# Semantic Theory
# Week 2 – Type Theory

Noortje Venhuizen
Harm Brouwer

Universität des Saarlandes

Summer 2019

# First-order logic

First-order logic talks about:

- Individual objects
- Properties of and relations between individual objects
- Quantification over individual objects

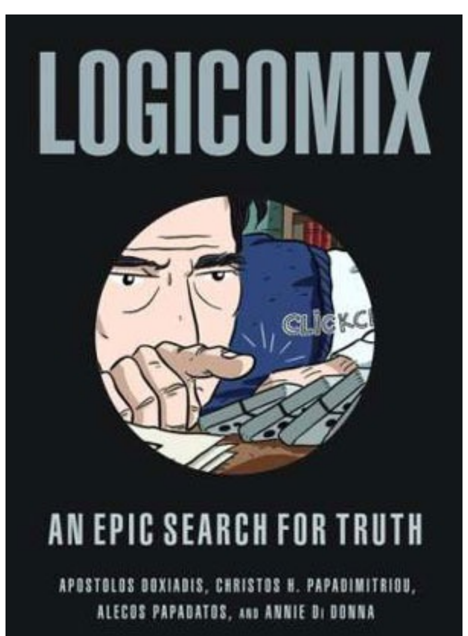
# Limitations of first-order logic

FOL is not expressive enough to capture all meanings that can be expressed by basic natural language expressions:

| | |
|---|---|
| *Jumbo is a small elephant.* | (Predicate modifiers) |
| *Happy is a state of mind.* | (Second-order predicates) |
| *Yesterday, it rained.* | (Non-logical sentence operators) |
| *Bill and John have the same hair color.* | (Higher-order quantification) |

➔ What *logically sound* system can capture this diversity?

Bertrand Russell


Q: Does the barber shave himself?

# Russell's paradox

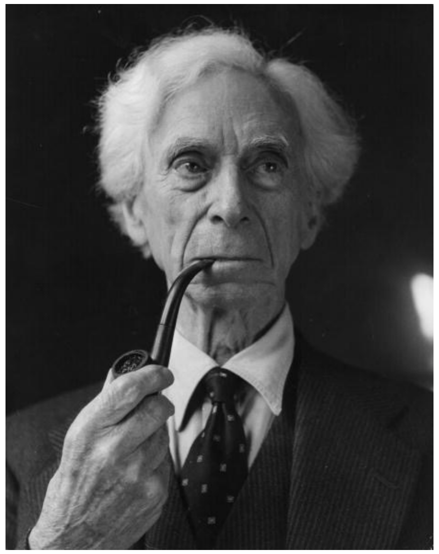
What if we extend the FOL interpretation of predicates, and interpret higher-order predicates as sets of sets of properties?

For every predicate P, we can define a set $\{x \mid P(x)\}$ containing all and only those entities for which P holds.

Then we can define a set $S = \{X \mid X \notin X\}$ representing the set of all sets that are not members of itself.

Q: does S belong to itself?

➔ We need a more restricted way of talking about *properties* and *relations between properties*!



# Type Theory

## Basic types:

- **e** – the type of individual terms (“entities”)
- **t** – the type of formulas (“truth-values”)


## Complex types:

- If **σ**, **τ** are types, then ⟨**σ, τ**⟩ is a type
(representing a functor expression that takes a **σ** type expression as argument and returns a type **τ** expression; sometimes written as: **(σ→ τ)** ).

# Types & Function Application

Types of first-order expressions:

- Individual constants (Luke, Death Star) : **e**
- One-place predicates (walk, jedi): ⟨**e, t**⟩
- Two-place predicates (fight, admire): ⟨**e,** ⟨**e, t**⟩⟩
- Three-place predicates (give, introduce): ⟨**e,** ⟨**e,** ⟨**e, t**⟩⟩⟩

Function application: Combining a functor of complex type with an appropriate argument, resulting in an expression of a less complex type: $\langle \alpha, \beta \rangle(\alpha) \mapsto \beta$

- jedi'(luke') :: ⟨**e, t**⟩(**e**) $\Longrightarrow$ **t**
- fight'(luke') :: ⟨**e,**⟨**e, t**⟩⟩(**e**) $\Longrightarrow$ ⟨**e, t**⟩

# More examples of types

Types of higher-order expressions:

- Predicate modifiers (expensive, small): ⟨⟨**e, t**⟩**,** ⟨**e, t**⟩⟩
- Second-order predicates (state of mind): ⟨⟨**e, t**⟩**, t**⟩
- Sentence operators (yesterday, possibly, unfortunately): ⟨**t, t**⟩
- Degree particles (very, too): ⟨⟨⟨**e, t**⟩**,** ⟨**e, t**⟩⟩**,** ⟨⟨**e, t**⟩**,** ⟨**e, t**⟩⟩⟩

Tip: If **σ**, **τ** are basic types, ⟨**σ, τ**⟩ can be abbreviated as **στ**. Thus, the type of predicate modifiers and second-order predicates can be more conveniently written as ⟨**et, et**⟩ and ⟨**et, t**⟩, respectively.

# Type Theory — Vocabulary

## Non-logical constants:

- For every type $\tau$ a (possibly empty) set of non-logical constants $CON_\tau$ (pairwise disjoint)

## Variables:

- For every type $\tau$ an infinite set of variables $VAR_\tau$ (pairwise disjoint)

## Logical symbols: $\forall$, $\exists$, $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $=$

## Brackets: (, )

# Type Theory — Syntax

For every type $\tau$, the set of well-formed expressions $WE_\tau$ is defined as follows:

(i) $CON_\tau \subseteq WE_\tau$ and $VAR_\tau \subseteq WE_\tau$;

(ii) If $\alpha \in WE_{\langle\sigma, \tau\rangle}$, and $\beta \in WE_\sigma$, then $\alpha(\beta) \in WE_\tau$; (function application)

(iii) If A, B are in $WE_t$, then $\neg A$, $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$, $(A \leftrightarrow B)$ are in $WE_t$;

(iv) If A is in $WE_t$ and x is a variable of arbitrary type, then $\forall xA$ and $\exists xA$ are in $WE_t$;

(v) If $\alpha$, $\beta$ are well-formed expressions of the same type, then $\alpha = \beta \in WE_t$;

(vi) Nothing else is a well-formed expression.

# Function application

*(ii) If* $\alpha \in WE_{\langle\sigma,\tau\rangle}$*, and* $\beta \in WE_{\sigma}$*, then* $\alpha(\beta) \in WE_{\tau}$

"Luke is a talented jedi"

$$\dfrac{\text{luke'} :: e \qquad \dfrac{\text{jedi'} :: \langle e, t\rangle \qquad \text{talented'} :: \langle\langle e, t\rangle, \langle e, t\rangle\rangle}{\text{talented'(jedi')} :: \langle e, t\rangle}}{\text{talented'(jedi')(luke')} :: t}$$

# Type inferencing: examples

1. Yoda$_e$ [is faster than Palpatine$_e$].

2. Yoda$_e$ [is much [faster than]] Palpatine$_e$.

3. [[Han Solo]$_e$ fights] [because [[the Dark Side]$_e$ is rising]].

4. Obi-Wan$_e$ [[told [Qui-Gon Jinn]$_e$] he will take [the Jedi-exam]$_e$].

# Higher-order predicates

## Higher-order quantification:

- *Leia has the same hair colour as Padmé*

$$\exists C\ (\text{hair\_colour}(C) \wedge C(l') \wedge C(p'))$$

$\langle\langle e, t\rangle, t\rangle$ $\langle e, t\rangle$ $e$

## Higher-order equality:

- For p, q $\in CON_t$, "p=q" expresses material equivalence: "p $\leftrightarrow$ q".
- For F, G $\in CON_{\langle e, t\rangle}$, "F=G" expresses co-extensionality: "$\forall x(Fx \leftrightarrow Gx)$"
- For any formula $\phi$ of type *t*, $\phi=(x=x)$ is a representation of "$\phi$ is true".

# Type Theory — Semantics [1]

Let **U** be a non-empty set of entities.

The domain of possible denotations $\mathbf{D_\tau}$ for every type $\boldsymbol{\tau}$ is given by:

- $D_e = U$
- $D_t = \{0,1\}$
- $D_{\langle\sigma,\tau\rangle}$ is the set of all functions from $D_\sigma$ to $D_\tau$

*Expressions of type* $\boldsymbol{\sigma}$ *denote elements of* $\mathbf{D_\sigma}$

# Characteristic functions

Many natural language expressions have a type ⟨**σ, t**⟩

Expressions with type ⟨**σ, t**⟩ are functions mapping elements of type **σ** to truth values: {**0,1**}

Such functions with a range of {**0,1**} are called *characteristic functions*, because they uniquely specify a subset of their domain $\mathbf{D_\sigma}$

The characteristic function of set M in a domain U is the function $F_M$: U→{0,1} such that for all a ∈ U, $F_M$(a) = 1 iff a ∈ M.

NB: For first-order predicates, the FOL representation (using sets) and the type-theoretic representation (using characteristic functions) are equivalent.

# Interpretation with characteristic functions: example

For M = ⟨U, V⟩, let U consist of five entities. For selected types, we have the following sets of possible denotations:

- $D_t = \{0,1\}$
- $D_e = U = \{e_1, e_2, e_3, e_4, e_5\}$
- $D_{<e,t>} = \left\{ \begin{bmatrix} e_1 \rightarrow 1 \\ e_2 \rightarrow 0 \\ e_3 \rightarrow 1 \\ e_4 \rightarrow 0 \\ e_5 \rightarrow 1 \end{bmatrix}, \begin{bmatrix} e_1 \rightarrow 1 \\ e_2 \rightarrow 1 \\ e_3 \rightarrow 0 \\ e_4 \rightarrow 1 \\ e_5 \rightarrow 1 \end{bmatrix}, \begin{bmatrix} e_1 \rightarrow 0 \\ e_2 \rightarrow 1 \\ e_3 \rightarrow 1 \\ e_4 \rightarrow 0 \\ e_5 \rightarrow 0 \end{bmatrix}, \ldots \right\}$

Alternative set notation: $D_{<e,t>} = \{\{e_1,e_3,e_5\},\{e_1,e_2,e_4,e_5\},\{e_2,e_3\},\ldots\}$

# Type Theory — Semantics [2]

A model structure for a type theoretic language is a tuple $\mathbf{M} = \langle \mathbf{U}, \mathbf{V} \rangle$ such that:

- **U** is a non-empty domain of individuals
- **V** is an interpretation function, which assigns to every $\boldsymbol{\alpha} \in \mathbf{CON}_{\boldsymbol{\tau}}$ an element of $\mathbf{D}_{\boldsymbol{\tau}}$ (where $\boldsymbol{\tau}$ is an arbitrary type)

The variable assignment function g assigns to every typed variable $\mathbf{v} \in \mathbf{VAR}_{\boldsymbol{\tau}}$ an element of $\mathbf{D}_{\boldsymbol{\tau}}$

# Type Theory — Interpretation

Given a model structure M = ⟨U, V⟩ and a variable assignment g:

$[\![\alpha]\!]^{M,g}$ = V(α) if α is a constant
= g(α) if α is a variable

$[\![\alpha(\beta)]\!]^{M,g} = [\![\alpha]\!]^{M,g}([\![\beta]\!]^{M,g})$

$[\![\alpha = \beta]\!]^{M,g}$ = 1 iff $[\![\alpha]\!]^{M,g} = [\![\beta]\!]^{M,g}$

$[\![\neg\phi]\!]^{M,g}$ = 1 iff $[\![\phi]\!]^{M,g} = 0$

$[\![\phi \wedge \psi]\!]^{M,g}$ = 1 iff $[\![\phi]\!]^{M,g} = 1$ and $[\![\psi]\!]^{M,g} = 1$

$[\![\phi \vee \psi]\!]^{M,g}$ = 1 iff $[\![\phi]\!]^{M,g} = 1$ or $[\![\psi]\!]^{M,g} = 1$

…

*For any variable v of type σ:*

$[\![\exists v\phi]\!]^{M,g}$ = 1 iff there is a d ∈ $D_\sigma$ such that $[\![\phi]\!]^{M,g[v/d]} = 1$

$[\![\forall v\phi]\!]^{M,g}$ = 1 iff for all d ∈ $D_\sigma$ : $[\![\phi]\!]^{M,g[v/d]} = 1$

# Interpretation: Example

*Luke is a talented jedi*

$$\frac{\dfrac{}{\text{luke'} :: e} \qquad \dfrac{\text{jedi'} :: \langle e, t\rangle \qquad \text{talented'} :: \langle\langle e, t\rangle, \langle e, t\rangle\rangle}{\text{talented'(jedi')} :: \langle e, t\rangle}}{\text{talented'(jedi')(luke')} :: t}$$

$[\![\text{talented'(jedi')(luke')}]\!]^{M,g}$

$= [\![\text{talented'(jedi')}]\!]^{M,g} ([\![\text{luke'}]\!]^{M,g})$

$= [\![\text{talented'}]\!]^{M,g}([\![\text{jedi'}]\!]^{M,g}) ([\![\text{luke'}]\!]^{M,g})$

$= V_M(\text{talented'})(V_M(\text{jedi'}))(V_M(\text{luke'}))$

# Interpretation: Example (cont.)

$[\![\textit{Luke is a talented jedi}]\!]^{M,g} = V_M(\text{talented'})(V_M(\text{jedi'}))(V_M(\text{luke'}))$

$V_M(\text{luke'}) = e_1 \ (\in \mathbf{D_e})$

$$V_M(\text{jedi'}) = \begin{bmatrix} e_1 \to 1 \\ e_2 \to 0 \\ e_3 \to 1 \\ e_4 \to 0 \\ e_5 \to 1 \end{bmatrix} (\in \mathbf{D}_{\langle \mathbf{e,t} \rangle}) \iff \{e_1, e_3, e_5\}$$

$$V_M(\text{talented'}) = \begin{bmatrix} \begin{bmatrix} e_1 \to 1 \\ e_2 \to 0 \\ e_3 \to 1 \\ e_4 \to 0 \\ e_5 \to 1 \end{bmatrix} \to \begin{bmatrix} e_1 \to 1 \\ e_2 \to 0 \\ e_3 \to 0 \\ e_4 \to 0 \\ e_5 \to 1 \end{bmatrix} \\ \begin{bmatrix} e_1 \to 0 \\ e_2 \to 0 \\ e_3 \to 1 \\ e_4 \to 1 \\ e_5 \to 1 \end{bmatrix} \to \begin{bmatrix} e_1 \to 0 \\ e_2 \to 0 \\ e_3 \to 0 \\ e_4 \to 0 \\ e_5 \to 1 \end{bmatrix} \\ \dots \end{bmatrix} (\in \mathbf{D}_{\langle \langle \mathbf{e,t} \rangle \langle \mathbf{e,t} \rangle \rangle}) \iff \begin{bmatrix} \{e_1, e_3, e_5\} \to \{e_1, e_5\} \\ \{e_3, e_4, e_5\} \to \{e_5\} \\ \dots \end{bmatrix}$$

$V_M(\text{talented'})(V_M(\text{jedi'}))$

$V_M(\text{talented'})(V_M(\text{jedi'}))(V_M(\text{luke'}))$

# Background reading material

- Gamut: Logic, Language, and Meaning Vol II (Chapter 4)
- Winter: Elements of Formal Semantics (Chapter 3)
  http://www.phil.uu.nl/~yoad/efs/main.html