COMP 409, Lecture 16

Tues., March 19$^{\text{th}}$, 2002

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 $\alpha = (\mathrm{H} \to \mathrm{M})$ and $\beta = (\mathrm{G} \to \mathrm{H})$ and $\gamma = (\mathrm{G} \to \mathrm{M})$ (where H, M, G are propositions) then we could say that $((\alpha \wedge \beta) \to \gamma)$ is always valid. That is called *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).

But what about using "some" instead of "all" in one of α or β in the statements above ? We could see that propositional logic is not enough to handle that case.

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' father is a function of Socrates, while Socrates' son(s) is a relation about the object Socrates). (Properties would be then mapped to relations on objects).

3 Syntax of First-Order Logic

Using functions and relations, and using the notation x_i to denote a *variable* (of type Object) to name individuals (so that we have a set of $Vars =$

$\{x_1, x_2, ..., x_n\}$) we could now define the syntax of formulas in First-Order Logic.

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

1. $Vars \subseteq Terms$
2. If $t_1, t_2, ..., t_k \in Terms$ and f^k is a k-ary function name
   then $f^k(t_1, t_2, ..., t_k) \in Terms$

and so *Formulas* of First-Order Logic could be defined as:

1. If $t_1, t_2, ..., t_k \in Terms$ and P^k is a k-ary relation name
   then $P^k(t_1, t_2, ..., t_k)$ is an atomic formula
2. If θ, ψ are formulas then $(\neg\theta)$, $(\theta \wedge \psi)$ 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 something like:

$((\forall x\ (\ \mathrm{H}(x) \to \mathrm{M}(x)) \wedge (\exists x)(\mathrm{G}(x) \wedge \mathrm{H}(x))) \to (\exists x)(\mathrm{G}(x) \wedge \mathrm{M}(x))$

to mean that if some humans are mortal and some Greeks are human then some Greeks are mortal.

Note There is a special relation in predicate calculus which is the identity relation, which is a binary relation and is denoted by $\approx$)

4 Hierarchy of Logic

It is interesting to observe the following hierarchy of logic and logic formulas. First we have first-order logic which is concerned about 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 1^{st} Order logic quantifiers ($\forall$, $\exists$) quantify over elements (objects), while in 2^{nd} 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 2^{nd} Order logic would be 1^{st} Order logic!).

5 Relational Structure of Predicate Calculus

Let $A = (D, P_1^A, ..., P_i^A, ..., f_1^A, ..., f_i^A, ...)$

where D is the domain of objects, and the P_i^A and f_i^A are called the interpretation I.

If P_i is a k-ary relation name, then $P_i^A \subseteq D^k$.

If f_i is a k-ary function name then $f_i^A : D^k \to D$.

If f_i is a 0-ary function then $f_i^0 : D^0 \to D$ (f_i is a constant function), and same for the boundary case for relations. These boundary cases correspond to propositional logic we had before.

6 Semantics of First-Order Logic

To define semanrtics of 1^{st} Order logic we define:

$\alpha : Vars \to D$ (a variable assignment)

$\overline{\alpha}$: Terms $\to D$ such that:

1. $\overline{\alpha}(x) = \alpha(x)$
2. $\overline{\alpha}(f(t_1, ..., t_k)) = f^A(\overline{\alpha}(t_1),\ ...,\ \overline{\alpha}(t_1))$ (recursive evaluation).

and so for a syntax structure A, we write $A, \alpha \models \varphi$ (or $A \models \varphi[\alpha]$) and we define it as follows:

1. $A, \alpha \models P(t_1, ..., t_k)$ if $(\overline{\alpha}(t_1), ..., \overline{\alpha}(t_1)) \in P^A$

2. $A, \alpha \models \neg\varphi$ if $A, \alpha \not\models \varphi$
$A, \alpha \models (\theta \wedge \varphi)$ if $A, \alpha \models \theta$ and $A, \alpha \models \varphi$

3. $A, \alpha \models (\exists x)\varphi$ if $A, \alpha[x \to a] \models \varphi$ for some $a \in D$
$A, \alpha \models (\forall x)\varphi$ if $A, \alpha[x \to a] \models \varphi$ for all $a \in D$