CS1113
Multiple Quantifiers in Predicate Logic

Lecturer:

Professor Barry O'Sullivan

Office: 2.65, Western Gateway Building

email: *b.osullivan@cs.ucc.ie*

http://osullivan.ucc.ie/teaching/cs1113/

More than one quantifier

Understanding exactly what a quantifier is quantifying

Using multiple quantifiers

"somewhere in the world, a pedestrian is knocked down by a car every minute"

So far, all our example predicate logic statements have had a single quantified variable

e.g. $\forall x\ (uccStudent(x) \rightarrow uccCard(x))$

But the language definition doesn't restrict us:

2. If *W* is a wff, and *x* is a variable, then
   $\forall x\ W$ and $\exists x\ W$ are wffs

We can keep adding new quantifiers and variables onto the front, and we still get a well-formed statement.

What would it mean? How would we define which part of the sentence each quantifier applies to?

Quantifier Scope

In an expression in predicate logic, the *scope* of a quantifier is the sub-part of the expression to which it applies – either the predicate immediately following it, or a wff in brackets.

E.g. in $\forall x\ (P(x) \rightarrow \exists y\ Q(y))$

the scope of $\forall x$ is $(P(x) \rightarrow \exists y\ Q(y))$
the scope of $\exists y$ is $Q(y)$.

In $\forall x\ P(x) \rightarrow Q(y)$

the scope of $\forall x$ is $P(x)$

In quantified expressions, we can re-use the variable in the quantifiers if the scope of the quantifiers do not overlap.

e.g. $\forall x\ P(x) \wedge \exists y\ Q(y)$
and $\forall x\ P(x) \wedge \exists x\ Q(x)$ are logically equivalent.

Bindings and free variables

In a wff, a variable x is bound if it is inside the scope of either $\forall x$ or $\exists x$ (so it is quantified).

So in the wffs $\forall x\ P(x)$, $\exists x\ P(x)$, $\forall x\ Q(x,y)$ and $\exists x\ Q(x,y)$, x is a bound variable.

In a wff, a variable x is free if it is not bound (it is not quantified).

So in the wffs $P(x)$ and $Q(x,y)$, x is a free variable.

Note that in $\forall x\ P(y)$, $\exists x\ P(y)$, $\forall x\ Q(x,y)$ and $\exists x\ Q(x,y)$, y is a free variable – it is not in the scope of $\forall y$ or $\exists y$

More formal definitions of $\forall$ and $\exists$

Let W be some wff, and let x be a variable.

$\forall x\ W$ is true if and only if for each constant $a \in U$, when we replace all free occurrences of x in W by a, we get a true statement.

Note: free in W
(not free in $\forall x$ W)

- if we can find one or more constants $b \in U$ where this doesn't apply, then $\forall x\ W$ is false

$\exists x\ W$ is true if and only if there is at least one constant $a \in U$, such that when we replace all free occurrences of x in W by a, we get a true statement.

- if there is no constant $a \in U$ for which this applies, then $\exists x\ W$ is false

Consistent assignments in quantifier scope

Let *formula* be a wff in predicate logic in which a free variable x appears more than once. Then in $\exists x$ (*formula*) when we interpret it by assigning values to x, we must assign the same value to each occurrence of x
(and similarly for $\forall x$ (*formula*))

For example, in $\exists x\ (P(x) \wedge Q(x))$, we want to find at least one value v such that $(P(v) \wedge Q(v))$ is true.

In $\forall x\ (P(x) \rightarrow Q(x))$, if that is true, we must show that $(P(v_1) \rightarrow Q(v_1))$ is true, and $(P(v_2) \rightarrow Q(v_2))$ is true, and so on,

but we do not care about e.g. $(P(v_1) \rightarrow Q(v_2))$, because it has different values assigned to x.

Logically equivalent formulae

We saw the idea of logically equivalent formulae in propositional logic:

"Two formulas in propositional logic are equivalent if and only if they have the same truth functions. I.e. no matter what values we assign to the atomic propositions, the two formulas have the same truth value."

We can extend this to handle statements in predicate logic:

Two formulae are logically equivalent if and only if they have the same truth value regardless of what interpretation we give to the predicate symbols, or what domains we specify for the variables.

Example equivalences

$$\exists x\, (P(x) \lor Q(x)) \equiv (\exists x\, P(x)) \lor (\exists x\, Q(x))$$

$$\forall x\, (P(x) \land Q(x)) \equiv (\forall x\, P(x)) \land (\forall x\, Q(x))$$

We will demonstrate that the first one is true. To do this, we will show

(i) whenever the left hand side is true, the right hand side is also true, and
(ii) whenever the right hand side is true, the left hand side is also true.

We will do this without giving any interpretation for P or Q, and without specifying a domain for the variables.

Example Proof

(i) Suppose $\exists x\ (P(x) \vee Q(x))$ is true.
Then there is some value v in the domain so that $P(v) \vee Q(v)$ is true.
Therefore, from propositional logic truth tables, we know that either $P(v)$ is true, or $Q(v)$ is true (or both).
But then either $\exists x\ P(x)$ is true or $\exists x\ Q(x)$ is true.
Therefore, from truth tables, $(\exists x P(x)) \vee (\exists x Q(x))$ is true.

(ii) Suppose $(\exists x P(x)) \vee (\exists x Q(x))$ is true.
Therefore, either $\exists x\ P(x)$ is true or $\exists x\ Q(x)$ is true.
So there is some v such that $P(v)$ is true or there is some v such that $Q(v)$ is true.
So there is some v such that $P(v) \vee Q(v)$ is true.
Therefore, $\exists x\ (P(x) \vee Q(x))$ is true.

U = all students

younger(x,y): student x is younger than (or same age as) student y

So what does

$$\forall x\ \exists y\ younger(x,y)$$

mean?

And what about

$$\exists x\ \forall y\ younger(x,y)$$

$\forall x\, W$ is true if and only if for each constant $a \in$ U, when we replace all free occurrences of x in W by a, we get a true statement.	$\exists x\, W$ is true if and only if there is at least one constant $a \in$ U, such that when we replace all free occurrences of x in W by a, we get a true statement.

Nested Quantifiers

$\forall x\ \forall y\ P(x,y)$ says no matter which constant a we choose from U, $\forall y\ P(a,y)$ is true. That means no matter which b we choose from U, $P(a,b)$ is true.

- So no matter what pair of constants (a,b) we choose from U, $P(a,b)$ is true (and a and b might be same constant)

$\exists x\ \exists y\ P(x,y)$ says there exists at least one constant a $\in$U such that $\exists y\ P(a,y)$ is true. That means there exists at least one constant $b \in U$ such that $P(a,b)$ is true.

- So there is at least one pair of constants (a,b) in U such that $P(a,b)$ is true (and a and b might be the same constant)

Nested Quantifiers

$\forall x\, \exists y\, P(x,y)$ says no matter which constant *a* we choose from U, $\exists y\, P(a,y)$ is true. That means there exists at least one constant $b \in U$ such that $P(a,b)$ is true.

- So no matter what constant *a* we choose from U, we must then be able to find a constant $b \in$U such $P(a,b)$ is true (and *b* might be different for each different *a*).

$\exists x\, \forall y\, P(x,y)$ says there is at least one constant $a \in$U such that $\forall y\, P(a,y)$ is true. That means no matter which constant $b \in U$ we choose, $P(a,b)$ is true.

- So there is at least one constant $a \in$U, such that no matter what constant $b \in$U we choose, $P(a,b)$ is true (so *a* is the same for each different *a*).

Nested Quantifiers: example

Let the domain be Z (i.e. the integers = {...,-3,-2,-1,0,1,2, 3,...})

$\forall x\ \forall y\ x < y$ is false, since we could set x=3, y=2

$\forall x\ \exists y\ x < y$ is true – for any value of x that you give me, I can find a value of y such that x < y

$\exists x\ \forall y\ x < y$ is false – the statement says there is an integer *x* smaller than every other integer

$\exists x\ \exists y\ x < y$ is true, since we could set x = 2, y = 10

Quantifier Order vs Variable order in the predicate

The order of the quantified variables does **not have to match** the order that the variables appear in the predicate or wff that is being quantified.

It is OK to write $\exists y\ \forall x\ P(x,y)$

(it means there is some constant *a* such that $\forall x\ P(x,a)$ is true)

Often, you will have to write it in this order, to be able to say what you mean to say ...

Example Specifications

"Every current student in the UCC database must be registered for some degree program"

S = current students, D = degree programs, M = modules, U = $S \cup D \cup M$

registered(x,y): x is registered for *y*

$$\forall x \in S\ \exists y \in D\ registered(x,y)$$

"There is a module that is studied by every CS student"

enrolled(x,y): x is enrolled on y

$$\exists z \in M\ \forall x \in S\ (registered(x,CS) \rightarrow enrolled(x,z))$$

The order of quantifiers is important!

$\forall x \, \exists y \, P(x,y)$

To check this is true, we try the first value for *x*, and try to find a value for y that makes *P(x,y)* true; we then select the next value for *x*, and try to find a value for *y*; and repeat for each value of *x*. Note that the value of *y* may be different each time.

$\exists y \, \forall x \, P(x,y)$

To check this is true, we try the first value for *y*, and check that each value of *x* makes *P(x,y)* true; if that fails, we try the next value for *y*, and check that each value of *x* makes *P(x,y)* true; and we keep going until we find a value of y that works. So we are looking for a single value for *y* which works for <u>every</u> value of *x*.

Example

Translate the following into logic:

"*a pedestrian is knocked down by a car every minute*"

P = pedestrians, M = minutes, U = $P \cup M$
knockedDown(x,m): x is knocked down by a car in minute *m*.

Is the answer (a) or (b)?

(a) $\exists x \in P\ \forall m \in M\ knockedDown(x,m)$

(b) $\forall m \in M\ \exists x \in P\ knockedDown(x,m)$

Examples

Translate into English:

$$\exists x\ \forall y\ (notequal(x,y) \rightarrow \neg sameTown(x,y))$$

notequal(x,y): $x \neq y$
sameTown(x,y): x and *y* live in the same town
U is the set of all students in CS1113.

Translate into logic:

"every student knows the mobile phone number of some other student"

and assume the domain is the set of all students.

Next Lecture

interpreting statements with multiple quantifiers

formal arguments with quantifiers