Lecture 14: First-Order Logic

1 Need For More Than Propositional Logic

In normal speaking we could use logic to say something like: If all humans are mortal (α) and all Greeks are human (β) then all Greeks are mortal (γ). So if we have $\alpha = (\mathrm{H} \rightarrow \mathrm{M})$ and $\beta = (\mathrm{G} \rightarrow \mathrm{H})$ and $\gamma = (\mathrm{G} \rightarrow \mathrm{M})$ (where H, M, G are propositions) then we could say that $((\alpha \wedge \beta) \rightarrow \gamma)$ is valid. That is called a *Syllogism.*

This is good because it allows us to focus on a true or false statement about one person. This kind of logic (propositional logic) was adopted by Boole (Boolean Logic) and was the focus of our lectures so far.

But what about using "some" instead of "all" in β in the statements above? Would the statement still hold? (After all, Zeus had many children that were half gods and Greeks, so they may not necessarily be mortal.) We would then need to say that "Some Greeks are human" implies "Some Greeks are mortal". Unfortunately, there is really no way to express this in propositional logic.

This points out that we really have been abstracting out the concept of *all* in propositional logic. It has been assumed in the propositions. If we want to now talk about *some*, we must make this concept explicit.

Also, we should be able to reason that if Socrates is Greek then Socrates is mortal. Also, we should be able to reason that Plato's teacher, who is Greek, is mortal.

First-Order Logic allows us to do this by writing formulas like the following. Here we have a generic x which we can talk about, and $\forall$ means "for all" and $\exists$ means "there exists". So let $H(x)$ mean x is human, $M(x)$ mean x is mortal, $G(x)$ mean x is Greek. Now we can finally say what we mean:

$$(\forall x)(H(x) \rightarrow M(x)), (\exists x)(G(x) \wedge H(x)) \models (\exists x)M(x))$$

and

$$(\forall x)(H(x) \rightarrow M(x)), (G(Teacher(Plato)) \wedge H(Teacher(Plato))) \models M(teacher(Plato)))$$

2 Predicate Calculus and First-Order Logic

Frege and Pierce noted the need for more than propositional logic, and they started developing *First-Order Logic* (or *Predicate Calculus*).

First-Order Logic speaks about *objects*, which are the domain of discourse or the universe. First-Order Logic is also concerned about *Properties* of these objects (called *Predicates*), and the *Names* of these objects.

Also we have *Functions* of objects and *Relations* over objects. (For example, Socrates's Father is a function of Socrates, while Socrates's son(s) is a relation involving the object Socrates). (Properties would be then mapped to relations on objects).

In addition we have added the *quantifiers* $\exists, \forall$. We also retain the constructs for combining formulas from propositional logic, namely $\wedge, \vee, \neg, \rightarrow, \leftrightarrow$

In summary, the elements of First-Order logic are:

1. propositional logic
2. variables
3. predicates - functions that always return true or false
4. functions

An interesting observation here is that these items give you exactly what you need to talk about mathematics!

3 Syntax of First-Order Logic

Using functions and relations and variables, we can now define the syntax of formulas in First-Order Logic.

Terms are built from

1. *Vars*: A set $\{x_1, \ldots\}$ of individual variables, and
2. *Function symbols*: A set $\{f_1, ldots\}$ of function symbols of arity ≥ 0. We use the notation $f^{(k)}$ to denote a k-ary function. 0-ary function symbols are constants symbols.

Terms of First-Order Logic formulas are defined recursively as follows.

1. Variables are terms.
2. If $t_1, t_2, ..., t_k$ are terms and f^k is a k-ary function symbol then $f^k(t_1, t_2, ..., t_k)$ is a term.

Note that if c is a 0-ary function symbol, then c is a term.

We can now define formulas using formulas and predicate symbols. *Formulas* of First-Order Logic are defined by:

1. If $t_1, t_2, ..., t_k \in Terms$ and P^k is a k-ary predicate symbol then $P^k(t_1, t_2, ..., t_k)$ is an atomic formula. (Note that if P is a 0-ary predicate symbol, then P is an atomic formula.)
2. If θ, ψ are formulas then $(\neg\theta)$, $(\theta \wedge \psi)$, $\ldots$ are formulas.

3. If θ is a formula and $x \in Vars$ then $(\exists x)\ \theta$ and $(\forall x)\ \theta$ are formulas.

So now we can write:
$((\forall x\ (\ \mathrm{H}(x) \to \mathrm{M}(x)) \wedge (\exists x)(\mathrm{G}(x) \wedge \mathrm{H}(x))) \to (\exists x)(\mathrm{G}(x) \wedge \mathrm{M}(x))$
to mean that if some humans are mortal and some Greeks are human then some Greeks are mortal.

Note: There is a special predicate symbol in first-order logic, which is the identity relation. It is a binary relation and is denoted by $\approx$.

Note: Our definition of $Terms$ above just tells us how to combine terms. What we really want to say is that $Terms$ is the smallest set that contains $Vars$ and is closed under these operations.

4 Hierarchy of Logics

It is interesting to observe the following hierarchy of logics and logic formulas. First we have first-order logic which is concerned with objects, while for second-order logic the elementary elements are functions and relations (i.e., sets of objects), while (finally) in third-order logic the main objects are sets of sets of objects. In First Order logic quantifiers ($\forall$, $\exists$) quantify over elements (objects), while in Second Order logic quantifiers are over relations and functions. (Note that there is a controversial point regarding this, because you could simply decide to make relations and functions be your objects and then Second Order logic would be First Order logic!). The entire system taken together is called *type theory*.

5 Mathematical Structures

First-order logic is interpreted over *mathematical structures*.

A mathematical structure is a tuple: $A = (D, P_1^A, ..., P_i^A, ..., f_1^A, ..., f_i^A, ...)$, where D is the domain of objects, and the P_i^A are relations, these are the interpretations of the preducate symbols, and the f_i^A are functions, these are the interpretation of the function symbols. Thus, if P_i is a k-ary relation name, then $P_i^A \subseteq D^k$, and, if f_i is a k-ary function name, then $f_i^A : D^k \to D$ is a k-ary function.

Note: If f_i is a 0-ary function then $f_i^A : D^0 \to D$, so f_i is a constant function. If P_i is a 0-ary relation, then $P_i^A \subseteq D^0$, so P_i^A is either $\emptyset$ (false) or $\{\langle\rangle\}$ (true). This boundary case correspond to propositional logic, which we discussed earlier.