COMP30026 Models of Computation

Predicate Logic: Syntax

Harald Søndergaard

Lecture 6

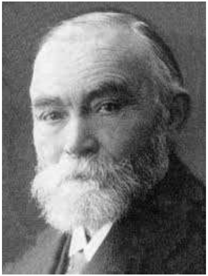
Semester 2, 2017

From Propositional to Predicate Logic

Propositional logic is useful for many purposes, but there is much in our everyday vocabulary (and in the mathematician's arguments) that it cannot express.

Propositional logic as we know it was developed early in the second half of the 19th century.

By 1879 Gottlob Frege had designed his "Begriffschrift", which was really the language of first-order predicate logic, although it looked very different, with statements being written as tree structures.



Why Predicate Logic?

Unlike propositional logic, predicate logic allows us to

- finitely express statements that deal with infinite collections of objects (integers, for example);
- express relations, capturing transitive verbs and relative pronouns.

To enable this, predicate logic uses variables that are assumed to range over collections of individuals, such as integers, graphs, people, or whatever, as well as quantifiers.

Propositional letters become generalised to predicates, that is, functions that map tuples of individuals to **f** or **t**.

Expressiveness of Predicate Logic

No emus fly: $\forall x\ (Emu(x) \Rightarrow \neg Flies(x))$

There are black swans: $\exists x\ (Black(x) \land Swan(x))$
or: $\exists y\ (Black(y) \land Swan(y))$

If all push the cart, the donkey will be happy: $\forall x\ (P(x)) \Rightarrow H$

If somebody pushes, the donkey will be happy: $\exists x\ (P(x)) \Rightarrow H$
or (strangely): $\forall x\ (P(x) \Rightarrow H)$

Expressing Relations

Tom found Rover and returned him to Anne:

$$Found(tom, rover) \wedge Gave(tom, rover, anne)$$

Tom found a dog and gave it to Anne:

$$\exists x \; (Dog(x) \wedge Found(tom, x) \wedge Gave(tom, x, anne))$$

Jill inhabits the house that Jack built:

$$\exists x \; (House(x) \wedge Inhabits(jill, x) \wedge BuiltBy(jack, x))$$

Mothers' mothers are grandmothers:

$$\forall x, y, z \; ((Mother(x, y) \wedge Mother(y, z)) \Rightarrow Grandmother(x, z))$$

Existential and Universal Quantification

Tom found an amount of money and gave it to Red Cross:

$$\begin{array}{l}(Found(tom, \$1) \wedge Gave(tom, \$1, redcross)) \vee \\ (Found(tom, \$2) \wedge Gave(tom, \$2, redcross)) \vee \\ \quad \vdots \end{array}$$

Existential quantification, $\exists$, is generalised $\vee$.

Universal quantification, $\forall$, is generalised $\wedge$.

Sidebar: Other Variable Binders

$\sum_{i=1}^{100} i^2$ is another example of variable binding.

Similarly with the lambda term $\lambda x.\ x^2 + 1$, or in Haskell notation

```
\x -> x^2 + 1
```

A lambda term allows us to create a function without giving it a name. This is surprisingly useful.

Mathematicians sometimes talk about "the function $x^2 + 1$", but really that notation is ambiguous, because it does not allow us to distinguish, say, $\lambda x.\ x^2 + 1$ and $\lambda xy.\ x^2 + 1$.

Mechanical Inference with Predicate Logic

Consider this argument:

- Every shark eats a tadpole.
- All large white fish are sharks.
- Colin is a large white fish living in deep water.
- Any tadpole eaten by a deep water fish is miserable.
- Therefore some tadpole is miserable.

Later we shall see how to automate reasoning about such arguments.

Our Vocabulary

The alphabet of a first-order language:

- variables (x, y, z, u, v, w,...)
- function symbols (f, g, h, ..., $+$, $\cdot$, ...)
- constants (a, b, c, ..., 0, 1, tom, ...)
- predicate symbols (P, Q, R, A, B, ..., $<$, $=$)
- connectives
- quantifiers
- parentheses
- (sometimes: **f**, **t**)

Each function symbol comes with an arity: a number that says how many arguments the function takes. Each predicate symbol similarly comes with an arity.

Terminology

A term is a variable or a constant or a construction

$$f(t_1, \ldots, t_n)$$

where $n > 0$, f is a function symbol of arity n, and each t_i is a term. (Or we may think of a constant as a function of arity 0.)

An atomic formula (or atom) is a construction $P(t_1, \ldots, t_n)$ where $n \geq 0$ and P is a predicate symbol of arity n.

Term	$\longleftrightarrow$	Individual, object
Atom	$\longleftrightarrow$	Assertion (false or true)

A literal is an atomic formula or its negation.

Case Matters

Note carefully the convention we adopt:
A predicate starts with an upper case letter; nothing else does.

So $father(ron)$ is a term; it denotes some object.

(Probably we intend: “the father of Ron”)

On the other hand, $Father(ron)$ is a formula; it denotes a truth value.

(Probably we intend: “Ron is a father”)

First-Order Predicate Logic: Syntax

Well-formed formulas (wffs) are generated by the grammar

$$
\begin{array}{rcl}
\mathit{wff} & \rightarrow & \mathit{atom} \\
& | & \neg\ \mathit{wff} \\
& | & \mathit{wff} \land \mathit{wff} \\
& | & \mathit{wff} \lor \mathit{wff} \\
& | & \mathit{wff} \Rightarrow \mathit{wff} \\
& | & \mathit{wff} \Leftrightarrow \mathit{wff} \\
& | & \mathit{wff} \oplus \mathit{wff} \\
& | & \forall \mathit{var}\ (\mathit{wff}) \\
& | & \exists \mathit{var}\ (\mathit{wff})
\end{array}
$$

Bound and Free Variables

A variable which is in the scope of a quantifier (binding that variable) is bound. If it is not bound then it is free.

A variable may occur both free and bound in a formula—witness y in

$$\forall z\ (P(x, y, z) \land \forall y\ (P(f(x), z, y)))$$

A formula with no free variable occurrences is closed.

It is possible for scopes to have "holes":

$$\forall x \exists y\ (x < y \land \exists x\ (y < x))$$

The last occurrence of x is bound by the closest quantifier, so the scope of $\forall x$ is not all of $\exists y(\ldots)$.

Bound Variable Renaming and Capture

The bound variable of a quantified formula is just a placeholder—its exact name is inessential.

$\exists x \forall y \ (x < y)$ means the same as $\exists x \forall z \ (x < z)$.

If a variable occurs bound in a certain expression then the meaning of that expression does not change when all bound occurrences of that variable are replaced by another one.

However, to avoid variable capture, we cannot change the variable bound by $\forall y$ to a variable in an enclosing scope:

$\exists x \forall y \ (x \leq y)$ is very different to $\exists x \forall x \ (x \leq x)$.

From English to Predicate Logic

Introduce symbols for predicates.

Sentence: "He is a man." Predicate: is a man Symbol: $M(\ )$

"x is a man", $M(x)$, cannot be assigned a truth value.

Kim is a man, $M(kim)$, can be assigned a truth value.

Sentence: "Bob is taller than Kim" $T(bob, kim)$

Quantifier examples:

"Every man is mortal"	$\forall x\ (Man(x) \Rightarrow Mortal(x))$
"Some cat is mortal"	$\exists x\ (Cat(x) \land Mortal(x))$

Usually use $\Rightarrow$ with $\forall$ and $\land$ with $\exists$.

Example Translations

Let $L(x, y)$ stand for "x loves y".
Let $I(x, y)$ stand for "x is y".

$L(bob, eva)$	Bob loves Eva
$\forall x\ L(x, eva)$	Everyone loves Eva (also Eva!)
$\forall x\ (\neg I(x, eva) \Rightarrow L(x, eva))$	Eva is loved by everyone else
$\exists x\ (\neg I(x, bob) \wedge L(x, bob))$	Someone other than Bob loves Bob
$\forall x\ (\exists y\ L(x, y))$	Everybody loves somebody
$\exists y\ (\forall x\ L(x, y))$	Someone is loved by everybody
$\exists x\ (\forall y\ L(x, y))$	Someone loves everybody

Quiz: Translate This

Translate the following statement to predicate logic:

Every Melburnian barracks for a footy team.

Use these predicates:

$M(x)$	x is a Melburnian
$T(x)$	x is a footy team
$B(x, y)$	x barracks for y

Quiz: Translate This

Translate the following statement to predicate logic:

Every Melburnian barracks for a footy team.

Use these predicates:

$M(x)$	x is a Melburnian
$T(x)$	x is a footy team
$B(x,y)$	x barracks for y

$$\forall x\ (M(x) \Rightarrow \exists y\ (T(y) \land B(x,y)))$$

or, equivalently:

$$\forall x\ \exists y\ (M(x) \Rightarrow (T(y) \land B(x,y)))$$

Word Order

Consider word order with care:

- "There is something which is not P": $\exists y\ \neg P(y)$
- "There is not something which is P" ("nothing is P"): $\neg\exists y\ P(y)$
- "All S are not P" vs "not all S are P:" $\forall x\ (S(x) \Rightarrow \neg P(x))$ or $\neg\forall x\ (S(x) \Rightarrow P(x))$???

Natural languages offer many examples of sentences that do not mean what they say; consider "All that glitters is not gold".

Quantifier Order

The order of different quantifiers is important.

$\forall x \exists y$ is not the same as $\exists y \forall x$.

The former says each x has a y that satisfies $P(x, y)$; the latter says there's an individual y that satisfies $P(x, y)$ for every x.

But $\forall x \forall y$ is the same as $\forall y \forall x$ and $\exists x \exists y$ is the same as $\exists y \exists x$.

Quantified Formulas as a Two-Person Game

The truth or falsehood of a quantified formula can be expressed as a question of winning strategies for a two-person game. Say I make a claim (the quantified statement) and you try to disprove it. You get to supply values for the universally quantified variables.

- If I claim $\forall x \exists y\ P(x, y)$, then you can challenge me by choosing an x and asking me to find the y that satisfies $P(x, y)$, but I get to know the x you chose.
- If I claim $\exists y \forall x\ P(x, y)$, then you can challenge me by asking me to provide the y, and then you just have to find an x that does not satisfy $P(x, y)$, knowing the y that I chose.
- If I claim $\exists x \exists y\ P(x, y)$, then I have to find both x and y, so it doesn't matter what order they appear.
- If I claim $\forall y \forall x\ P(x, y)$, then you get to pick both x and y, so again their order does not matter.

Implicit Quantifiers

Often quantifiers are implicit in English. Look for nouns (especially plural) without determiners (words to indicate which members of a group are intended).

"Men are mortal" means "all men are mortal":
$\forall x\ (Man(x) \Rightarrow Mortal(x))$

"A woman is stronger than a man" would usually mean:
$\forall x \forall y\ ((Woman(x) \wedge Man(y)) \Rightarrow Stronger(x, y))$

"If a girl owns a poodle, she spoils it":

$$\forall x \forall y\ ((Girl(x) \wedge Poodle(y) \wedge Owns(x, y)) \Rightarrow Spoils(x, y))$$

Reading Materials

O'Donnell, Hall and Page discuss predicate logic in Chapter 7, including translations from English. They also discuss Haskell's `forall` and `exists`.

In Section 7.3 they make use of a style of inference also known as "natural deduction" (not covered by us, and not examinable).

A rather different introduction to predicate logic is in Makinson's Chapter 9.

Coming Up

We next cover the semantics of first-order predicate logic in more detail.

After that we will want to extend the resolution principle to predicate logic.