Section 1.3: Quantifiers, Predicates, and Validity

January 18, 2021

Abstract

We now inject logical **variables** into the mix, and investigate wffs which describe properties of the domains of those variables in given "interpretations." We still test their truth values, either for the specific domain in question, or even in all domains (validity).

1 Predicates and quantifiers

- **quantifier**: describes **how many** objects in a given domain have a certain property.

  Examples:

  - **universal quantifier** - $\forall$ - "for any", "for every", "for all"
  - **existential quantifier** - $\exists$ - "there exists", "for at least one", "for some"

  Have you encountered these quantifiers before, in other courses? Lewis Carroll said that there are **three** types of propositions:

  a. Some cakes are nice.
  b. No cakes are nice.
  c. All cakes are nice.

  No cakes are nice $\Leftrightarrow$ All cakes are not nice.

  (Why do we need only **two** quantifiers? Because we have **negation**!)

- **predicate**: a property of a variable (e.g. "x is prime"), generally containing one or more variables (and perhaps some constants).

  We combine the quantifiers and predicates to create expressions (predicate wffs) such as

$$(\forall x)P(x)$$

Every x in the universe has property P.

which we then must *interpret*. For example, this might be said in the context of the integers, with $P(x)$ standing for "x is prime". (So this wff would be false in this context. The same wff would be true – but trivially – in the context of all prime numbers!)

There is nothing special about the variable x, so this wff is the same as $(\forall y)P(y)$, $(\forall z)P(z)$, etc. We say that x is a *dummy* variable.

Predicates may have any number of variables in them: the example above is a *unary* predicate, with only a single variable.

- Truth value hence now depends on the **Interpretation** of an expression:
  - **domain of interpretation** – a non-empty set to which the predicate expression is applied;
  - assignment of a property of the objects to each predicate in the expression;
  - assignment of particular objects to each constant symbol in the expression.

We start with something abstract, and replace it with concrete instances in a given context.

Example: Selections from #3, p. 50: What is the truth value of each in the interpretation where the domain is the integers?

$(\forall x)\left[(\exists y)(\underbrace{x+y=x}_{Q(x,y)})\right]$

a. $(\forall x)\,[(\exists y)(x+y=x)]$ T: Yes: can always pick $y=0$
b. $(\exists y)\,[(\forall x)(x+y=x)]$ T: $y=0$ $x=2$, $x+y=2+0=2=x$ ✓
c. $(\forall x)\,[(\exists y)(x+y=0)]$ T: additive inverses exist $x=2$, $y=-2$ works
d. $(\exists y)\,[(\forall x)(x+y=0)]$ F: is there some ~~wo~~ special y that works as additive inverse for all x? No
e. $(\forall x)\,[(\forall y)(x<y \vee y<x)]$ F: might pick $y=x$
f. $(\forall x)\,[x<0 \rightarrow (\exists y)(y>0 \wedge x+y=0)]$ T
pick an $x=-2$, then $y=2>0$ and $x+y=0$.

Example: #7(a, c), p. 51. For each wff, find an interpretation in which it is true, and one in which it is false.

a. $(\forall x)(\underbrace{[A(x) \vee B(x)] \wedge [A(x) \wedge B(x)]'}_{\text{exclusive or}})$

Domain: Integers
T: $A(x)$ - x is even ; $B(x)$ - x is odd
F: $A(x)$ - $x<20$; $B(x)$ - $x>10$

c. $(\forall x)[P(x) \rightarrow (\exists y)Q(x,y)]$ Domain: People

In exercise 7(c), the predicate $Q(x,y)$ is an example of a **binary** predicate.

T: $P(x)$ - x is a boy ; $Q(x,y)$ - y is x's mother.
F: $P(x)$ - x is a mom ; $Q(x,y)$ - y is x's son.

$(\forall x)$ $(\exists x)$ (uglier, but safer!)

- **scope** of a quantifier: the portion of a predicate to which the quantifier applies; indicated with parentheses or brackets (but these may be neglected **if the scope is clear**).

Always safer to over-parenthesize!

- **free variable**: a variable in a predicate wff outside the scope of a quantifier involving that variable.

Example: Ex. 9(c), p. 51. Indicate the scope of each quantifier in $(\exists x)[(\forall y)P(x,y) \wedge Q(x,y)]$.

scope of $(\exists x)$ $\quad$ $(\forall y)P(x,y)$ scope of y $\quad$ $Q(x,y)$ quantified by $(\exists x)$; y is free - interpreted as different from the quantified y

2 Translation of English statements into predicate wffs

2.1 "there exists" with "and" – $\exists$ with $\wedge$ "for all" with "implies" – $\forall$ with $\rightarrow$

As noted before, translation can be a very tricky business, but it's obviously an important one. As is often the case, the process of translation does not result in a unique expression: there may be several different ways to "say the same thing" in wffs.

Our author encourages us to remember that

- typically $\exists$ and $\wedge$ go together, whereas
- typically $\forall$ and $\rightarrow$ go together.

$\exists$ & $\rightarrow$ are dangerous together: $(\exists x)[P(x) \rightarrow Q(x)]$ If anything in the domain doesn't have P, then this is True!

I can't emphasize it enough! Also, a single English "formula" or theorem may be given by numerous wffs: the **redundancy in our connectives ensures that**.

Example: Ex. 21(c, e, g), p. 54. Write each statement as a predicate wff:

$L(x)$, $J(x)$, $W(x)$, $A(x,y)$ x admires y

c. Some lawyers admire only judges. $(\exists x)[L(x) \wedge (\forall y)[A(x,y) \rightarrow J(y)]]$ = friends, = friends

e. Only judges admire judges. $(\forall x)(\forall y)[J(y) \wedge A(x,y) \rightarrow J(x)]$

g. Some women admire no lawyer. $(\exists x)[W(x) \wedge (\forall y)[A(x,y) \rightarrow L(y)']]$

Example: Ex. 23(a-d), p. 54. Write each as a predicate wff: $B(x)$, $F(x)$, $L(x,y)$: x loves y

a. All bees love all flowers. $(\forall x)(\forall y)[B(x) \wedge F(y) \rightarrow L(x,y)]$

b. Some bees love all flowers. $(\exists x)[B(x) \wedge (\forall y)[F(y) \rightarrow L(x,y)]]$

c. All bees love some flowers. $(\forall x)[B(x) \rightarrow (\exists y)[F(y) \wedge L(x,y)]]$

d. Every bee hates only flowers.

$(\forall x)(\forall y)[B(x) \wedge L(x,y)' \rightarrow F(y)]$

2.2 Negation

Negation of predicate wffs: some cases are standard, e.g.

- The negation of "Every x has property A." is "There is an x which doesn't have property A."; or "There is an x which has property A'."

(Everything has property A)'

$$[(\forall x)A(x)]' \iff (\exists x)[A(x)']$$

There's something that doesn't have property A

- The negation of "There is an x which has property A." is "No x has property A."; or "Every x has property A'."

(There is something w/ property A)'

$$[(\exists x)A(x)]' \iff (\forall x)[A(x)']$$

For all x, none has property A.

In general, English makes negation kind of tricky. Watch your step!

Example: Ex. 27(c,d), p. 56. Negate each:

c. All people are tall and thin.

$[(\forall x)[Tall(x) \wedge Thin(x)]]'$

$(\exists x)[(Tall(x) \wedge Thin(x))']$

$(\exists x)[Tall(x)' \vee Thin(x)']$

Author's: There is someone who is short or fat.

d. Some pictures are old and faded.

3 Validity

$[(\exists x)[O(x) \wedge F(x)]]' \iff (\forall x)[(O(x) \wedge F(x))']$

$\iff (\forall x)[O(x)' \wedge F(x)']$

All pictures are either not old or not faded.

The truth value of a predicate wff depends on the interpretation, but there are some for which the wff is true independent of the interpretation. These are called **valid** predicate wffs (the analogue of tautology for propositional wffs).

Whereas we can check the "validity" of a propositional wff (just check the truth table to see if it's a tautology), there is no general check for the validity of a predicate wff, since it depends on context. In spite of that, there are some valid predicate wffs (context free truth!), as demonstrated in the text:

$$(\forall x)P(x) \to (\exists x)P(x)$$
$$(\forall x)P(x) \to P(a)$$
$$P(x) \to (Q(x) \to P(x))$$

Example: Ex. 33(d,e), p. 57. Explain why each wff is valid:

d. $A(a) \to (\exists x)A(x)$

If there is an element a in the domain with property A, then there exists some element in the domain with property A.

e. $(\forall x)[A(x) \to B(x)] \to [(\forall x)A(x) \to (\forall x)B(x)]$

If for every x in the domain having property A means you have property B, then if everything in the domain has property A, then " " " " " B.