Introduction to Quantification

∀ and ∃

William Starr

10.18.11

Announcements

10.18

1. The Take-Home part of the Midterm is due now!

Outline

What we've Done

A Bird's Eye View

- There's been a pattern to this course so far
- Here's we've done several times now:
  1. Hone in on a class of logically interesting sentences: *basic sentences, conjunctions, etc.*
  2. Learn how to represent them in FOL: $\mathsf{P}(\mathsf{n}), \wedge, \neg$, etc.
  3. Then use the **semantic methods of logic** to understand what these representations **mean**
     - Truth tables
     - Game rules
  4. Understanding their meaning allows us to say precisely what patterns of inference they validate
     - That is, which methods of proof (formal and informal) and inference steps they support

What we are Going to Do

A Bird's Eye View

- Things aren't really going to change
- Except that today we are going to meet our last class of logically interesting sentences
  - So-called quantificational sentences
- Quantificational sentences are pretty nuanced
- Indeed, we'll need the rest of the semester to run our routine on them:
  1. Learn how to represent them in FOL
  2. Learn what they mean
  3. Learn how to handle them in proofs
- Today, we'll learn the basics about FOL representation and meaning for quantificational sentences

Quantities

In Thought & Talk

- In our daily lives, we think & talk about quantities
  - Some money
  - Every ex-girlfriend
  - Two siblings
  - No friends
  - Many friends
- As it turns out, this thought & talk is governed by interesting logical principles
- These logical principles cannot be captured with the version of FOL that we've learned so far
- Before meeting a version of FOL that can, let's learn more about these quantifier phrases

Quantifiers Phrases

Reasoning about Quantifies

- In English, basic sentences are made by combining a verb phrase and an noun phrase
- So far, the only noun phrases we have in FOL are names: jay, kay, a, fluffy, etc.
- Reasoning about quantities involves new kind of noun phrase: quantifier phrase (aka determiner phrase)
- Let's look at some examples

Quantifiers

And Quantifier Phrases

(1) Some money is wasted

(2) Every magician is a vampire

(3) Two cats are meowing

(4) No friends showed up to George's party

(5) Many friends came to my party

- The above sentences contain quantifier phrases
- Simple quantifier phrases have two parts:
  1. A quantifier
  2. A noun
- How can we represent quantifiers and quantifier phrases in FOL?

Quantifiers in FOL

Meet ∀ and ∃

- Gottlob Frege (1876) came up with a way to represent some quantifiers in FOL
- We'll be using his method (but different notation)
- Involves introducing two quantifier symbols into FOL:
  The Universal Quantifier ∀ (*everything*)
  The Existential Quantifier ∃ (*something*)
- As it turns out, these two quantifier symbols plus the truth-functional connectives allow us to represent many different quantificational sentences

Quantifiers in FOL

Variables

- Quantifier symbols alone aren't enough
- We also need variables: x, y, z, ...
- Using these two tools we can represent quantified sentences in FOL:
  - Example:

Everything is a cube

$$\Downarrow$$

$$\forall x\, \mathsf{Cube}(x)$$

- What's the variable doing?
  - You can paraphrase *Everything is a cube* as *For every object x, x is cube*
  - Use of x corresponds to the use of x with ∀ above

The Game Plan

Quantifiers & Variables

- So much for the basics of how quantifiers and variables can be combined to represent quantificational sentences
- Now we'll learn more of the details about variables and quantifier symbols
- Think about what quantified sentences **mean**

Variables

Grammatical Details

- FOL has infinitely many variables:
  $t, u, v, w, x, y, z, t_1, \ldots, t_n, u_1, \ldots, u_n, v_1, \ldots, v_n, \ldots$
- Grammatically, variables are like constants
- They go in the slots of predicates:
  $\mathsf{Cube}(y), \mathsf{FrontOf}(u, v), \mathsf{Between}(z, u_{21}, w)$
- These formulas look like familiar atomic sentences
  - Except, there are variables where the constants normally go:
    $\mathsf{Cube}(a), \mathsf{FrontOf}(c, d), \mathsf{Between}(n_4, e, f)$

Variables
Semantic Details

- Grammatically, variables are like names
- But, semantically they are quite different
  - They are more like pronouns
- Names are used to refer to objects
- Variables are used as placeholders that indicate relationships between quantifiers and the argument positions of various predicates
- To understand this difference, it is necessary to see more of how the quantifier symbols work

The Universal Quantifier
The Basics

The Universal Quantifier ∀

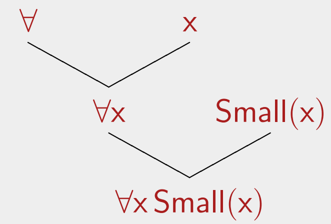
- ∀ expresses a universal claim like those expressed in English with *everything*, *each thing*, *all things* & *anything*
- ∀ is always used in tandem with a variable, e.g. ∀x Small(x)
- ∀x is read as *for every object x...*

The Universal Quantifier
Universal Statements

- How do you represent a universal statement in FOL?

Example

Everything is small:



1. Its a universal statement, so use ∀
2. Pick a variable to use, like x
3. Pair ∀ with that variable
4. Plug that variable into the predicate of the claim
5. Stick together the two things you've made

- We read ∀x Small(x) as *For every object x, x is small*
- Intuitive paraphrase of *Everything is small*

The Universal Quantifier
Restricting Universals

- A claim like *Everything is small* an unrestricted claim about absolutely any object
- We are rarely interested in claims about absolutely everything, since they're rarely true
- Rather, we are usually interested in restricted universal claims like *Every cube is small*
- How do we represent a claim like this in FOL?

Restricted Universals
How to Restrict?

(6) *Every cube is small*

- We can't represent (7) using the procedure from two slides ago because there are two predicates
- **Question**: how should these predicates be connected?
- Paraphrasing (7) with variables yields advice:
  For every object x, if x is a cube then x is small
- This suggests that we use → to connect the predicates:

$$\forall x\,(\text{Cube}(x) \rightarrow \text{Small}(x)))$$

Restricted Universals
Restrict with →

Restricted Universals are Universal Conditionals

- Restricted universals like *Every cube is small* are represented in FOL as universal conditionals
- We will discuss this more next class

The Existental Quantifier
The Basics

The Existential Quantifier ∃

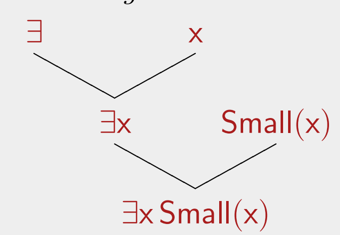
- ∃ expresses an existental claim like those expressed in English with *something*, *at least one thing*, *a* & *an*
- ∃ is always used in tandem with a variable, e.g. $\exists x\,\text{Small}(x)$
- $\exists x$ is read as *for some object x...*

The Quantifier
Existential Statements

- How do you represent a Existential statement in FOL?

Example

Something is small:



1. Its a existential statement, so use ∃
2. Pick a variable to use, like x
3. Pair ∃ with that variable
4. Plug that variable into the predicate of the claim
5. Stick together the two things you've made

- We read $\exists x\,\text{Small}(x)$ as *For some object x, x is small*
- Intuitive paraphrase of *Something is small*

The Existential Quantifier

Restricting Existentials

- A claim like *Something is small* an unrestricted claim about some object
- We are often interested in more descriptive claims about objects
- That is, in restricted existential claims like *Some cube is small*
- How do we represent a claim like this in FOL?

Restricted Existentials

How to Restrict?

(7) *Some cube is small*

- We can't represent (7) using the procedure from two slides ago because there are two predicates
- The question is how these two predicates should be connected
- Paraphrasing (7) with variables yields advice:

  For some object x, x is a cube and x is small
- This suggests that we use ∧ to connect the predicates:

$$\exists x\,(\mathsf{Cube}(x) \land \mathsf{Small}(x)))$$

Restricted Existentials

Restrict with ∧

Restricted Existentials are Existential Conjunctions

- Restricted existentials like *Some cube is small* are represented in FOL as existentials conjunctions
- We will discuss this more next class

Two Issues

An Overview

1. Syntax:

   The importance of and difference between free and bound variables
2. Scope:
   - What does it take for a quantifier and variable to be connected?

The Basic Issue

Mastering the use of quantifiers and the variables they 'hook up' with requires getting clear on some distinctions between different kinds of formulas. It also requires thinking more seriously about what it takes for a quantifier and variable to be hooked up

Syntax
Motivation

- There's a big difference between these two formulas:
  (8) Small(x)
  (9) Small(a)
- (9) makes a claim that is true or false
  - Either *a* is small or it isn't
- (8) does not
- x is a mere placeholder and does not refer to any object
- (8) is missing something; it's an incomplete claim
  - It's like saying *it is small* without telling us what *it* is!

Syntax
A Distinction

- But, not all formulas containing variables are incomplete:
  (10) ∃x Small(x)
- (10) makes a perfectly determinate claim
  - Namely: *something is small*
- Some terminology for making this distinction between complete and incomplete formulas:

Terminology (First Approximation)

1. *Sentences* are formulas that make complete claims
2. *Well-formed formulas* or *wffs* is the set of **all** grammatical expressions of FOL, including both incomplete claims (Tet(x)) and sentences

Grammatical Housekeeping
Making the Distinction Precise

- Okay, so we've said that sentences are formulas that express complete claims
- And, we've said that wffs are a larger class that include sentences as well as these incomplete formulas
- This is an important distinction because it marks an important semantic distinction
- For this reason, we will take a moment to be more precise about what exactly it is
- To do this, we need to do two things:
  1. Say more precisely what a wff is
  2. Say more precisely what a sentence is

Defining Well-Formed Formulas
A Comprehensive Recipe for Building Formulas

Definition of a Well-Formed Formula

1. If P is an n-ary predicate and $a_1, \dots, a_n$ are names, then $P(a_1, \dots, a_n)$ is a wff
2. If P is an n-ary predicate and $v_1, \dots, v_n$ are variables, then $P(v_1, \dots, v_n)$ is a wff
3. If A is a wff, so is $(\neg A)$
4. If $A_1, \dots, A_n$ are wffs, so is $(A_1 \wedge \dots \wedge A_n)$
5. If $A_1, \dots, A_n$ are wffs, so is $(A_1 \vee \dots \vee A_n)$
6. If A and B are wffs, so is $(A \rightarrow B)$
7. If A and B are wffs, so is $(A \leftrightarrow B)$
8. If A is a wff and v a variable, then $(\forall v A)$ is a wff
9. If A is a wff and v a variable, then $(\exists v A)$ is a wff

Defining Wffs
What Was That?

- We just saw a list of ways to build wffs
- If you can't build a given expression with those rules, it isn't a wff
  - Remember our policy on parentheses
- Let's look at some examples

Examples I
Wffs v. Non-Wffs

Wffs

(11) Tet(a)

(12) Cube(y)

(13) (Cube(y) ∧ Tet(a))

(14) (∃y (Cube(y) ∧ Tet(a)))

(15) (∃y Cube(y)) ∧ Tet(a)

(16) Tet(a) → (Cube(b) ∧ Small(b))

Non-Wffs

(17) Tet

(18) (y)Cube

(19) Cube(y, x)

(20) ∧Cube(y) Tet(a)

(21) ∃(Cube(y) ∧ Large(y))

(22) Tet(a) → Cube(b) ∧ Small(b)

- Now that we're clear on the wff v. non-wff distinction, let's draw the one we set out to draw
  - The wff v. sentence distinction

Defining Sentences
Freedom & Bondage

Definition of a Sentence

Any wff A which does **not** contain a **free variable** is called a *sentence*

- Now, I owe you a definition of what it counts for a variable to be free

Definition of Freedom

1. (Atomic) v is free in $P(v_1, \ldots, v_n)$ if $v = v_1, \ldots, v = v_n$
2. (Connectives) If v is free in A then it is still free in ¬A, A ∧ B, A ∨ B, A → B, A ↔ B
3. (Quantifiers) If v is free in A then it is still free in ∀x A and ∃x A, unless v = x, in which case v is *bound*

Sentences v. Wffs
Some Examples

Non-Sentence Wffs

(23) Tet(y)

(24) ¬Cube(y)

(25) (Cube(y) ∧ Tet(a))

(26) ((∃y Cube(y)) ∧ Tet(y))

(27) (∃y (Cube(y) ∧ Tet(x)))

- Free variables

Sentences

(28) Tet(a)

(29) ¬Tet(a)

(30) (Cube(a) ∧ Tet(a))

(31) (∃y (Cube(y) ∧ Tet(y)))

(32) (∃y (Cube(y) ∧ (∃x Tet(x))))

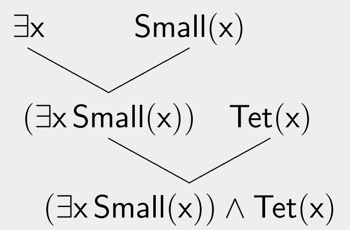
- No free variables

Scope

When a Quantifier Loves a Variable. . .

- In order for a quantifier to bind a variable, that variable must occur in the formula the quantifier attaches to

Where Quantifiers Bind



1. x was free in Small(x)
2. But, when ∃x was attached, that occurrence of x became bound
3. However, ∃x does not bind x in Tet(x)!

The Point Quantifiers only bind variables in the formula they immediately attach to

Scope

Some Terminology

Scope

1. A quantificational wff ∀v A is formed by sticking together some wff A and quantifer-phrase ∀v
2. We call A that quantifier's *scope*.
3. A quantifier can only bind (i.e. hook up with) a variable in it's scope.

- ∀x (Small(x) ∧ Tet(x)) says that everything is a small tet
- ∀x Small(x) ∧ Tet(x) says that everything is small and it is a tet (but doesn't say what it is)