Exercise 5: Let X be the set of all wffs of sentential logic and $\approx$ be tautological equivalence. Define a binary (2-place) relation on equivalence classes by

$$[\alpha] \models [\beta] \iff \alpha \models \beta.$$

Determine whether this relation is well-defined and prove your answer is correct.

Exercise 6: In general, an n-place relation r on X can (sometimes) be used to define an n-place relation R on $X/\approx$ by

$$R([x_1], \ldots, [x_n]) \iff r(x_1, \ldots, x_n).$$

Give a criterion for determining whether or not this proposed relation is well-defined. Your criterion should be analogous to the second criterion given above for determining whether a function F on equivalence classes defined by

$$F([x_1], \ldots, [x_n]) = [f(x_1, \ldots, x_n)]$$

is well-defined. In particular, it should be expressed in terms of the equivalence relation $\approx$ on X rather than making explicit mention of equivalence classes.

We say the relation R on equivalence classes is *induced* by the relation r on X.

In the same way, a function f on X can *induce* a function F on equivalence classes, defined by $F([x_1], \ldots, [x_n]) = [f(x_1, \ldots, x_n)]$, provided of course that this function is well-defined.

Now it is time to make a connection with first-order logic. We will define $\mathfrak{L}$ to be the language of first-order logic that has equality (that is, it has the equality symbol $=$), a two-place predicate symbol P, and no other predicate, constant or function symbols. Given a mathematical structure that consists of a set X with a binary relation R on X, which we may denote

$$\mathfrak{A} = \langle X, R \rangle ,$$

we translate the parameters of $\mathfrak{L}$ as folllows: $\forall x$ means "for all x in X," and Pxy means "x and y are related by R." Then, informally, if σ is any sentence of $\mathfrak{L}$, we say that the structure $\mathfrak{A}$ *satisfies* σ, or more informally that σ is *true in* $\mathfrak{A}$, if and only if the translation of σ is true. We denote "$\mathfrak{A}$ satisfies σ" by

$$\models_{\mathfrak{A}} \sigma.$$

Later we will have a formal definition of this concept, by recursion on the complexity of the wff σ, that does not depend on our understanding of the English word "true." But this will do for now.

For example, suppose $\mathfrak{A} = \langle X, \approx \rangle$. This structure is an equvalence relation if and only if the following three sentences of $\mathfrak{L}$ are satisfied by $\mathfrak{A}$:

1. $\forall x \, Pxx$;
2. $\forall x \, \forall y \, (Pxy \to Pyx)$;
3. $\forall x \, \forall y \, \forall z \, ((Pxy \wedge Pyz) \to Pxz)$.

An *linear ordering* consists of a set X with a binary relation $\leq$ on X that is:

1. *reflexive*: For every x in X, we have $(x \leq x)$,
2. *antisymmetric*: For every x and y in X, if $x \leq y$ and $y \leq x$ then $x = y$,
3. *transitive*: For every x, y and z in X, if $x \leq y$ and $y \leq z$ then $x \leq z$,
4. *total*: For every x and y in X, we have $x \leq y$ or $x = y$ or $y \leq x$.

Sometimes we say "$\leq$ is a linear ordering of the set X."

If $\leq$ is reflexive, antisymmetric, and transitive, but not necessarily total, it is called a *partial ordering*. If $\leq$ is reflexive and transitive, but not necessarily antisymmetric or total, it is sometimes called a *preordering*.

Exercise 7: Write down four sentences of the language $\mathfrak{L}$ such that a structure $\mathfrak{A} = \langle X, \leq \rangle$ is a linear ordering if and only if it satisfies those four sentences.

Exercise 8: Give at least two examples each of a preordering that is not a partial ordering, a partial ordering that is not a linear ordering, and a linear ordering. One of your examples should be the relation of tautological implication on the set of wfffs of sentential logic. Another should be the induced relation of tautological implication on the set of equivalence classes of wffs (as in Exercise 5.)

Exercise 9: Suppose that X is a set and $\leq$ is a preordering of X. Define a new binary relation on X by

$$x \approx y \iff (x \leq y \;\&\; y \leq x).$$

Show that $\approx$ is an equivalence relation on X, that $\leq$ induces a well-defined relation on equivalence classes, and that this induced relation is a partial ordering of $X/\approx$.

A preview of things to come: Suppose Σ is the set of the four sentences that you wrote down for Exercise 7. Then we say that Σ is a set of *axioms* for the *theory* of linear orderings. As in the case of sentential logic, we will have a notion of *logical* (rather than tautological) *implication*: $\Sigma \models \sigma$ if and only if every structure that satisfies all the sentences in Σ also satisfies σ. In this example, that will mean that $\Sigma \models \sigma$ if and only if σ is true in every linear ordering.

Also as in sentential logic, we will have a notion of deduction. (Actually, it will be exactly the same notion of deduction, except that instead of taking the set of all tautologies as the set of "tautological axioms," we will have a set of "logical axioms.") We will say that $\Sigma \vdash \sigma$ if and only if there is a deduction of σ from Σ.

Also as in sentential logic, we will have the Soundness and Completeness Theorems: For any set of sentences Σ and sentence σ,

$$\Sigma \models \sigma \iff \Sigma \vdash \sigma.$$

(The Completeness Theorem for first-order logic is much harder to prove.) In our case, this means that a sentence σ of $\mathfrak{L}$ is true in every linear ordering if and only if it is deducible (in the formal sense) from the four axioms you wrote down. Or, to put it another way, given any sentence σ of $\mathfrak{L}$, exactly one of the following two possibilities holds:

1. The sentence σ can be deduced from the four axioms. That is, σ has a proof.

2. The sentence σ is false in some linear ordering. That is, σ has a counterexample.

Definition: A linear ordering is *without endpoints* if it has no largest element and no smallest element.

It is *dense* if between any two points there is another point.

It is *discrete* if, given any point that is not the largest element there is a next-largest point (an *immediate successor* in the ordering), and given any point that is not the smallest point there is a next-smallest point (an *immediate predecessor* in the ordering.)

For example, $\langle \mathbb{Z}, \leq \rangle$ is a discrete linear ordering without endpoints, and $\langle \mathbb{Q}, \leq \rangle$ is a dense linear ordering without endpoints.

Exercise 10: Write down a sentence σ of the language $\mathfrak{L}$, such that a linear ordering $\mathfrak{A} = \langle X, \leq \rangle$ satisfies σ if and only if:

(i.) The linear ordering $\mathfrak{A}$ is without endponits.

(ii.) The linear ordering $\mathfrak{A}$ is dense.

(iii.) The linear ordering $\mathfrak{A}$ is discrete.

Definition: A linear ordering $\mathfrak{A} = \langle X, \leq \rangle$ is called *countable* iff the set X is countable, that is, the elements of X can be indexed by the natural numbers, $X = \{x_0, x_1, x_2, \ldots, x_n, \ldots\}$. (The same goes for an equivalence relation $\langle X, \approx \rangle$ or any other structure $\langle X, \ldots \rangle$.)

If $\mathfrak{A} = \langle X, \leq_X \rangle$ and $\mathfrak{B} = \langle Y, \leq_Y \rangle$ are two linear orderings, a function $f : A \to B$ is an *isomorphism* between $\mathfrak{A}$ and $\mathfrak{B}$ if and only if:

1. The function f is a bijection, that is, f is one-to-one and onto
2. For all x and w in X, we have that

$$x \leq_X w \iff f(x) \leq_Y f(w).$$

This definition may remind you of the definition of an isomorphism of vector spaces, or any other sort of isomorphism you have seen defined. Basically, f is an isomorphism between two structures if f maps one set bijectively onto the other and f *preserves* the structure (the functions, relations, and so forth.)

Exercise 11: Define the notion of isomorphism between two equivalence relations $\mathfrak{A} = \langle X, \approx_X \rangle$ and $\mathfrak{B} = \langle Y, \approx_Y \rangle$.

Later we will give a general definition of isomorphism that applies to all kinds of mathematical structures.

Example: An abelian group consists of a set G and a binary operation $+$ satisfying the following axioms:

1. The operation $+$ is associative.
2. The operation $+$ is commutative.
3. There is an element $0 \in G$ that is an *identity element* for the operation $+$.
4. For each element $x \in G$ there is another element $-x \in G$ that is an *inverse* of x for the operation $+$.

If you have never seen the definition of a group before, you may recognize the first vector space axioms. If you take a vector space and forget about multiplication by scalars, just considering vector addition, you have an abelian group.

Occasionally textbooks list an additional axiom, namely

Axiom 0. The set G is closed under the operation $+$ (whenever x and y are in G, then $x + y$ is in G as well.)

We do not list this as an axiom, because by "a binary operation on G" we mean a function from pairs of elements of G (that is, from $G \times G$) to G. In other words, closure is part of our definition of an operation on G.

Now for a connection to first order logic: Consider a first-order language with equality, a constant symbol 0, a one-place function symbol $-$, and a two-place function symbol $+$. If G is an abelian group, we can translate our symbols as follows:

Translate $\forall x$ a "for every $x \in G$," translate $+$ as the binary operation, translate 0 as the identity element, translate $-$ as the function taking x to its inverse. Then the four axioms can be phrased as sentences of first order logic:

1. $\forall x \, \forall y \; = + x + yz + + xyz$.

Using the more usual infix notation for $-$ and $+$, with parentheses for ease of understanding and for unique readability, this becomes

$\forall x \, \forall y \, ((x + (y + z)) = ((x + y) + z))$.

We will use these conventions from now on.

2. $\forall x \forall y (x + y = y + x)$.
3. $\forall x (x + 0 = x)$.
4. $\forall x (x + (-x) = 0)$.

Example: You all know what a vector space is. But what is the appropriate first order language for talking about vector spaces over the real numbers?

One option is to do the following: Start with the symbols of the language on the previous page: a constant symbol 0, a one-place function symbol $-$, and a two-place function symbol $+$. Then add infinitely many one-place function symbols, a symbol f_r for every real number r. Our intended translation here is that $\forall x$ means "for every vector x," 0, $+$, $-$ denote the zero vector, vector addition, and additive inverse, and $f_r x$ denotes rx (where r is a real number and x is a vector.)

Those who know about cardinality will realize that I have just described a language that is not merely infinite, but *uncountable*. We will prove some theorems only for countable languages. For example, our proof of the Compactness Theorem for sentential logic relied heavily on the countability of the language. The textbook will often comment on whether the theorem holds for uncountable languages as well.

Now, the axioms for a vector space include the four axioms for an abelian group and four more axioms dealing with multiplying vectors by scalars. One of these four is fairly easy to state in our language: the one that says that if you multiply any vector v by the scalar 1, the product is v:

5. $\forall x\, f_1 x = x$.

The other axioms require more thought. For example, there is an axiom that says multiplication by scalars distributes over vector addition. It is usually phrased something like this: "For every real number r and all vectors x and y, we have that $r(x + y) = rx + ry$." This requires thought because while, with our intended translation, we can say "for every vector x" in our language ($\forall x$), we cannot say "for every real number r." What we need here is *infinitely many* axioms, one for every real number r.

6. For every real number r, we have the axiom
$\forall x\, \forall y\, (f_r(x + y) = f_r x + f_r y)$.

The remaining axioms are treated similarly.

7. For every pair of real numbers r and s, we have the axiom
$\forall x\, (f_r x + f_s x = f_{r+s} x)$.

8. For every pair of real numbers r and s, we have the axiom
$\forall x\, (f_r(f_s x) = f_{rs} x)$.