15414/614 Optional Lecture 3: Predicate Logic

Anvesh Komuravelli

1 Why Predicate Logic?

Consider the following statements.

1. *Every student is younger than some instructor.*
2. *Not all birds can fly.*

Propositional logic cannot capture the detailed semantics of these sentences. For the first sentence, propositional logic might help us encode it with a single proposition but perhaps, not more than that. For the second sentence, one might have a proposition standing for *every bird can fly* and say that the sentence is the negation of this proposition. But there are hidden details (that *every* bird can fly) within the proposition.
To encode a richer class of statements, predicate logic introduces the following additional constructs (to what we already have in propositional logic).

Variables : To stand for arbitrary entities.

Constants : To stand for specific entities.

Predicates : To denote relations between entities.

Functions : To map entities to entities.

Quantifiers : To talk about universal (for all, every, etc., denoted $\forall$) and existential (exists, some, etc., denoted $\exists$) events.

This kind of logic is called *first-order* logic. There are richer logics (n^{th}-order logics or higher-order logics) where, for e.g., predicate variables are also considered which can be quantified but we will not consider those in this treatment.
We are also interested in the special predicate, *equality*, denoted =. This is the *extensional equality* we saw in Coq, i.e. equality in terms of computational results and not just by defintion.

2 Examples

2.1 Predicates

Consider encoding the statement : *Every child is younger than its brother.* This is not true in general (consider the case when the brother in question is younger than the child). But our concern now is to encode it, not to decide whether it is true or false. Consider the following predicates.

$C(x)$: x is a child.

$B(x, y)$: x is a brother of y.

$Y(x, y)$: x is younger than y.

It is not hard to see that there are multiple sources of ambiguity in the statement.

1. How many brothers does a child have? We will assume that the statement is about *every* brother, for simplicity.
2. What does *its* mean? Does it imply the *existence* of at least one brother? Not clear.

The encoding depends on how we answer the second question above. If the answer is no, we might encode as follows.

$$\forall x, y.\, (C(x) \wedge B(y, x) \rightarrow Y(x, y))$$

If the answer is yes, however, we might encode as follows.

$$\forall x.\, (C(x) \rightarrow \exists y.\, (B(y, x) \wedge Y(x, y)))$$

2.2 Constants

If we want to encode the statement *Andy is younger than Paul*, we might use the above predicates to get something like

$$Y(A, P)$$

where A and P are *constants* standing for specific entities, Andy and Paul, respectively.

2.3 Functions

If we replace *brother* with *mother* in the previous example, we can simply replace the predicate $B(\cdot, \cdot)$ with $M(\cdot, \cdot)$ standing for *x is mother of y*. But, as mothers are unique, a more precise encoding is the following.

$$\forall x.\, (C(x) \rightarrow (\exists y.\, (M(y, x) \wedge Y(x, y)) \wedge \forall y_1, y_2.\, (M(y_1, x) \wedge M(y_2, x) \rightarrow y_1 = y_2)))$$

The newly added conjunct says that if y_1 and y_2 are both mothers of x, then $y_1 = y_2$.
We can simplify the notation by introducing a function $m(x)$ for the mother of x. The encoding will now be simply

$$\forall x. (C(x) \to Y(x, m(x))) .$$

3 Natural Deduction

In addition to the rules we saw for Propositional Logic, we have the following ones.

3.1 Equality

$$\frac{}{t = t} \text{ EQ-I}$$

$$\frac{t_1 = t_2 \qquad \phi[t_1/x] \qquad t_1, t_2 \text{ free for } x \text{ in } \phi}{\phi[t_2/x]} \text{ EQ-E}$$

In the above rule, $\phi[t/x]$ stands for substituting t for the free occurences of x in ϕ. And whenever we do such substitutions, we should remember that t should not have any variables which are already bound in ϕ (usually called, *t is free for x in ϕ*). This is exactly the side-condition of the rule above.
For example, in $(\exists y.(x < y))[y/x]$, doing a blind substitution would result in $\exists y.(y < y)$ which is absurd. The problem arose because y was already bound in the formula ($\exists y$). But that's usually not the intended purpose of the substitution. Suppose that we have a side-condition that *y is free for x in the formula*. As y is already bound in the formula, it is as if the bound variable is first changed, say to z, resulting in $\exists z.(x < z)$ which is essentially the same formula and then substituting y for x resulting in $\exists z.(y < z)$ which is what we desired!
In the rest of this note, we assume that *t is free for x* in such substitutions and not specify explicitly.
The two tactics above correspond to the tactics `reflexivity` and `rewrite`, respectively.

3.2 Universal Quantifier

$$\frac{\boxed{\begin{array}{r} x_0 (\textit{fresh variable}) \\ \vdots \\ \phi[x_0/x] \end{array}}}{\forall x. \phi} \text{ FORALL-I}/x_0$$

Note that the introduction rule for $\forall$ corresponds exactly to how we prove universal statements in mathematics. We let x_0 be an arbitrary entity and prove the statement in terms of x_0 wherever x appears in the statement. Then as x_0 was arbitrary, we conclude that the statement holds for every such x. That is exactly the rule above! As is the case with the rule of implication introduction, we should remember to *discharge* these fresh variable assumptions before the proof finishes. This corresponds to the tactic `intro` in Coq.

$$\frac{\forall x.\phi}{\phi[t/x]}\ \text{FORALL-E}$$

Here, t is any term which can be substituted for x. This implicitly assumes that when ϕ is true *for every* x, there *exists at least one* x for which ϕ is true. This corresponds to the tactic `apply` in Coq.

3.3 Existential Quantifier

$$\frac{\phi[t/x]}{\exists x.\phi}\ \text{EXISTS-I}$$

This rule says that if ϕ is true for some value of x, we can conclude that $\exists x.\phi$.

$$\frac{\exists x.\phi \qquad \boxed{\begin{array}{r} x_0(\textit{assumption}) \\ \phi[x_0/x] \rightarrow \chi(\chi \textit{ has no free occurences of } x_0) \end{array}}}{\chi}\ \text{EXISTS-E}$$

This rule says how to *use* the fact that $\exists x.\phi$. We *assume* that ϕ is true for x_0, a new variable. Then, if we can prove χ from $\phi[x_0/x]$ we can conclude that χ is true. Again, we should remember to *discharge* the assumptions before we end the proof. Note that, this parallels how we prove statements in mathematics. This corresponds to the tactic `destruct` in Coq.

4 Exercises

1. $(x+1=1+x),(x+1>1 \rightarrow x+1>0) \vdash (1+x>1 \rightarrow 1+x>0)$
2. $t_1 = t_2 \vdash t_2 = t_1$
3. $t_1 = t_2, t_2 = t_3 \vdash t_1 = t_3$
4. $\forall x.(P(x) \rightarrow Q(x)), \forall x.P(x) \vdash \forall x.Q(x)$
5. $P(t), \forall x.(P(x) \rightarrow \neg Q(x)) \vdash \neg Q(t)$
6. $\forall x.P(x) \vdash \exists x.P(x)$

7. $\forall x.(P(x) \to Q(x)), \exists x.P(x) \vdash \exists x.Q(x)$
8. $\forall x.(Q(x) \to R(x)), \exists x.(P(x) \land Q(x)) \vdash \exists x.(P(x) \land R(x))$
9. $\neg \forall x.P(x) \vdash \exists x.\neg P(x)$ (*not intuitionistic*)
10. $\exists x.\neg P(x) \vdash \neg \forall x.P(x)$
11. $(\forall x.P(x)) \land Q \vdash \forall x.(P(x) \land Q)$ (assume that Q does not depend on x.)
12. $\exists x.P(x) \lor \exists x.Q(x) \vdash \exists x.(P(x) \lor Q(x))$
13. $\exists x.\exists y.P(x, y) \vdash \exists y.\exists x.P(x, y)$

Some of these are done in Coq which you can find on the Lectures page.

5 References

1. *Logic in Computer Science: Modelling and Reasoning about Systems.* Michael Huth and Mark Ryan, Cambridge University Press.