Intermediate Logic
Lecture Five

Interpretations

Rob Trueman
rob.trueman@york.ac.uk

University of York

Exercises!

1. $R \vee (\neg Q \wedge \neg S),\ \neg R \rightarrow (Q \vee S) \vdash (R \vee S) \vee Q$
2. $\neg P \leftrightarrow Q \vdash (P \vee Q) \wedge \neg(P \wedge Q)$
3. $\vdash \neg((P \rightarrow Q) \rightarrow P) \vee P$

Interpretations

Atomic Sentences

- You can make an atomic sentence by combining an *n*-adic predicate with *n* names
  - F^1a; R^2ab; S^3abc
- You can also make an atomic sentence by putting two names either side of the identity sign
  - $a = b$

The Truth-Functional Connectives

- You can use the truth-functional connectives to make new sentences out of old ones
  - If $\mathcal{A}$ is a sentence, then $\neg\mathcal{A}$ is a sentence
  - If $\mathcal{A}$ and $\mathcal{B}$ are both sentences, then $\mathcal{A} \wedge \mathcal{B}$ is a sentence
  - If $\mathcal{A}$ and $\mathcal{B}$ are both sentences, then $\mathcal{A} \vee \mathcal{B}$ is a sentence
  - If $\mathcal{A}$ and $\mathcal{B}$ are both sentences, then $\mathcal{A} \rightarrow \mathcal{B}$ is a sentence
  - If $\mathcal{A}$ and $\mathcal{B}$ are both sentences, then $\mathcal{A} \leftrightarrow \mathcal{B}$ is a sentence

From Sentences to Formulae

- You can make a formula out of a sentence by replacing one or more names with variables
  - $F^1a \Rightarrow F^1x$
  - $R^2ab \Rightarrow R^2xb$
  - $R^2ab \Rightarrow R^2ax$
  - $R^2ab \Rightarrow R^2xy$
- These formulae are not *sentences*; they do not *mean* anything as they stand

From Formulae to Sentences

- You can turn a formula back into a sentence by adding quantifiers at the front which "bind" all of the variables in the formula
  - $F^1a \Rightarrow \exists x F^1 x$
  - $R^2ab \Rightarrow \forall x R^2 xb$
  - $R^2ab \Rightarrow \forall x R^2 ax$
  - $R^2ab \Rightarrow \exists x \forall y R^2 xy$

- A **bound variable** is an occurrence of a variable χ that is within the scope of $\forall\chi$ or $\exists\chi$
  - 'x' is bound in '$\forall x Rxy$', but the 'y' is free
  - The first 'y' is bound in '$\exists y Fy \wedge Gy$', but the second 'y' is free

- A variable is **free** iff it is not bound; sentences do not contain *any* free variables

The Main Logical Operator in FOL

- The **scope** of a logical operator in a formula is the subformula for which that operator is 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 in FOL

- 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:
  - $\exists x \neg (Fx \wedge Gx) \rightarrow \forall y \neg Fy$

The Main Logical Operator in FOL

- 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:
  - $(\exists x \neg (Fx \wedge Gx) \rightarrow \forall y \neg Fy)$

The Main Logical Operator in FOL

- 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:
  - $({}_1\exists x\neg({}_2Fx \wedge Gx)_1 \rightarrow \forall y\neg Fy)_0$

Interpretations

What is a Semantics?

- The aim of this lecture is to present a **semantics** for FOL
- You can think of a **semantics** as a method for assigning *truth-values* to sentences of arbitrary complexity
- You already know how the semantics for TFL works
  - You assign truth-values however you like to the atomic sentences, i.e. you give a valuation
  - Then you use the truth-tables to figure out what truth-values the more complex sentences get on that valuation
- What we need now is a method for assigning truth-values to the sentences of FOL

Introducing Interpretations

- The first thing we need to do is find a way of giving meaning to the building blocks of our sentences
- In TFL, that was just a matter of assigning truth-values to the atomic sentences, but now we have split the atoms into sub-sentential bits:
  - Names; Predicates; Quantifiers
- So what we need to do is fix the meanings of these bits:
  - We need to pick objects for the names to refer to
  - We need to pick extensions for the predicates to be true of
  - We need to pick a domain for the quantifiers to quantify over
- We call a specification of all these things an **interpretation**

Names

- Each name we are dealing with must be assigned something to refer to
- We can specify what each name is referring to like this:

  c: Chris Jay

  l: Louise Richardson

  m: Mary Leng

- So far, this looks **exactly** like a symbolisation key!

Predicates

- Each predicate must be assigned an extension
  - A predicate's **extension** is the collection of things it is true of

- Again, we can achieve this with a symbolisation key:

  W: ___ is wise

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

  P: $\_\_\__1$ mentioned $\_\_\__2$ to $\_\_\__3$

- You should understand these entries like this:

  'W' is to have the same extension as the English predicate '___ is wise'

  'R' is to have the same extension as the English predicate '$\_\_\__1$ respects $\_\_\__2$'

  'P' is to have the same extension as the English predicate '$\_\_\__1$ mentioned $\_\_\__2$ to $\_\_\__3$'

The Exception: Identity

- I just said that you have to assign an extension for every predicate you are dealing with, but there is one exception
- You do not need to specify an extension for the identity sign, '$=$'
- '$=$' always means *identity* on every extension

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

Domains

- Every interpretation needs to include a specification of the domain of quantification
- The domain can contain *anything* you like
- We can specify our domain like this:
  - domain: people in York
- It is important to remember that the domain **must** contain every object referred to by a name on the interpretation!

Symbolisation Keys as Interpretations

- An **interpretation** is a specification of these three things:
  (1) The referent of each name we are dealing with
  (2) The extension of each predicate we are dealing with
  (3) The domain of quantification
- Clearly, then, writing out a symbolisation key is one way of presenting an interpretation
- **But it is not the only way!**

Interpretations

Direct versus Indirect

- Symbolisation keys specify a predicate's extension only **indirectly**
- A symbolisation key does not actuall tell you what is in the extension of a predicate
- It just tells us that a certain FOL predicate has the same extension as an English predicate

  W: __ is wise

  'W' is to have the same extension as the English predicate '__ is wise'

- Sometimes it is more useful to specify a predicate's extension **directly**

The Direct Method

- The direct method is just to list the objects that are to be in the extension of a given predicate
- You can pick any objects you like: they do not need to have anything in common

  H: Danny DeVito, the number π, Einstein's Nobel Prize

- Sometimes we want to use **empty** predicates, i.e. predicates which are not true of anything
- If you are stipulating extensions directly, and want 'P' to be empty, just write:

  P:

Two-Place Predicates

- If you want to directly stipulate an extension for a **two-place** predicate, then you do it like this:

  R: $\langle \text{Danny DeVito}, \pi \rangle$, $\langle \pi, \text{Einstein's Nobel Prize} \rangle$

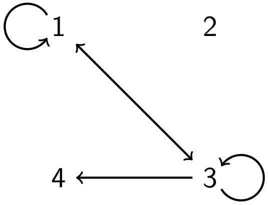
- The idea is that 'R' is true of each pair of objects, in the order they have been written

- We extend this to predicates with more gaps in the obvious way

  P^3: $\langle$Danny DeVito, π, Einstein's Nobel Prize$\rangle$

Diagrams

- Some people like to think in diagrams; if you are like that, then you can directly stipulate an extension for a two-place predicate with a diagram:



R: $\langle 1, 1\rangle$, $\langle 1, 3\rangle$, $\langle 3, 1\rangle$, $\langle 3, 3\rangle$, $\langle 3, 4\rangle$

Using Numbers

- In that last example, I used numbers as the objects in the extension of 'R'
- It is *very* common practice to use numbers when directly stipulating the extension of a predicate
- If you are using the direct method to specify the extension of a predicate, I recommend that you use numbers too

Domains and the Direct Method

- You can also specify the domain of an interpretation directly:

  domain: Danny DeVito, Micahel Keaton, π, Einstein's Nobel Prize

- It is important to remember that the domain **must** contain every object referred to by some name on the interpretation!

Putting it All Together

domain: 0, 1

a: 0

b: 1

c: 1

F: 0, 1

G: 1

R: $\langle 0, 1 \rangle$, $\langle 1, 0 \rangle$

H^1:

S^2:

Interpretations

Three Kinds of Sentence

- We want to see how we can use an interpretation to assign truth-values to the sentences of FOL
- There are three kinds of sentence:
  - Atomic sentences
  - Sentences whose main logical operator is a sentential connective
  - Sentences whose main logical operator is a quantifier
- We will look at each kind of sentence one by one

Our Example Interpretation

domain: all people born before 2000CE

h: Tom Holland

g: Lady Gaga

S: ___ is a singer

B: $\_\_\__1$ was born before $\_\_\__2$

Atomic Sentences

- Let $\mathcal{R}^n$ be an n-place predicate, and a_1, a_2,..., a_n be names
  - $\mathcal{R}^n a_1 a_2, \ldots, a_n$ is true in an interpretation iff $\mathcal{R}$ is true of the objects named by a_1, a_2,..., a_n in that interpretation (in that order)

- **Exercise:** Which of these atomic sentences are true on our Example Interpretation?

(1) Sh

(2) Sg

(3) Bhg

(4) Bgh

Identity

- It is worth recalling at this point that the identity sign, '$=$' is forced to express *identity* in every interpretation
- So if a and b are names, we can say:
  - $a = b$ is true in an interpretation iff a and b name the very same object in that interpretation

Sentential Connectives

- $\neg\mathcal{A}$ is true in an interpretation iff $\mathcal{A}$ is not true in that interpretation
- $\mathcal{A} \wedge \mathcal{B}$ is true in an interpretation iff $\mathcal{A}$ is true in that interpretation and $\mathcal{B}$ is true in that interpretation
- $\mathcal{A} \vee \mathcal{B}$ is true in an interpretation iff $\mathcal{A}$ is true in that interpretation or $\mathcal{B}$ is true in that interpretation (or both)
- $\mathcal{A} \rightarrow \mathcal{B}$ is true in an interpretation iff $\mathcal{A}$ is false in that interpretation or $\mathcal{B}$ is true in that interpretation (or both)
- $\mathcal{A} \leftrightarrow \mathcal{B}$ is true in an interpretation iff $\mathcal{A}$ and $\mathcal{B}$ have the same truth-value in that interpretation

Quantifiers: Some False Starts

- It turns out to be a bit tricky to say how to assign truth-values to sentences whose main logical operator is quantifier
- You want to say something like this:
  - '$\forall x Sx$' is true in an interpretation iff 'S' is true of everything in the domain of that interpretation
- That works great when we have just put a quantifier together with a predicate, but what about examples like the following?
  - '$\forall x(Sx \rightarrow Bxh)$' is true on an interpretation iff '$Sx \rightarrow Bxh$' is true of everything in the domain of that interpretation
- That won't work: our interpretation only assigns extensions to predicate letters, not to complex formulae like '$Sx \rightarrow Bxh$'

Quantifiers: Some False Starts

- You might instead suggest this:
  - '$\forall x(Sx \rightarrow Bxh)$' is true iff '$S\mathcal{a} \rightarrow B\mathcal{a}h$' is true, for *every* name $\mathcal{a}$ that we have included in our interpretation
- The trouble is, this doesn't work if there are some things in the domain which are not named
  - The domain of our Example Interpretation is everyone who is born before 2000CE
  - '$\forall x(Sx \rightarrow Bxh)$' clearly shouldn't be true on that interpretation
  - But '$Sg \rightarrow Bgh$' and '$Sh \rightarrow Bhh$' are both true, and 'g' and 'h' are the only names in our interpretation

Quantifiers: The Fundamental Idea

- Although ‘g’ and ‘h’ are the only names in our Example Interpretation, we **could** add another, ‘c’
- We could use ‘c’ to name any object we liked in the domain
- This leads us to a very good idea:
  - Imagine all the ways of extending our interpretation by adding the name ‘c’ and picking an object for it to refer to
  - If ‘$Sc \rightarrow Bch$’ would come out true on *all* of these different ways of extending the Example Interpretation, then ‘$\forall x(Sx \rightarrow Bxh)$’ is true on the Example Interpretation
  - ‘$Sc \rightarrow Bch$’ would come out false if ‘c’ named Lewis Capaldi, so ‘$\forall x(Sx \rightarrow Bxh)$’ is not true on the Example Interpretation

Quantifiers: Working towards the Rigorous Definition

- Suppose that $\mathcal{A}$ is a formula containing at least one instance of the variable χ, and that χ is free in $\mathcal{A}$
  - $\mathcal{A}(\ldots\chi\ldots\chi\ldots)$
- We will then write the result of replacing **every** occurrence of χ with a name c like this:
  - $\mathcal{A}(\ldots c\ldots c\ldots)$
- An example:
  - $(Fx \land Ga) \to Rxa$
  - $(Fb \land Ga) \to Rba$

Quantifiers: The Rigorous Definition

- Let c be a new name added to the language
- $\forall \chi \mathcal{A}(\ldots\chi\ldots\chi\ldots)$ is true in an interpretation iff $\mathcal{A}(\ldots c \ldots c \ldots)$ is true in *every* interpretation that extends the original interpretation by assigning an object to c (without changing the interpretation in any other way)
- $\exists \chi \mathcal{A}(\ldots\chi\ldots\chi\ldots)$ is true in an interpretation iff $\mathcal{A}(\ldots c \ldots c \ldots)$ is true in *some* interpretation that extends the original interpretation by assigning an object to c (without changing the interpretation in any other way)

Exercises!

- Consider the following interpretation:

  domain: Thor and the Hulk
  h: the Hulk
  A: Thor, the Hulk
  N: Thor
  S: $\langle$the Hulk, Thor$\rangle$

- What is the truth-value of the following sentences on this interpretation?
  1. $Ah \wedge Nh$
  2. $\forall y Ay$
  3. $\exists x(Ax \wedge Nx)$
  4. $\forall x(Shx \rightarrow Nx)$
  5. $\exists x \forall y(Sxy \leftrightarrow Ny)$
  6. $\forall x \exists y(Ax \wedge Ny)$

Exercises!

- Consider the following interpretation:

  domain: $0, 1, 2$
  a: 2
  b: 1
  F: $0, 2$
  G: 0
  R: $\langle 0, 0\rangle, \langle 1, 2\rangle, \langle 2, 0\rangle$

- What is the truth-value of the following sentences on this interpretation?
  1. Rab
  2. $\forall x(Fx \rightarrow Gx)$
  3. $\forall x \exists y Rxy$
  4. $\forall x \exists y Ryx$
  5. $\forall x \forall y(Rxy \rightarrow Fy)$
  6. $\exists x \forall y(Fy \rightarrow Ryx)$

Exercises!!!

- For each list of sentences, provide one interpretation which makes them all true, and one which makes them all false:

1. Fb, $\neg Gb$, $\exists x Gx$
2. Rab, $\exists x(Rax \land Gx)$
3. $\exists x \exists y(\neg x = y \land (Fx \land Gy))$, $\forall x(Fx \rightarrow Gx)$
4. $\neg \exists x(Fx \land Gx)$, Fa, Gb
5. Rab, $\forall x \forall y(Rxy \rightarrow Ryx)$
6. $\forall x \exists y Rxy$, $\neg \exists y \forall x Rxy$
7. Rab, $\forall x \forall y(Rxy \rightarrow Ryx)$, $\neg \exists x \exists y(\neg x = y \land (Rxy \land Ryx))$
8. Fb, $\forall y(Fy \rightarrow y = a)$
9. $\exists x(Fx \land \forall y(Fy \rightarrow y = x) \land Rxb)$