Philosophy 240
Symbolic Logic

Russell Marcus
Hamilton College
Fall 2014

Class #23 - Translation into Predicate Logic II (§3.2)

‘Only’ as a Quantifier

- ‘Only Ps are Qs’ is logically equivalent to ‘all Qs are Ps’.
  - ‘All numbers divisible by two are even’ is logically equivalent to ‘Only even numbers are divisible by two’
- Only PQs are R’ is often the same as ‘All RQs are P’ but sometimes better as ‘All Rs are PQs’
  - Only famous men have been presidents.
    - $(\forall x)[(Px \supset (Mx \bullet Fx)]$
  - Only hard-working students take logic.
    - $(\forall x)[Lx \supset (Hx \bullet Sx)]$
  - There’s no *syntactic* or *grammatical* rule.

The Only

Often just an ‘All’

- The only people who came to the party were late.
  - All people who came to the party were late.
  - $(\forall x)(Cx \supset Lx)$
- The only people who came to the party were Al and Beth.
  - $(\forall x)[Cx \supset (Ax \vee Bx)]$
  - Wait until 3.11 for a translation using constants.
- The only way to know whether you’re translating correctly is to understand the logic of your assertions (and those of other folks) and to understand how **M** and, later, **F** work.
- ‘Only’ can be used in other ways.
  - Interesting more-formal paper topic

More than One Quantifier

- If anything is damaged, then everyone in the house complains.
  - $(\exists x)Dx \supset (\forall x)[(Ix \cdot Px) \supset Cx]$
- Either all the gears are broken, or a cylinder is missing.
  - $(\forall x)(Gx \supset Bx) \vee (\exists x)(Cx \cdot Mx)$
- Some philosophers are realists, while other philosophers are fictionalists.
  - $(\exists x)(Px \bullet Rx) \bullet (\exists x)(Px \bullet Fx)$
- 3.1.36: It’s not the case that all conventionalists are logical empiricists if and only if some holists are conventionalists.
  - $\sim[(\forall x)(Cx \supset Lx) \equiv (\exists x)(Hx \bullet Cx)]$
    - Or?
  - $\sim(\forall x)(Cx \supset Lx) \equiv (\exists x)(Hx \bullet Cx)$

Monadic and Relational Predicate Logics

- Predicate logic is monadic if the predicates only take one singular term.
- When predicates take more than one singular term, we call the predicates relational.
- Andrés loves Beatriz
  - Monadic: La
  - Relational: Lab
    - ‘Lxy’: x loves y:
- Relational predicates will allow us greater generality.
- We will look to reveal as much logical structure as we can.

Extensions of Monadic Predicate Logic

- Full First-Order Predicate Logic
- A specific predicate for identity
- Functors
- Second-order quantifiers (predicate variables)

Names of Languages We Will Study

- **PL**: Propositional Logic
- **M**: Monadic (First-Order) Predicate Logic
- **F**: Full (First-Order) Predicate Logic
- **FF**: Full (First-Order) Predicate Logic with functors
- **S**: Second-Order Predicate Logic

Languages and Systems of Deduction

- With PL, we used one language and one set of inference rules.
- But we can use the same language in different deductive systems and we can use the same deductive system with different languages.
- We will use **M** and **F** with the same deductive system.
  - We'll tweak one rule just a bit to accommodate the relational predicates.
- We will introduce new rules for the language **F**, covering a special identity predicate.
  - We'll call this a new deductive system, though it's again just a small extension.

Vocabulary of M

- Capital letters A...Z used as one-place predicates
- Lower case letters for singular terms
  - a, b, c,...u are used as constants.
  - v, w, x, y, z are used as variables.
- Five connectives: ~, •, ∨, ⊃ ≡
- Quantifier symbols: ∃, ∀
- Punctuation: ( ), [ ], { }

Toward the Formation Rules for M

- Formation rules for PL were pretty easy:
  1. A single capital English letter is a wff.
  2. If α is a wff, so is ~α.
  3. If α and β are wffs, then so are:
     (α · β)
     (α ∨ β)
     (α ⊃ β)
     (α ≡ β)
  4. These are the only ways to make wffs.
- For M, we need some further concepts:
  Scope
  Binding
  Open and Closed formulas

On Scope

- $(\forall x)(Px \supset Qx)$
  Every P is Q
- $(\forall x)Px \supset Qx$
  If everything is P, then x is Q
- The difference between these two expressions is the scope of the quantifier.

Scope of a Negation

(in propositional logic)

The **scope of a negation** (in **PL**) is whatever directly follows the tilde.

- If what follows the tilde is a single propositional variable, then the scope of the negation is just that propositional variable.
- If what follows the tilde is another tilde, then the scope of the first (outside) negation is the scope of the second (inside) negation plus that inside tilde.
- If what follows the tilde is a bracket, then the entire formula which occurs between the opening and closing of that bracket is in the scope of the negation.

$\sim\{(P \bullet Q) \supset [\sim R \vee \sim\sim(S \equiv T)]\}$

Scope of a Quantifier

If what follows the quantifier is a bracket, then any formulas that occur until that bracket is closed are in the scope of the quantifier.

If what follows the quantifier is a tilde, then the tilde and every formula in its scope is in the scope of the quantifier.

If what follows the quantifier is another quantifier, then the inside quantifier and every formula in the scope of the inside quantifier is in the scope of the outside quantifier.

Quantifier Scope Example

$$(\forall w)\{Pw \supset (\exists x)(\forall y)[(Px \bullet Py) \supset (\exists z)\sim(Qz \vee Rz)]\}$$

- $(\forall w)$
  - Widest scope
  - $\{Pw \supset (\exists x)(\forall y)[(Px \bullet Py) \supset (\exists z)\sim(Qz \vee Rz)]\}$
- $(\exists x)$
  - $(\forall y)[(Px \bullet Py) \supset (\exists z)\sim(Qz \vee Rz)]$
- $(\forall y)$
  - $[(Px \bullet Py) \supset (\exists z)\sim(Qz \vee Rz)]$
- $(\exists z)$
  - $\sim(Qz \vee Rz)$
  - Narrowest scope

Binding

- Quantifiers bind every instance of their variable in their scope.
- A **bound variable** is attached to the quantifier which binds it.
  1. $(\forall x)(Px \supset Qx)$
  2. $(\forall x)Px \supset Qx$
  - In 1, the ‘x’ in ‘Qx’ is bound.
  - In 2, the ‘x’ in ‘Qx’ is not bound.
- An unbound variable is called a **free variable**.
  3. $(\forall x)Px \vee Qx$
  4. $(\exists x)(Px \vee Qy)$
  - In 3, ‘Qx’ is not in the scope of the quantifier, so that ‘x’ is unbound.
  - In 4, ‘Qy’ is in the scope of the quantifier, but ‘y’ is not the quantifier variable, so is unbound.
- Free variables lack character.
- Sentences with free variables are meaningless.

Open and Closed Sentences

- Wffs that contain at least one unbound variable are called **open sentences**.
  - Ax
  - (∀x)Px ∨ Qx
  - (∃x)(Px ∨ Qy)
  - (∀x)(Px ⊃ Qx) ⊃ Rz
- If a wff has no free variables, it is a **closed sentence**, and expresses a **proposition**.
  - (∀y)[(Py • Qy) ⊃ (Ra ∨ Sa)]
  - (∃x)(Px • Qx) ∨ (∀y)(Ay ⊃ By)
- Both closed and open sentences may be wffs.
- Translations from English into **M** should ordinarily yield closed sentences.
- We will use open sentences during proofs.
- We will remember the character (existential or universal) of each variable.

Formation Rules for Wffs of M

1. A predicate (capital letter) followed by a constant or variable (lower-case letter) is a wff.

2. For any variable β, if α is a wff that does not contain either ‘(∃β)’ or ‘(∀β)’ then ‘(∃β)α’ and ‘(∀β)α’ are wffs.

3. If α is a wff, so is ~α.

4. If α and β are wffs, then so are:

   (α • β)
   (α ∨ β)
   (α ⊃ β)
   (α ≡ β)

5. These are the only ways to make wffs.

Atomic Formulas Subformulas

- A wff constructed only using rule 1 is called an **atomic formula**.
  - Pa
  - Qt
  - Ax
- A wff that is part of another wff is called a **subformula**.
  - A **proper subformula** of α is a subformula of α not identical to α.
- In ‘(Pa • Qb) ⊃ (∃x)Rx’, the following are all proper subformulas:
  - Pa
  - Qb
  - Rx
  - (∃x)Rx
  - Pa • Qb

Main Operators

- Quantifiers and connectives are called **operators**, or logical operators.
  - Atomic formulas lack operators.
  - The last operator added according to the formation rules is called the **main operator**.

Overlapping Quantifiers

- Not allowed
- $(\exists x)[Px \bullet (\forall x)(Qx \supset Rx)]$
  - ill-formed