Applied Logic	Lecture 19: Expressing Mathematical Concepts (I)
CS 4860 Spring 2009	Tuesday, March 31, 2009

In the lectures so far we have studied the mathematics of propositional and first-order logic itself. We have described syntax, semantics and several calculi for constructing logical proofs. We have investigated metamathematical properties such as correctness, completeness, decidability, and compactness, but we haven't looked into "applied" logic yet.

We will now show how to make use of the formal apparatus in a rigorous account of mathematics. After all, this was the main purpose for developing formal logics: they should provide an "independent" mechanism for checking mathematical arguments and help removing ambiguities in the formulation of mathematical theories.

A mathematical theory usually proceeds by introducing new mathematical concepts, giving axioms that uniquely specify the properties of these concepts, and then proving new insights in the form of theorems that can be derived from the axioms. To make sure that the theory doesn't lead to nonsensical results one also has to prove that the axioms are consistent. Mathematicians with a more constructive mindset also prove that the new concepts can be simulated by already existing ones and that in this simulation the axioms actually become theorems. The account of real numbers, for instance, can be given in a completely axiomatic way but one could also describe real numbers as Cauchy sequences of rationals, which in turn are described as pairs of integers.

First-order logic is well suited to make the mathematical method more precise. Mathematical concepts are denoted by predicate symbols and axioms are represented by sets of formulas that will be added as assumptions whenever a statement about the properties of these concepts has to be proven. As a shorthand notation we will sometimes write $\mathcal{L}(ops; axioms)$ to denote the formal mathematical theory that is based on the operations ops and the axioms $axioms$. The completeness theorem of first-order logics tells us that a formula can be derived in finitely many steps if it is a valid theorem of the theory. However, compactness tells us that the set of axioms needs to be denumerable. Anything that cannot be formulated that way is out of the reach of first-order logic.

Using first-order logic as foundation for expressing mathematics is a conservative approach: the language of first-order logic will remain unchanged while we designate a few predicates as special and provide axioms for these predicates. The obvious advantage of doing so is that we can rely on the results that we have accomplished so far: we can use first-order tableaux or refinement logic to prove theorems about equality, functions, and later algebraic structures, integers, small algorithms, etc. and know that these proofs are correct and that every true statement of the theory can actually be proven that way. We don't have to extend the proof calculus anymore (which would require us to prove correctness and completeness again) – we simply add axioms.

However, there is also a drawback to this approach. With an increasing number of axioms, formal reasoning becomes more and more complex, as the size of the formula to be fed to the proof system grows with the number of axioms provided, even if not all of them are actually used in the proof. Many simple properties become astonishingly difficult to prove, as one has to isolate and instantiate the relevant axiom first before they can be used as "reasoning rule". Furthermore, mathematical theories are usually based on a "carrier set" (or type) S on which the operations and axioms are defined. While first-order logic can emulate a simple type structure it becomes increasingly difficult to reason about operations on types that related (like $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, and $\mathbb{R}$) or constructed from other types.

At the end of this course we will therefore explain how to make many of the designated predicates primitives of our formal language and convert axioms into corresponding proof rules. This will make the proof process much simpler and more elegant, but requires us to go over issues such as correctness, completeness, and compactness again.

For now, however, we will proceed by expressing mathematical concepts within the language of first order logic.

19.1 Equality

Equality is probably the most fundamental concept in mathematics. It allows us to treat two objects x and y as undistinguishable: every property of x holds in the same way for y as well. In mathematics equality is usually described as a binary predicate `E(_,_)` that comes with three axioms:

```
ref:    (∀x) E(x,x)
sym:    (∀x,y) (E(x,y) ⊃ E(y,x))
trans:  (∀x,y,z) ((E(x,y) ∧ E(y,z)) ⊃ E(x,z))
```

These axioms describe the usual reflexivity, symmetry, and transitivity laws of equality. But they are not sufficient to characterize what equality is really about.

Q: *Why are these axioms insufficient?*

To see that the three axioms do not uniquely specify equality we have to take a closer look at the possible semantical models of these axioms. A *model* for a set of axioms (see Smullyan page 49) is a universe U (sometimes also called a *domain* D) and an interpretation I of all the predicate symbols and parameters within that universe such that all the axioms evaluate to true. If the set of axioms is finite we write a model for a theory $\mathcal{L}(P_1,..,P_n;\, ax_1,..,ax_k)$ as $\langle D,\ R^{P_1},..,R^{P_n},\ c^{a_1},..,c^{a_m}\rangle$, where R^{P_i} is the relation that interprets the predicate P_i and c^{a_j} is the element of `D` that interprets the parameter a_j occurring in the axioms.

Obviously there are many models for the equality predicate, not all of them being *standard models*, i.e. the models that one would conventionally have in mind. For instance, when we talk about the integers, $\langle \mathbb{Z}, x{=}y\rangle$ is the model of equality we have in mind, but $\langle \mathbb{Z}, x{=}y \bmod 2\rangle$ is also a model. In fact, the above three axioms are only sufficient to characterize an *equivalence relation*. The law `(∀x,y) (E(x,y) ⊃ P(x) ⊃ P(y))` cannot be proven from the axioms.

The reason for this is that this formula is not true in every model of the three axioms `ref`, `sym`, and `trans`. For instance, if `P(x)` is interpreted as "x is a positive integer", then `P(1)` would be true in $\langle \mathbb{Z}, x{=}y \bmod 2\rangle$ but `P(-1)` would be false although $1{=}{-1} \bmod 2$. Thus if we want to characterize equality instead of a simple equivalence relation, we need to add more axioms that exclude all the models that do not describe an equality.

Q: *What axioms are missing?*

The best way to describe an equality is to characterize undistinguishability: if `x` is equal to `y` then we can replace `x` by `y` in every formula without changing its meaning. This insight is expressed by the so-called *substitution axiom*.

```
subst:  (∀x,y) (E(x,y) ⊃ P(..,x,..)  ⊃ P(..,y,..))
```

Actually, this axiom is not a pure axiom but an *axiom scheme* that needs to be instantiated for every predicate symbol that occurs in a formal theory and for every argument position in that predicate. Thus a first-order theory usually has to include a huge (but finite) number of substitution axioms.

On the other hand, the substitution axiom (scheme) is extremely powerful, as it simplifies many arguments that are quite difficult to expess with just reflexivity, symmetry, and transitivity. In fact, both the symmetry and the transitivity axiom can now be derived using reflexivity and substitution. For instance, the symmetry law can be proven with he following instance of the substitution axiom:

```
(∀x) E(x,x), (∀x,y,z) (E(x,y) ⊃ E(x,z) ⊃ E(y,z)) ⊢ (∀x,y) (E(x,y) ⊃ E(y,x))
                                                                    allR, allR, impR
 (∀x) E(x,x), (∀x,y,z) (E(x,y) ⊃ E(x,z) ⊃ E(y,z)) E(x,y) ⊢ E(y,x)
                                                    allL x, allL y, allL x
  (∀x) E(x,x), E(x,y) ⊃ E(x,x) ⊃ E(y,x) E(x,y) ⊢ E(y,x)                impL
1.  (∀x) E(x,x), E(x,y) ⊢ E(x,y)                                     axiom 2
2.  (∀x) E(x,x), E(x,x) ⊃ E(y,x), E(x,y) ⊢ E(y,x)                       impL
2.1.  (∀x) E(x,x), E(x,y) ⊢ E(x,x)                                    allL x
      E(x,x), E(x,y) ⊢ E(x,x)                                        axiom 1
2.2.  (∀x) E(x,x), E(y,x), E(x,y) ⊢ E(y,x)                           axiom 2
```

Using substitution the transitivity law can even be derived without using the reflexivity axioms. We leave that as an exercise to the reader.

An important derived concept is the unique-existence operator, which makes it possible to express that a mathematical object can be uniquely specified by a given property. We define

$$(\exists!x)P(x) \equiv (\exists x)(P(x) \wedge (\forall y)(P(y) \supset E(x,y)))$$

where P stands for an arbitrary unary predicate. For n-ary predicates this operator can be defined accordingly. We will need this operator in many of the subsequent formalizations.

19.2 Functions

Although functions are a part of the term language in many accounts of first-order logic, they are not considered fundamental in a rigorous approach to mathematics. Instead they are defined by their *graph*, i.e. the predicate that describes the input-output behavior of the function. Thus formally, n-ary Functions are described by (n+1)-ary predicates. A unary function f, for instance is described by a predicate R_f, where $R_f(x,y)$ is supposed to express that $f(x)=y$. To ensure that the predicate does in fact represent a function we need to state two axioms.

```
functionality:          (∀x)(∃!y) R_f(x,y)
functional equality:    (∀x,x',y,y')((E(x,x') ∧ R_f(x,y) ∧ R_f(x',y')) ⊃ E(y,y'))
```

The functionality axiom guarantees that R_f specifies a function and not just an arbitrary relation: for every "input" x there must be an output "y" and that output must be unique. The axiom of functional equality states that equal inputs must lead to the same output. While this axiom may appear trivial (and can, in fact, be derived from substitution and functionality), it becomes important in the formalization of residue classes like $\mathbb{Z}_3$ or $\mathbb{Q}$, where one has to show that the definition of a concrete function does not depend on the representative, e.g. that $\frac{3}{6}+\frac{2}{6}$ and $\frac{2}{4}+\frac{1}{3}$ will give the same result.

Q: *How do we prove functional equality?*

Both axioms are again *axiom schemes*: they have to be stated for every function symbol to be introduced. For n-ary functions, we have to state them for the appropriate (n+1)-ary predicate accordingly.

For most function symbols we may want to give additional axioms characterizing their specific properties. For instance, we will deal quite often with *binary operators*, usually written in infix format $x \circ y$. For some of these operators, we may want to require additional properties such as commutativity or associativity.

comm: $(\forall x,y,z)\ (R_\circ(x,y,z) \supset R_\circ(y,x,z))$
assoc: $(\forall x,y,z,s,t,w)\ (R_\circ(x,y,s) \supset R_\circ(s,z,w) \supset R_\circ(y,z,t) \supset R_\circ(x,t,w))$

Note that the commutativity axiom would usually be written as

$(\forall x,y,z,z)\ (R_\circ(x,y,z) \supset R_\circ(y,x,z') \supset E(z,z'))$

because of `functionality`, however, that is equivalent to the shorter form given above.

Further axioms depend on what else we can state about the domain. We will revisit this issue once we have introduced axioms that describe, specific domains such as the integers or reals.

Using (n+1)-ary predicates instead of the conventional function notation makes writing formulas a bit awkward. From now on we will therefore write $f(x)=y$ instead of $R_f(x,y)$ and even use infix notation, where possible. It should be understood, however, that this is just a *notational abbreviation* and that we cannot use $f(x)=y$ like an ordinary equality. If we were to describe injectivity or surjectivity of a function f, for instance, then the usual formulation of the axioms

inj-f: $(\forall x,y)\ (f(x)=f(y) \supset x=y)$
surj-f: $(\forall y)(\exists x)(f(x)=y)$

is actually just an abbreviation for the following formulas

inj-f: $(\forall x,y,z)(R_f(x,z) \wedge R_f(y,z) \supset E(x,y))$
surj-f: $(\forall y)(\exists x)(R_f(x,y))$

19.3 Defining Constants

Constants are best described by their effect on operators. The integer 0, for instance is known to be the neutral element of addition and the neutralizing one of multiplication. After introducing the axioms for + and *, one could therefore characterize 0 by the axiom

zero: $(\forall x)(x+0=x \wedge x*0=0)$

Alternatively, if one wants to avoid designating and axiomatizing parameters, one may formulate an axiom stating the existence of a unique element with the desired properties.

zero-exists: $(\exists!\mathtt{zero})(\forall x)(x+\mathtt{zero}=x \wedge x*\mathtt{zero}=\mathtt{zero})$

19.4 Ordering Relations

An ordering relation is a binary predicate $LE(\_,\_)$ that is very similar to an equivalence relation except that symmetry is replaced by antisymmetry.

le-ref: $(\forall x)\ LE(x,x)$
antisym: $(\forall x,y)\ ((LE(x,y) \wedge LE(y,x)) \supset E(x,y))$
le-trans: $(\forall x,y,z)\ ((LE(x,y) \wedge LE(y,z)) \supset LE(x,z))$

Both ordering relations and equalities are conventionally written in infix notation with predefined predicate symbols. We will adopt this convention from now on and write $x \leq y$ instead of $LE(x,y)$

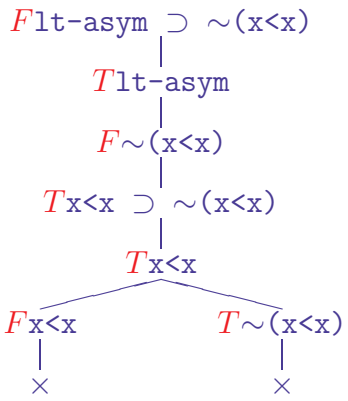
and x=y instead of E(x,y). One has to keep in mind, that these are only notational abbreviations for the "real" formulas, since otherwise we would have to redefine the language of first-order logic. With these notational changes the three axioms receive a more familiar form.

```
le-ref:     (∀x) x≤x
antisym:    (∀x,y) ((x≤y ∧ y≤x) ⊃ x=y)
le-trans:   (∀x,y,z) ((x≤y ∧ y≤z) ⊃ x≤z)
```

Note that ordering relations require the existence of an equivalence predicate. It is, however, possible to axiomatize *strict orders* without referring to an equality. A strict order is a binary (infix) predicate < that satisfies the following axioms

```
lt-asym:    (∀x,y) ((x<y ⊃ ∼(y<x))
lt-trans:   (∀x,y,z) ((x<y ∧ y<z) ⊃ x<z)
```

These two axioms also imply the irreflexivity of strict orders. The formula (∀x) ∼(x<x) can be derived by instantiating the axiom of antisymmetry, as the tableau proof to the right shows.



Typical models for a strict order are the conventional less-than relations on natural numbers, integers, rationals, or reals. One could, however, also define a strict order on booleans, with false being less than true. Orders also don't have to be linear. The (partial) order of nodes in a tree, for instance, satisfies the axioms for orders as well.

Strict orders can be derived from standard orders and vice versa if one has equality. One could define x≤y ≡ x<y ∨ x=y or x<y ≡ x≤y ∧ ∼(x=y), depending on which of the two predicates is axiomatized. It is easy to derive the corresponding axioms from the respective other ones and the equality axioms.