Intermediate Logic
Lecture Four

First-Order Logic

Rob Trueman
rob.trueman@york.ac.uk

University of York

First-Order Logic

The Limits of TFL

- Consider the following obviously valid argument:
  - Sharon studies archaeology
  - Everyone who studies archaeology wishes that they were Indiana Jones
  - So Sharon wishes that she were Indiana Jones
- We cannot use TFL to show that this argument is valid
- The trouble is that, as far as TFL is concerned, the three sentences are all just atoms

  A: Sharon studies archaeology

  B: Everyone who studies archaeology wishes that they were Indiana Jones

  C: Sharon wishes that she were Indiana Jones

Splitting the (Logical) Atom

- To improve on TFL, we need to find a way of breaking atomic sentences down into subatomic units
  - An **atom** is a sentence which is not built out of any smaller *sentences*
  - In TFL, atoms have absolutely no internal structure
  - What we need is a logical system in which atomic sentences are built out of smaller *sub-sentential* expressions
- The system which does this is known as **First-Order Logic** (FOL)
  - This is the system you called Predicate Logic
  - We are calling it 'First-Order Logic' because there is another kind of predicate logic out there, called 'Second-Order Logic'

The Three Basic Building Blocks

- **Names**
  - Names in English: 'Gottlob Frege', 'Ludwig Wittgenstein', 'Rob Trueman'
  - Names in FOL: 'a', 'b', 'c', ... 'r'
- **Predicates**
  - Predicates in English: '__ is wise', '__ is human', '__ is a dog'
  - Predicates in FOL: 'A', 'B', 'C'...
- **Quantifiers**
  - Quantifiers in English: 'Everything', 'Something'
  - Quantifiers in FOL: '$\forall$', '$\exists$'

First-Order Logic

Names versus Singular Terms

- In general, a **singular term** is an expression which stands for a specific person, place or thing
  - 'Bertrand Russell' stands for a specific person, Bertrand Russell
  - 'The inventor of quantified logic' stands for a specific person, Gottlob Frege
- These two expressions are quite different:
  - 'Bertrand Russell' is a *proper name*; it's job is just to stand for Bertrand Russell
  - 'The inventor of quantified logic' is a *definite description*; it's job is to pick out whoever satisfies that description
- The **names** in FOL are meant to be symbolisations of proper names, not definite descriptions
  - We'll come back to definite descriptions later!

Names in FOL

- Names in FOL are lower case letters between 'a' and 'r', and if we want even more names, then we can add numerical subscritps (e.g. 'q_{27}')
- Each name stands for **exactly one** thing
  - There are no *ambiguous* names which sometimes refer to one thing, sometimes to another
  - However, there is nothing wrong with **one** object being referred to by **two** (or more!) names
- When we provide a symbolisation key for FOL, here is how we specify what each name refers to:

  b: Bertrand Russell
  f: Gottlob Frege

First-Order Logic

English Predicates

- The simplest predicates in English are expression which attribute properties to individuals; they let us say things about objects
- Here's an example of an English predicate: '__ is wise'
  - '__ is wise' attributes the property of wisdom
  - '__ is wise' says of an individual that they are wise
- Here's another example: '__ loves *Intermediate Logic*'
  - '__ loves *Intermediate Logic*' attributes the property of loving *Intermediate Logic*
  - '__ loves *Intermediate Logic*' says of an individual that they love *Intermediate Logic*

How to Make Predicates

- In general, you can think of predicates as things which combine with singular terms to make sentences
  - When you combine the predicate '__ is wise' with the name 'Socrates', you get the sentence 'Socrates is wise'
- Alternatively, you can think of a predicate as what you get when you remove a singular term from a sentence
  - Start with the sentence 'Daniel stole the ball from Simon'
  - If you remove 'Daniel', then you get: '__ stole the ball from Simon'
  - If you remove 'the ball', then you get: 'Daniel stole __ from Simon'
  - If you remove 'Simon', then you get: 'Daniel stole the ball from __'

Predicates of Higher Adicities

- The predicates that we have been looking at so far are all **monadic** predicate, meaning that they combine with just one name at a time
  - '__ is wise' has one gap for a name to be plugged into
- But other predicates combine with *more than one* name at a time
  - **Dyadic predicates** combine with *two* names at a time, e.g. '__ loves __'
  - **Triadic predicates** combine with *three* names at a time, e.g. '__ is between __ and __'
- We call the number of names that a predicate can combine with its **adicity**, and you can have predicates of any adicity whatsoever

Predicates in FOL

- Predicates in FOL are capital letters, and we can add numerical subscripts if we ever need more than 26 predicates (e.g. 'V_{342}')
- We also really need some way of indicating the adicity of each predicate; we will do that with numerical superscripts:
  - **Monadic predicates:**
    $A^1, B^1, \ldots, Z^1,\ A^1_1, B^1_1, \ldots, Z^1_1,\ A^1_2, B^1_2, \ldots, Z^1_2,\ \ldots$
  - **Dyadic predicates:**
    $A^2, B^2, \ldots, Z^2,\ A^2_1, B^2_1, \ldots, Z^2_1,\ A^2_2, B^2_2, \ldots, Z^2_2,\ \ldots$
  - ***n*-adic predicates:**
    $A^n, B^n, \ldots, Z^n,\ A^n_1, B^n_1, \ldots, Z^n_1,\ A^n_2, B^n_2, \ldots, Z^n_2,\ \ldots$

Symbolisation Keys for Predicates

- When we provide a symbolisation key for FOL, here is how we specify what each monadic predicate symbolises:

  A^1: __ is angry

  H^1: __ is happy

- So if 'g' symbolises 'Gottlob Frege', then'A^1g' symbolises 'Gottlob Frege is angry', and 'H^1g' symbolises 'Gottlob Frege is happy'

- And if 'b' symbolises 'Bertrand Russell', then 'A^1b' symbolises 'Bertrand Russell is angry', and 'H^1b' symbolises 'Bertrand Russell is happy'

Symbolisation Keys for Predicates

- Here is how to provide a symbolisation key for a dyadic predicate:

  L^2: $\_\__1$ loves $\_\__2$

- The little subscript numerals attached to the blanks are there to tell us the *order* in which 'L^2' applies to individuals
  - On this key, 'L^2' applies to the lover first, and to the beloved second
  - So 'L^2bg' symbolises 'Bertrand Russell loves Gottlob Frege'
- Contrast 'L^2' with 'M^2' on the following key:

  M^2: $\_\__2$ loves $\_\__1$

  - On this key, 'M^2' applies to the *beloved* first and the *lover* second
  - So 'M^2bg' symbolises 'Gottlob Frege loves Bertrand Russell'

Symbolisation Keys for Predicates

- Here is how to provide a symbolisation key for a dyadic predicate:

  K^2: $\_\__1$ kicks $\_\__2$

- The little subscript numerals attached to the blanks are there to tell us the *order* in which 'K^2' applies to individuals
  - On this key, 'K^2' applies to the kicker first, and to the kicked second
  - So 'K^2bg' symbolises 'Bertrand Russell kicks Gottlob Frege'
- Contrast 'K^2' with 'N^2' on the following key:

  N^2: $\_\__2$ kicks $\_\__1$

  - On this key, 'N^2' applies to the *kicked* first and the *kicker* second
  - So 'N^2bg' symbolises 'Gottlob Frege kicks Bertrand Russell'

Symbolisation Keys for Predicates

- Here is how to provide a symbolisation key for a dyadic predicate:

  Q^2: $\_\__1$ quizzes $\_\__2$

- The little subscript numerals attached to the blanks are there to tell us the *order* in which 'Q^2' applies to individuals
  - On this key, 'Q^2' applies to the quizzer first, and to the quizzed second
  - So 'Q^2bg' symbolises 'Bertrand Russell quizzes Gottlob Frege'
- Contrast 'Q^2' with 'R^2' on the following key:

  R^2: $\_\__2$ quizzes $\_\__1$

  - On this key, 'R^2' applies to the *quizzed* first and the *quizzer* second
  - So 'R^2bg' symbolises 'Gottlob Frege quizzes Bertrand Russell'

Let's Get Rid of those Superscripts!

- Strictly speaking, we need the superscript on an FOL predicate to tell us what its adicity is
- But in practice, we can usually tell what the adicity of a predicate is just by looking at how we actually use it
  - If I write '*Rab*', then unless I've messed up, '*R*' must be a dyadic predicate
  - Equally, '*S*' must be a triadic predicate if you give it the following entry in a symbolisation key:
    S: $\_\__1$ sold $\_\__2$ to $\_\__3$
- So from now on, we won't bother with those ugly superscripts unless we *really* have to

First-Order Logic

Quantifiers

- FOL has two basic **quantifiers**
  - The *existential* quantifier, '$\exists$', is the FOL for 'Something'
  - The *universal* quantifier, '$\forall$', is the FOL for 'Everything'
- A quantifier must always be followed by a **variable**
  - A variable is a lowercase letter from 's' to 'z', with subscripts if we need them (e.g. 'x_{3000}')
- Here is an example: '$\forall xHx$'
  - If 'H' is our sybmolisation for '__ is happy', then '$\forall xHx$' says that everyone is happy
  - You should think of the 'x' as a kind of placeholder: whoever we pick as x, x is happy
- If we wanted to say that *someone* was happy, we would write: '$\exists xHx$'

Domains of Quantification

- Very often, when we use the quantifier 'everyone' in English, we do not literally mean **everyone** in the *whole world*
- Normally, we are quantifying over a particular, limited **domain** of quantification
- Roughly, the domain of quantification is the collection of things we are talking about
- If we wanted to talk about the people in York, then we would pick the people in York to be our domain

  domain: people in York

- The quantifiers only quantify over things in the domain, and all our names need to pick out things in the domain

Scope

- Like other logical expressions, quantifiers come with a **scope**

  (1) If everyone is a singer, then Rob is a singer

  (2) Everyone is such that, if they are a singer, then Rob is a singer

- (1) is true: *everyone* includes me, so if everyone is a singer then I am a singer

- (2) is false: I am not a singer but Susanne Sundfør is; so it is not true of Susanne Sundfør that if she is a singer, then I am a singer!

- We can capture the difference between these two sentences in FOL by giving the universal quantifier different scope

  (1′) $\forall x Sx \rightarrow Sr$

  (2′) $\forall x(Sx \rightarrow Sr)$

Multiple-Generality

- Questions of scope become even more important when we are dealing with sentences which contain more than one quantifier:

  (1) Everyone loves someone

  (2) Someone is loved by everyone

- (1) means that each person loves someone, but leaves it open that different people may love different people

- (2) means that there is a single person who everyone loves

- We can capture the difference between these two sentences in FOL by giving the quantifiers different scope

  (1′) $\forall x \exists y Lxy$

  (2′) $\exists y \forall x Lxy$

First-Order Logic

An Example Symbolisation

domain: Everyone born after 1900

b: Bertrand Russell

g: Gottlob Frege

A: $\_\_$ is angry

R: $\_\__1$ respects $\_\__2$

M: $\_\__2$ loves $\_\__1$

An Example Symbolisation

domain: Everyone born after 1900

b: Bertrand Russell

g: Gottlob Frege

L: $\_\_$ is a logician

R: $\_\__1$ respects $\_\__2$

M: $\_\__2$ loves $\_\__1$

- Frege is angry, unless Russell respects him $\Rightarrow Ag \vee Rbg$
- Someone angry is loved by Frege $\Rightarrow \exists x(Ax \wedge Mxg)$
- Everyone is loved by someone $\Rightarrow \forall x \exists y(Myx)$

Practise!

- At this point, you may find it helpful to try the symbolisation exercises in §19 of *forall*χ
- **YOU WILL NOT BE ASSESSED ON YOUR ABILITY TO SYMBOLISE ENGLISH SENTENCES IN FOL**
- However, completing these practise exercises will remind you how FOL works, and that will help you understand what is to come

Some Tips for Good Symbolisations

- **(Usually) use conditionals in universal generalisations**
  - If you want to symbolise something of the form 'every F is G', then you should use symbolise it as follows: $\forall x(\mathcal{F}x \rightarrow \mathcal{G}x)$

- **(Usually) use conjunctions in existential generalisations**
  - If you want to symbolise something of the form 'some F is G', then you should use symbolise it as follows: $\exists x(\mathcal{F}x \wedge \mathcal{G}x)$

Some Tips for Good Symbolisations

- **Keep an eye on the scope of your quantifiers**
  - '$\forall x(Fx \rightarrow Ga)$' means something very different from '$\forall xFx \rightarrow Ga$'!
- **Keep an eye on the order of your quantifiers**
  - '$\forall x \exists y Rxy$' means something very different from '$\exists y \forall x Rxy$'

Some Tips for Good Symbolisations

- **You can use negation to 'flip' quantifiers**
  - In English, 'It is not the case that everything is worthwhile' implies 'Something is not worthwhile'
  - Similarly, in FOL, $\neg\forall x\mathcal{A}x$ implies $\exists x\neg\mathcal{A}x$
  - In English, 'It is not the case that something is worthwhile' implies 'Nothing is worthwhile' (i.e. 'Everything is not worthwhile')
  - Similarly, in FOL, $\neg\exists x\mathcal{A}x$ implies $\forall x\neg\mathcal{A}x$
- **Pick your domain of quantification carefully**
  - Remember that you can only talk about the things in your domain of quantification

First-Order Logic

A Limit of FOL So Far...

- Consider this English sentence:

  (1) Simon is mean to everyone

- On the face of it, it seems that we can easily symbolise this sentence:

  (2) $\forall xMsx$

- But (2) implies that Simon is mean to everyone, *including* Simon!

- That is not how we ordinarily hear (1): we normally take this to say that Simon is mean to everyone, *except* Simon

- But as it stands, FOL is unable to express this simple thought

Introducing Identity

- To deal with cases like this, we add an identity symbol to FOL

  $=$: $\_\__1$ is identical to $\_\__2$

- '$=$' is a dyadic predicate symbol, but unlike the other predicates it **has** to be used to express identity; we cannot change its meaning at any time

  (As a result, we don't need to bother including an entry for it in our symbolisation keys)

- Now return to this sentence:

  (1) Simon is mean to everyone

- We can symbolise it as:

  (2) $\forall x(\neg x = s \rightarrow Msx)$

There are at least...

- Consider these sentences:
  - (1) There is at least one apple
  - (2) There are at least two apples
  - (3) There are at least three apples
- Now that we have '=', we can symbolise these sentences in FOL:
  - (1′) $\exists x Ax$
  - (2′) $\exists x \exists y (Ax \wedge Ay \wedge \neg x = y)$
  - (3′) $\exists x \exists y \exists z (Ax \wedge Ay \wedge Az \wedge \neg x = y \wedge \neg y = z \wedge \neg z = x)$

There are at most...

- Consider these sentences:

  (1) There is at most one apple

  (2) There are at most two apples

- Now that we have '=', we can symbolise these sentences in FOL:

  (1′) $\neg\exists x\exists y(Ax \wedge Ay \wedge \neg x = y)$

  (2′) $\neg\exists x\exists y\exists z(Ax \wedge Ay \wedge Az \wedge \neg x = y \wedge \neg y = z \wedge \neg z = x)$

There are at most...

- Consider these sentences:

  (1) There is at most one apple

  (2) There are at most two apples

- Now that we have '=', we can symbolise these sentences in FOL:

  (1′) $\forall x \forall y((Ax \wedge Ay) \rightarrow x = y)$

  (2′) $\neg\exists x \exists y \exists z(Ax \wedge Ay \wedge Az \wedge \neg x = y \wedge \neg y = z \wedge \neg z = x)$

There are at most...

- Consider these sentences:

  (1) There is at most one apple

  (2) There are at most two apples

- Now that we have '=', we can symbolise these sentences in FOL:

  (1′) $\forall x \forall y((Ax \wedge Ay) \rightarrow x = y)$

  (2′) $\forall x \forall y \forall z((Ax \wedge Ay \wedge Az) \rightarrow (x = y \vee y = z \vee z = x))$

There are exactly...

- Consider this sentence:

  (1) There is exactly one apple

- (1) is the conjunction of these two sentences:

  (2) There is at least one apple

  (3) There is at most one apple

- So we can symbolise (1) in FOL as:

  (1′) $\exists x Ax \wedge \forall x \forall y((Ax \wedge Ay) \rightarrow x = y)$

There are exactly...

- Consider this sentence:

  (1) There is exactly one apple

- (1) is the conjunction of these two sentences:

  (2) There is at least one apple

  (3) There is at most one apple

- So we can symbolise (1) in FOL as:

  (1′) $\exists x(Ax \wedge \forall y(Ay \rightarrow x = y))$

There are exactly...

- Consider this sentence:

  (1) There is exactly one apple

- (1) is the conjunction of these two sentences:

  (2) There is at least one apple

  (3) There is at most one apple

- So we can symbolise (1) in FOL as:

  (1′) $\exists x \forall y (Ay \leftrightarrow x = y)$

Definite Descriptions

- Definite descriptions are expressions like 'the *F*'
  - The inventor of quantified logic
  - The present Queen of England
  - The present King of France
- On the face of it, they look like singular terms, i.e. expressions which stand for objects
- But Russell famously insisted that they were not
- We will not now look at Russell's reasons for this, but will just show how we can neatly formulate Russell's approach to definite descriptions in FOL

Russell's Theory of Definite Descriptions

- The Queen of England is having lunch
  - (a) There is at least one queen of England; and
  - (b) There is at most one queen of England; and
  - (c) Every queen of England is having lunch

- The author of *Harry Potter* is very rich
  - (a) There is at least one author of *Harry Potter*; and
  - (b) There is at most one author of *Harry Potter*; and
  - (c) Anyone who authored *Harry Potter* is very rich

Russell's Theory of Definite Descriptions

- The F is G
  - (a) There is at least one F; and
  - (b) There is at most one F; and
  - (c) All Fs are Gs
- In a short sentence:
  - There is exactly one F, and it is G
- In formal symbols:
  - $\exists x(Fx \wedge \forall y(Fy \rightarrow y = x) \wedge Gx)$
  - $\exists x(\forall y(Fy \leftrightarrow y = x) \wedge Gx)$

Practise!

Why not take a break to complete the exercises at the end of §21 of *forallx*?

First-Order Logic

The Vocabulary of FOL

Predicates	$=$, A^1, B^1, C^1,... A^2, B^2, C^2,...
(with subscripts if needed)	A^1_1, B^1_1,... A^2_1... A^2_2, B^2_2...
Names	a, b, c,... r
(with subscripts if needed)	a_1, b_1, c_1,... a_2, b_2...
Variables	s, t, u, v, w, x, y, z
(with subscripts if needed)	s_1, t_1,... s_2, t_2...
Connectives	$\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$
Brackets	(,)
Quantifiers	$\forall$, $\exists$

Terms

- It is easy to define what counts as a sentence of TFL
  - All atomic sentences are sentences of TFL, and whenever you use a TFL connective to combine sentences of TFL, you end up with another sentence of TFL
- Things are a bit more fiddly in FOL, and we need to build up to the definition of a FOL sentence in a few steps
- First, we define a term:
  - A **term** is any name or any variable

Atomic Formulae

- Now that we have the notion of a term, we need to define what it is for a string of symbols to be a **formula** of FOL
- We define this via *recursion*:
  - We start by defining what it is for a string of symbols to be an **atomic** formula
  - Then we explain how to build bigger formulae out of smaller formulae
- Here is the definition of an atomic formula:
  1. If $\mathcal{R}^n$ is an n-adic predicate and $t_1, t_2, ..., t_n$ are terms, then $\mathcal{R}^n t_1, t_2, ..., t_n$ is an atomic formula
  2. If t_1 and t_2 are terms, then $t_1 = t_2$ is an atomic formula
  3. Nothing else is an atomic formula

Formulae

1. Every atomic formula is a formula.
2. If $\mathcal{A}$ is a formula, then $\neg\mathcal{A}$ is a formula
3. If $\mathcal{A}$ and $\mathcal{B}$ are formulae, then $(\mathcal{A} \wedge \mathcal{B})$ is a formula
4. If $\mathcal{A}$ and $\mathcal{B}$ are formulae, then $(\mathcal{A} \vee \mathcal{B})$ is a formula
5. If $\mathcal{A}$ and $\mathcal{B}$ are formulae, then $(\mathcal{A} \rightarrow \mathcal{B})$ is a formula
6. If $\mathcal{A}$ and $\mathcal{B}$ are formulae, then $(\mathcal{A} \leftrightarrow \mathcal{B})$ is a formula
7. If $\mathcal{A}$ is a formula, χ is a variable, $\mathcal{A}$ contains at least one occurrence of χ, and $\mathcal{A}$ contains neither $\forall\chi$ nor $\exists\chi$, then $\forall\chi\mathcal{A}$ is a formula
8. If $\mathcal{A}$ is a formula, χ is a variable, $\mathcal{A}$ contains at least one occurrence of χ, and $\mathcal{A}$ contains neither $\forall\chi$ nor $\exists\chi$, then $\exists\chi\mathcal{A}$ is a formula
9. Nothing else is a formula

Free and Bound Variables

- Here are two formulae:

  (1) Rax

  (2) $\exists xRax$

- These are both formulae, but there is an important difference between them:
  - The variable 'x' is governed by a quantifier in (2), but not in (1)
  - We say that 'x' is *bound* by a quantifier in (2), but it is *free* in (2)

- We can give a formal definition of binding, but first we need to extend our old definition of a **main logical operator**

The Main Logical Operator

- The **main logical operator** of a formula is the last operator that was used in the construction of that formula
- It is the operator which governs, or controls, the whole formula
- In TFL, the main logical operator was always a connective; now it can be a connective or a quantifier

The Main Logical Operator

- Here is how to find the **main logical operator** in a formula:
  - First, make sure you have included *all* the brackets, even the ones you can normally get away with omitting
  - Now check if the first symbol in the formula is '$\neg$'; if so, then that '$\neg$' is the main logical operator
  - If not, then check if the first symbol in the formula is a quantifier; if so, then that quantifier is the main logical operator
  - If the first symbol isn't a '$\neg$' or a quantifier, then starting counting brackets. Open-brackets '(' are worth $+1$, close-brackets ')' are worth -1. The first connective you hit which isn't a '$\neg$' or a quantifier when your count is at exactly 1 is the main logical operator
- An example:
  - $({}_1\exists x \neg ({}_2 Fx \wedge Gx)_1 \rightarrow \forall y \neg Fy)_0$

Binding and Scope

- The **scope** of a logical operator in a formula is the subformula for which that operator is the main logical operator
- **Exercise:** what is the scope of each quantifier in the following?

$$\forall x \exists y (Fx \leftrightarrow \forall z (Gayz \rightarrow Syzyayx))$$

- Now we can define what it is for a variable to be bound:
  - A **bound variable** is an occurrence of a variable χ that is within the scope of $\forall\chi$ or $\exists\chi$
  - A **free variable** is any variable that is not bound

From Formulae to Sentences

- We can finally define what counts as a sentence in FOL:
  - A **sentence** of FOL is any formula of FOL that contains no free variables
- **Exercise:** which of the following formulae are sentences?

(1) Rab

(2) Rax

(3) $\exists xRax$

(4) $\exists x(Rax \wedge Rbx)$

(5) $\exists xRax \wedge Rbx$

From Formulae to Sentences

- It is important not to forget the difference between sentences and mere formulae
  - Only sentences can be **true** or **false**; no other formula is truth-evaluable
- But to see how that works, we need to look at the **semantics** for FOL
- We will do that in Lecture 5, so please read §§23–25 of *forall*χ before then