Philosophy 240: Symbolic Logic

Russell Marcus
Hamilton College
Fall 2015

Class #40 - Functions

Business

- Functions today
  - Translation and derivation
- Second-order logic on Wednesday
  - Just translation
  - We won't do inferences
- Friday: review, maybe reflect
- Final is Monday the 14th, 7pm
  - Review session on Sunday the 13th, 4pm
  - Here
- Compensatory Exams
  - Up to two
  - Respond to email which I will send
- Course Evaluations
- What else?

A Motivating Argument for Functions

1. No odd numbers are even.
2. One is odd.
3. One is the square of one.

So, not all square numbers are even.

- We can regiment into **F**.
  1. $(\forall x)(Ox \supset {\sim}Ex)$
  2. Oo
  3. $(\exists x)[Sxo \bullet (\forall y)(Syo \supset y{=}x) \bullet x{=}o]$

  $/ \; {\sim}(\forall x)[(\exists y)Sxy \supset Ex]$
- But, there is a more efficient and fecund option.
- Take ‘the square of x’ as a function.

Functions

- A small extension of **F** introduces functors to represent functions.
- A function is a special kind of relation.
- An object may bear the same relation to various different objects
  - Laa, Lab, Lac...
  - Gab, Gcb, Gdb...
  - I am taller than lots of things and younger than lots of things.
  - I love several things.
- A function associates a given object (or given objects) with exactly one object.
  - An n-place relation in which one place of the relation is unique for given n-tuples of the other places.
  - It takes one or more arguments and returns a single output, called its range.
    - one-place functions take one argument
    - two-place functions take two arguments
    - *n*-place functions take *n* arguments

Functions Are All Over

- Mathematics
  - linear functions
  - exponential functions
  - periodic functions
  - quadratic functions
  - trigonometric functions.
- Science
  - force is a function of mass and acceleration
  - momentum is a function of mass and velocity
  - Your genetic code is a function of the codes of your biological parents.
- Logic
  - semantics for **PL**: truth functions
- Natural language
  - the (biological) father of
  - the (biological) teacher of

Functions and Uniqueness

- Functions have a unique, determinate value for any given input.
- Any human being has one biological father
  - Putting genetic engineering aside
- Most relations are not functions
  - A person can **love** or **be loved** (**know** or **be known**) by many people
  - Many different things can be **between** x and z.
- These are not functions
  - the parents of a
  - the classes that a and b share
  - the square root of x

Some Functions and Their Logical Representations

- the father of: $f(x)$
- the successor of: $g(x)$
- the sum of: $f(x,y)$
- the teacher of: $g(x_1...x_n)$
  - Given no team teaching!
- The truth value of the conjunction of a and b: $f(a,b)$

Vocabulary of FF

- Capital letters A...Z, used as predicates
- Lower case letters
  - a, b, c, d, e, i, j, k...u are used as constants.
  - **f, g, and h are used as functors.**
  - v, w, x, y, z are used as variables.
- Five connectives: $\sim$, $\bullet$, $\vee$, $\supset$ $\equiv$
- Quantifiers: $\exists$, $\forall$
- Punctuation: (), [], {}

N-Tuples

- A functor term is a functor symbol followed by an n-tuple of singular terms.
- An **n-tuple of singular terms** is an ordered series of terms.
  - Singular terms: constants, variables, or functor terms
  - ‘single’, ‘double’, ‘triple’, ‘quadruple’, etc. are n-tuples.
  - Functions can take any number of arguments.
- We use n-tuples in the semantics of relational predicates.
  - In the metalanguage
  - Often: <a, b, c>
- We will represent n-tuples in our object language merely by listing the terms separated by commas.
- Some n-tuples
  - a,b
  - a,a,f(a)
  - x,y,b,d,f(x),f(a,b,f(x))
  - a

Functor Terms

- If α is an n-tuple of singular terms, then the following are all **functor terms**:
  - f(α)
  - g(α)
  - h(α)
- Note that an n-tuple of terms can include functor terms.
  - ‘Functor term’ is defined recursively.
  - We allow composition of functions.
- We can refer to the grandfather of x using just the functions for father, e.g. ‘f(x)’, and mother, e.g. ‘g(x)’:
  - f(f(x))
  - f(g(x))
- Composition of mathematical functions
  - Take ‘h(x)’ to represent the square of x
  - ‘h(h(h(x)))’ represents the eighth power of x, i.e. $((x^2)^2)^2$.

Formation Rules for Wffs of FF

1. An n-place predicate followed by n singular terms (constants, variables, **or functor terms**) 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.

The scope and binding rules are the same for **FF** as they were for **M** and **F**.

FF: Semantics

- The semantics for **FF** are basically the same as for **F**.
- We insert an interpretation of function symbols.
  - Step 1. Specify a set to serve as a domain of interpretation, or domain of quantification.
  - Step 2. Assign a member of the domain to each constant.
  - **Step 3. Assign a (metalinguistic) function with arguments and ranges in the domain to each function symbol.**
  - Step 4. Assign some set of objects in the domain to each one-place predicate; assign sets of ordered n-tuples to each relational predicate.
  - Step 5. Use the customary truth tables for the interpretation of the connectives.
- The function assigned in Step 3 will be a function in the meta-language used to interpret the function in the object language.
  - Remember, functions are just a kind of relation.
  - They don’t need any new bells or whistles.

Translations Into FF

- Translation key:
  - Lxy: x loves y
  - f(x): the father of x
  - g(x): the mother of x
- Olaf loves his mother.
  - $Log(o)$
- Olaf loves his grandmothers.
  - $Log(g(o)) \bullet Log(f(o))$
- No one is his/her own mother.
  - $(\forall x){\sim}x=g(x)$
- No one is her/his own grandmother.
  - $(\forall x)[{\sim}x=g(f(x)) \bullet {\sim}x=f(f(x))]$

Functions and Mathematics

- Many simple concepts in arithmetic are functions.
  - addition
  - multiplication
  - least common multiple
- The most fundamental function in mathematics is the successor function.
- All other mathematical functions can be defined in terms of successor and other basic concepts.
- All of arithmetic can be developed from five basic axioms.
- **Peano's Axioms for Arithmetic**

P1: 0 is a number
P2: The successor (x') of every number (x) is a number
P3: 0 is not the successor of any number
P4: If x'=y' then x=y
P5: If P is a property that may (or may not) hold for any number, and if
i. 0 has P; and
ii. for any x, if x has P then x' has P;
then all numbers have P.

Peano's Axioms, Regimented

Key: a: zero; Nx: x is a number; f(x): the successor of x

P1: 0 is a number

P2: The successor (x') of every number (x) is a number

P3: 0 is not the successor of any number

P4: If x'=y' then x=y

P5: If P is a property that may hold for any number, and if
- i. 0 has P; and
- ii. for any x, if x has P then x' has P;

then all numbers have P.

P1. Na

P2. $(\forall x)(Nx \supset Nf(x))$

P3. $\sim(\exists x)(Nx \bullet f(x)=a)$

P4. $(\forall x)(\forall y)[(Nx \bullet Ny) \supset (f(x)=f(y) \supset x=y)]$

P5. $\{Pa \bullet (\forall x)[(Nx \bullet Px) \supset Pf(x)]\} \supset (\forall x)(Nx \supset Px)$

Some Number-Theoretic Statements

- Key:
  - o: one
  - f(x): the successor of x
  - f(x, y): the product of x and y
  - Ex: x is even
  - Ox: x is odd
  - Px: x is prime

1. One is not the successor of any number.

- $(\forall x)(Nx \supset \sim f(x)=o)$

2. If the product of a pair of numbers is odd, then the product of the successors of those numbers is even.

- $(\forall x)(\forall y)\{(Nx \bullet Ny) \supset [Of(x, y) \supset Ef(f(x), f(y))]\}$

3. There are no prime numbers such that their product is prime.

- $\sim(\exists x)(\exists y)[Nx \bullet Px \bullet Ny \bullet Py \bullet Pf(x, y)]$

Derivations Using Functions

- No new rules
- Functions are singular terms.
- A functor can be either a constant or a variable.
  - It depends on what the arguments of the function are.
- We can UI to a variable, or a function of a variable, or any complex function all of whose arguments are variables.
- For UG, if the arguments of a function are all variables, then we are free to use UG over the variables in that function.
  - If the arguments contain any constants, then we can not use UG.
  - The restrictions on UG continue to hold for variables which are arguments of a function.
    - CP and IP
    - If a constant is present when the variable is introduced
- For EI, we must continue always to instantiate to a new term.
  - A functor is not a new term if any of its arguments, or any of the arguments of any of its sub-functors, have already appeared in the derivation.

The Motivating Argument

1. No odd numbers are even.
2. One is odd.
3. One is the square of one.
So, not all square numbers are even.

1. $(\forall x)(Ox \supset {\sim}Ex)$
2. Oo
3. $o=f(o)$

$/ \ {\sim}(\forall x)Ef(x)$

More Derivations

1. $(\forall x)[Ax \supset Bxf(x)]$
2. $(\exists x)Af(x)$ / $(\exists x)Bf(x)f(f(x))$

---

1. $\sim(\exists x)Cx$ / $(\forall x)\sim Cf(x, g(x))$

---

1. $(\forall x)\{(Nx \bullet Gxt) \supset (\exists y)(\exists z)[Py \bullet Pz \bullet x=f(y, z)]\}$
2. $Nb \bullet Gbt$ / $(\exists x)(\exists y)(\exists z)[Nx \bullet Py \bullet Pz \bullet x=f(y, z)]$