CS1113
Writing Statements 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/

Writing Predicate Logic Statements

Equivalence: $\neg\forall x\, P(x) \equiv \exists x\, \neg P(x)$

Translating and understanding statements

The language of predicate logic (v1)

Two Equivalences

Let *P* be a predicate that acts on a single object from U

$$\neg\exists x\, P(x) \equiv \forall x\, \neg P(x)$$

"it is not true that there is an object x that has property P" is logically equivalent to "for every possible object x, the property P is not held by x

$$\neg\forall x\, P(x) \equiv \exists x\, \neg P(x)$$

"it is not true that all objects x have property P" is logically equivalent to "there is at least one object x which does not have property P

$$\neg\forall x\, \mathit{inWC2014}(x) \equiv \exists x\, \neg\mathit{inWC2014}(x)$$

Example

Let the domain be the set of students in the UCC database.
Let *president(x)* be the predicate which is true when *x* is the President of Ireland.

There is no student who is the President of Ireland.

Rewrite as: there does not exist a student such that the student is the President of Ireland.

In logic: $\neg\exists x \; president(x)$
which is equivalent to $\forall x \; \neg president(x)$

which says for each student, the student is not the President of Ireland.

Proof of: $\neg \exists x \, P(x) \equiv \forall x \, \neg P(x)$

∃ is the existential quantifier
$\exists x \, f_1$ says there is <u>at least one</u> value we could assign to *x* that would make f_1 true

Suppose I claim $\neg \exists x \, P(x)$

- Then I am claiming *it is not true there is at least one value I could assign to x that makes P(x) true .*
- So I am claiming *there is no value I could assign to x that makes P(x) true.*
- So I am claiming *every single value I try to assign to x fails to make P(x) true*, no matter what that value is
- But assigning a value to x in P(x) turns it into a proposition, which must be either true or false
- So I am claiming *every value I assign to x makes P(x) false*
- So I am claiming *every value I assign to x makes ¬P(x) true*
- So I am claiming $\forall x \; \neg P(x)$

Exercise: do the other direction

Class Exercise

What do the following say? How could you rewrite them?

$\neg\forall x\ (\mathit{european}(x) \rightarrow \mathit{inWC2014}(x))$

$\neg\exists x\ (\neg\mathit{inWC2014}(x) \wedge \mathit{sameGroup}(x, \mathrm{Spain}))$

Example

Let the domain be the set of all computers connected to the campus network. Translate the following into logic, and then rearrange it into an equivalent formula.

"There does not exist a computer which has no virus protection and which is accredited by the Computer Centre."

Review: the language of propositional logic

- to make things more precise, we defined a language of propositional logic, by specifying what were *well-formed formulae (wffs):*

- We use:
  - propositional symbols: {*p, q, r, ..., p', p'', ...*}
  - logical connectives: {$\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$}
  - punctuation: { *(* , *)* }

1. All propositional symbols on their own are wffs
2. if *p* and *q* are wffs, then so are
   (p), $(\neg p)$, $(p \wedge q)$, $(p \vee q)$, $(p \rightarrow q)$ and $(p \leftrightarrow q)$
3. nothing else is a wff unless it can be formed by repeatedly applying rules 1 and 2 above.

The language of predicate logic (version 1)

U is the universal set (or domain of discourse) of constants
V is a set of variables, such that $V \cap U = \phi$
Π is a set of predicate symbols such that $\Pi \cap V = \Pi \cap U = \phi$

1. if P is a predicate symbol $\in \Pi$ and $t_1, t_2, \ldots, t_i \in V \cup U$
   then $P(t_1, t_2, \ldots, t_i)$ is a well formed formula (wff)
2. If W is a wff, and x is a variable, then
   $\forall x\ W$ and $\exists x\ W$ are wffs
3. If W_1 and W_2 are wffs, then
   $\neg W_1$, $W_1 \wedge W_2$, $W_1 \vee W_2$, $W_1 \rightarrow W_2$ and $W_1 \leftrightarrow W_2$ are wffs
4. If W is a wff, then (W) is a wff
5. Nothing else is a wff

Exercise: What is wrong with these attempted statements?

sameGroup(England $\wedge$ Spain)
is NOT a well-formed formula – we cannot have connectives inside the brackets of a predicate.

beats(Brazil, $\exists$x)
is NOT a well-formed formula – we cannot have quantifiers inside the brackets of a predicate.

$\forall$*european*(x)
is NOT a well-formed formula – each quantifier must act directly on a variable.

$\forall$P P(Ireland, x)
is NOT a well-formed formula – we cannot quantify a predicate
(at least, not in CS1113, where we only look at "First Order" Predicate Logic)

Restricting the domain

Sometimes, we will want to make statements about a subset of the universal set.

Suppose U is the set of integers Z (= {...,-2,-1,0,1,2,...}).
How do we say for all x bigger than 1, $x^2 > x$?

For all integers x, if x is bigger than 1 then x^2 is bigger than x

$$\forall x \ \ ((x > 1) \rightarrow (x^2 > x))$$

To make this easier, we can change the notation, and write

$$\forall x{>}1 \ x^2 > x$$

(but formally, this should be written as we did above)

Translating from English: Example

"All students in UCC have a UCC ID card"

Let the domain, U, be the set of all people
Let $uccStudent(x)$ mean x is a UCC student
Let $uccCard(x)$ mean x has a UCC ID card

$$\forall x\ (uccStudent(x) \rightarrow uccCard(x))$$

Alternatively if S is the set of all UCC students, we could say

$$\forall x \in S\ \ uccCard(x)$$

and if $U = S$

$$\forall x\ \ uccCard(x)$$

More Examples

Write predicate logic statements for the following.

- state the universe, and any subsets of interest
- define the predicates
- write a logic statement for each example

Exercise

- Some students in UCC were born in County Kerry
- Some students take both CS1105 and MG1002
- All students registered for CS1105 have access to the CS1105 restricted web pages
- It is not true that all students are registered for both CS1105 and MG1002
- Some students are either not taking CS1105 or not taking MG1002
- All students in 1st year CS have a CS email address and a UCC email address
- All students connecting from a UCC IP address or who provide the correct password can access the CS1105 restricted web pages

Next lecture

Examples