NATIONAL RESEARCH UNIVERSITY

The «Ontological Square» and Modern Type Theories

«Logic Today: Developments and Perspectives»,

St.Petersburg State University, May 31, 2018

Vitaliy Dolgorukov

vdolgorukov@hse.ru

International Laboratory for Logic, Linguistics and Formal Philosophy (IL LLFP) / School of Philosophy

National Research University Higher School of Economics

Logical Form?

1. *John is a man.*
2. *John is happy.*

Ontology & Semantics

- Ontology: the Ontological Square

Ontology & Semantics

- Ontology: the Ontological Square
- Semantics: Modern Type Theories

Ontological Square

Ontological Square

- (Angelelli 1967)

Ontological Square

- (Angelelli 1967)
- Aristotle, *'Categoriae', la, 20-lb,10*

Ontological Square

- (Angelelli 1967)
- Aristotle, *'Categoriae', la, 20-lb,10*

Ontological Square

- (Angelelli 1967)
- Aristotle, *'Categoriae', la, 20-lb,10*
-
  - Universal Things vs. Singular Things

Ontological Square

- (Angelelli 1967)
- Aristotle, *'Categoriae', la, 20-lb,10*
-
    - Universal Things vs. Singular Things
    - Substances vs. Accidents

Ontological Square

1. Universal Substances = universal essential things	*3. Universal Accidents = universal accidental things*
2. Individual Substances = singular essential things	*4. Individual Accidents = singular accidental things*

Table: The Aristotle's Ontological Square

Ontological Square

1. Universal Substances = universal essential things e.g. 'Man'	*3. Universal Accidents = universal accidental things e.g. 'Wisdom'*
2. Individual Substances = singular essential things e.g. 'Socrates'	*4. Individual Accidents = singular accidental things e.g. 'Socrates's Wisdom'*

Table: The Aristotle's Ontological Square

Criteria

Criteria

- Substances vs. Accidents
- P – Universal Substance: if x is P, then x is P at every time at which x exists
- Universal vs. Individual
- An individual object is a unique object

Againts «Fantology»

> *«A dark force haunts much of what is most admirable in the philosophy of the last one hundred years. It consists, briefly put, in the doctrine to the effect that one can arrive at a correct ontology by paying attention to certain superficial (syntactic) features of first-order predicate logic as conceived by Frege and Russell. More specifically, fantology is a doctrine to the effect that the key to the ontological structure of reality is captured syntactically in the 'Fa' (or, in more sophisticated versions, in the 'Rab') of first-order logic, where 'F' stands for what is general in reality and 'a' for what is individual».*

(Smith 2005, 153)

Againts «Fantology»

> *«..Frege's object/function distinction rides roughshod over two traditional ontological distinctions, between substance and property and between particular and universal».*

(Smith 2005, 163)

Frege's reduction of OS

1. Universal Substances *?*	*3. Universal Accidents* *OK*
2. Individual Substances *OK*	*4. Individual Accidents* *?*

Smith's Solution

Smith's Solution

- $= (x, y)$, for: x is identical to y

Smith's Solution

- $=(x, y)$, for: x is identical to y
- $Part(x, y)$, for: individual x is part of individual y

Smith's Solution

- $=(x, y)$, for: x is identical to y
- $Part(x, y)$, for: individual x is part of individual y
- $Inst(x, y)$, for: individual x instantiates universal y

Smith's Solution

- $=(x, y)$, for: x is identical to y
- $Part(x, y)$, for: individual x is part of individual y
- $Inst(x, y)$, for: individual x instantiates universal y
- $Inhere(x, y)$, for: individual x inheres in individual y

Smith's Solution

- $=(x, y)$, for: x is identical to y
- $Part(x, y)$, for: individual x is part of individual y
- $Inst(x, y)$, for: individual x instantiates universal y
- $Inhere(x, y)$, for: individual x inheres in individual y
- $Exemp(x, y)$, for: individual x exemplifies property y

Smith's Solution

- $=(x, y)$, for: x is identical to y
- $Part(x, y)$, for: individual x is part of individual y
- $Inst(x, y)$, for: individual x instantiates universal y
- $Inhere(x, y)$, for: individual x inheres in individual y
- $Exemp(x, y)$, for: individual x exemplifies property y
- $Dep(x, y)$, for: individual x depends for its existence on individual y

Smith's Solution

- $=(x, y)$, for: x is identical to y
- $Part(x, y)$, for: individual x is part of individual y
- $Inst(x, y)$, for: individual x instantiates universal y
- $Inhere(x, y)$, for: individual x inheres in individual y
- $Exemp(x, y)$, for: individual x exemplifies property y
- $Dep(x, y)$, for: individual x depends for its existence on individual y
- $Is\_a(x, y)$, for: universal x is a subkind of universal y

Smith's Solution

- $=(x,y)$, for: x is identical to y
- $Part(x,y)$, for: individual x is part of individual y
- $Inst(x,y)$, for: individual x instantiates universal y
- $Inhere(x,y)$, for: individual x inheres in individual y
- $Exemp(x,y)$, for: individual x exemplifies property y
- $Dep(x,y)$, for: individual x depends for its existence on individual y
- $Is\_a(x,y)$, for: universal x is a subkind of universal y
- $Precedes(x,y)$, for: individual process x precedes individual process y

Smith's Solution

- $=(x,y)$, for: x is identical to y
- $Part(x,y)$, for: individual x is part of individual y
- $Inst(x,y)$, for: individual x instantiates universal y
- $Inhere(x,y)$, for: individual x inheres in individual y
- $Exemp(x,y)$, for: individual x exemplifies property y
- $Dep(x,y)$, for: individual x depends for its existence on individual y
- $Is\_a(x,y)$, for: universal x is a subkind of universal y
- $Precedes(x,y)$, for: individual process x precedes individual process y
- $Has\_Participant(x,y)$, for: individual thing y participates in individual occurrent x

Smith's Solution

- $=(x, y)$, for: x is identical to y
- $Part(x, y)$, for: individual x is part of individual y
- $Inst(x, y)$, for: individual x instantiates universal y
- $Inhere(x, y)$, for: individual x inheres in individual y
- $Exemp(x, y)$, for: individual x exemplifies property y
- $Dep(x, y)$, for: individual x depends for its existence on individual y
- $Is\_a(x, y)$, for: universal x is a subkind of universal y
- $Precedes(x, y)$, for: individual process x precedes individual process y
- $Has\_Participant(x, y)$, for: individual thing y participates in individual occurrent x
- $Has\_Agent(x, y)$, for: individual thing y is agent of individual occurrent x

Smith's Solution

- $=(x, y)$, for: x is identical to y
- $Part(x, y)$, for: individual x is part of individual y
- $Inst(x, y)$, for: individual x instantiates universal y
- $Inhere(x, y)$, for: individual x inheres in individual y
- $Exemp(x, y)$, for: individual x exemplifies property y
- $Dep(x, y)$, for: individual x depends for its existence on individual y
- $Is\_a(x, y)$, for: universal x is a subkind of universal y
- $Precedes(x, y)$, for: individual process x precedes individual process y
- $Has\_Participant(x, y)$, for: individual thing y participates in individual occurrent x
- $Has\_Agent(x, y)$, for: individual thing y is agent of individual occurrent x
- $Realizes(x, y)$, for: individual process x realizes individual function y

Smith's Solution

- $Realizes(x, y) \rightarrow \exists z(Dep(x, y) \wedge Dep(y, z))$
- $Exemp(x, y) \rightarrow \exists z(Inst(z, y) \wedge Inhere(z, x))$

Problems of Smith's Solution

- set of predicates
- non-compositional

Another solution: MTTs

- Types as Manageable Sets
- MTTs and Montague Grammar

Types as Manageable Sets

- $a \in A$
- $a : A$
- $a : A$ is decidable

MTTs

Martin- Löf's type theory (Martin-Löf (Martin-Löf 1984), propositions-as-types principle

MTTs vs. Montague Grammar

MTTs vs. Montague Grammar

- basic types in MG: e, t
  (in some extensions of MG: s, v, etc.)

MTTs vs. Montague Grammar

- basic types in MG: e, t
  (in some extensions of MG: s, v, etc.)
- basic types in MTT: $[[Man]]$, $[[Animal]]$, etc.
  CNs are types

MTTs vs. Montague Grammar

- basic types in MG: e, t
  (in some extensions of MG: s, v, etc.)
- basic types in MTT: $[[Man]]$, $[[Animal]]$, etc.
  CNs are types
- type formation operation in MG: $Type \to Type$

MTTs vs. Montague Grammar

- basic types in MG: e, t
  (in some extensions of MG: s, v, etc.)
- basic types in MTT: $[[Man]]$, $[[Animal]]$, etc.
  CNs are types
- type formation operation in MG: $Type \rightarrow Type$
- type formation operations in MTT

MTTs vs. Montague Grammar

- basic types in MG: e, t
  (in some extensions of MG: s, v, etc.)
- basic types in MTT: $[[Man]]$, $[[Animal]]$, etc.
  CNs are types
- type formation operation in MG: $Type \rightarrow Type$
- type formation operations in MTT
  - $Type \rightarrow Type$

MTTs vs. Montague Grammar

- basic types in MG: e, t
  (in some extensions of MG: s, v, etc.)
- basic types in MTT: $[[Man]]$, $[[Animal]]$, etc.
  CNs are types
- type formation operation in MG: $Type \rightarrow Type$
- type formation operations in MTT
  - $Type \rightarrow Type$
  - $Type \times Type$

MTTs vs. Montague Grammar

- basic types in MG: e, t
  (in some extensions of MG: s, v, etc.)
- basic types in MTT: $[[Man]]$, $[[Animal]]$, etc.
  CNs are types
- type formation operation in MG: $Type \to Type$
- type formation operations in MTT
  - $Type \to Type$
  - $Type \times Type$
  - $\Sigma(Type, Type)$ //or $\Sigma x : Type.Type(x)$//

MTTs vs. Montague Grammar

- basic types in MG: e, t
  (in some extensions of MG: s, v, etc.)
- basic types in MTT: $[[Man]]$, $[[Animal]]$, etc.
  CNs are types
- type formation operation in MG: $Type \to Type$
- type formation operations in MTT
  - $Type \to Type$
  - $Type \times Type$
  - $\Sigma(Type, Type)$ //or $\Sigma x\!:\!Type.Type(x)$//
  - $\Pi(Type, Type)$ //or $\Pi x\!:\!Type.Type(x)$//

MTTs vs. Montague Grammar

- basic types in MG: e, t
  (in some extensions of MG: s, v, etc.)
- basic types in MTT: $[[Man]]$, $[[Animal]]$, etc.
  CNs are types
- type formation operation in MG: $Type \rightarrow Type$
- type formation operations in MTT
  - $Type \rightarrow Type$
  - $Type \times Type$
  - $\Sigma(Type, Type)$ //or $\Sigma x : Type.Type(x)$//
  - $\Pi(Type, Type)$ //or $\Pi x : Type.Type(x)$//
- MTT: coercive subtyping ($Type_1 \leq Type_2$)

Dependent Sum-Type

Dependent Sum-Type

- $\Sigma x : A . B(x)$ $//\Sigma(A, B)//$

Dependent Sum-Type

- $\Sigma x : A . B(x)$ $//\Sigma(A, B)//$
- dependent extension of $A \times B$

Dependent Sum-Type

- $\Sigma x\!:\!A.B(x)$ $//\Sigma(A,B)//$
- dependent extension of $A \times B$
- $(a,b) : \Sigma x\!:\!A.B(x)$ is a type of a pair $a : A$ and $b : B(a)$

Dependent Sum-Type

- $\Sigma x\!:\!A.B(x)$ $//\Sigma(A,B)//$
- dependent extension of $A \times B$
- $(a,b) : \Sigma x\!:\!A.B(x)$ is a type of a pair $a : A$ and $b : B(a)$
- type of pairs of natural numbers s.t $a \leq b$:
  $\Sigma x\!:\!\mathbb{N}.\lambda n.a + n = b$

Dependent Product-Type

Dependent Product-Type

- $\Pi x\!:\!A.B(x)$ // $\Pi(A, B)$//

Dependent Product-Type

- $\Pi x : A.B(x)$ // $\Pi(A, B)$//
- dependent extension of $A \to B$

Dependent Product-Type

- $\Pi x\!:\!A.B(x)$ // $\Pi(A, B)$//
- dependent extension of $A \to B$
- $(a, b) : \Pi x\!:\!A.B(x)$ is a type of dependent functions f on A so that $f(a)$ has type $B(a)$ for $a\!: A$

Dependent Product-Type

- $\Pi x : A . B(x)$ // $\Pi(A, B)$//
- dependent extension of $A \to B$
- $(a, b) : \Pi x : A . B(x)$ is a type of dependent functions f on A so that $f(a)$ has type $B(a)$ for $a : A$
- type of functions which return the list consisting of natural numbers from x down to 0
  $\Pi x : \mathbb{N} . List(x)$

MG vs. MTTs

MG vs. MTTs

- CN: man, human

MG vs. MTTs

- CN: man, human
  - $man', human' : e \rightarrow t$

MG vs. MTTs

- CN: man, human
  - $man', human' : e \to t$
  - $[[Man]], [[Human]] : Type$

MG vs. MTTs

- CN: man, human
  - $man', human' : e \to t$
  - $[[Man]], [[Human]] : Type$
- IV: talk

MG vs. MTTs

- CN: man, human
  - $man', human' : e \rightarrow t$
  - $[[Man]], [[Human]] : Type$
- IV: talk
  - $talk' : e \rightarrow t$

MG vs. MTTs

- CN: man, human
  - $man', human' : e \to t$
  - $[[Man]], [[Human]] : Type$
- IV: talk
  - $talk' : e \to t$
  - $[[talk]] : [[Human]] \to Prop$

MG vs. MTTs

- CN: man, human
  - $man', human' : e \to t$
  - $[[Man]], [[Human]] : Type$
- IV: talk
  - $talk' : e \to t$
  - $[[talk]] : [[Human]] \to Prop$
- Adj: man, handsome

MG vs. MTTs

- CN: man, human
  - $man', human' : e \to t$
  - $[[Man]], [[Human]] : Type$
- IV: talk
  - $talk' : e \to t$
  - $[[talk]] : [[Human]] \to Prop$
- Adj: man, handsome
  - $handsome' : (e \to t) \to (e \to t)$

MG vs. MTTs

- CN: man, human
  - $man', human' : e \to t$
  - $[[Man]], [[Human]] : Type$
- IV: talk
  - $talk' : e \to t$
  - $[[talk]] : [[Human]] \to Prop$
- Adj: man, handsome
  - $handsome' : (e \to t) \to (e \to t)$
  - $[[handsome]] : [[Man]] \to Prop$

MG vs. MTTs

- CN: man, human
  - $man', human' : e \rightarrow t$
  - $[[Man]], [[Human]] : Type$
- IV: talk
  - $talk' : e \rightarrow t$
  - $[[talk]] : [[Human]] \rightarrow Prop$
- Adj: man, handsome
  - $handsome' : (e \rightarrow t) \rightarrow (e \rightarrow t)$
  - $[[handsome]] : [[Man]] \rightarrow Prop$
- MCN: handsome man

MG vs. MTTs

- CN: man, human
  - $man', human' : e \to t$
  - $[[Man]], [[Human]] : Type$
- IV: talk
  - $talk' : e \to t$
  - $[[talk]] : [[Human]] \to Prop$
- Adj: man, handsome
  - $handsome' : (e \to t) \to (e \to t)$
  - $[[handsome]] : [[Man]] \to Prop$
- MCN: handsome man
  - $handsome'(man') : (e \to t)$

MG vs. MTTs

- CN: man, human
  - $man', human' : e \to t$
  - $[[Man]], [[Human]] : Type$
- IV: talk
  - $talk' : e \to t$
  - $[[talk]] : [[Human]] \to Prop$
- Adj: man, handsome
  - $handsome' : (e \to t) \to (e \to t)$
  - $[[handsome]] : [[Man]] \to Prop$
- MCN: handsome man
  - $handsome'(man') : (e \to t)$
  - $\Sigma x : [[Man]].[[handsome]](x) : Type$

MG vs. MTTs

- CN: man, human
  - $man', human' : e \to t$
  - $[[Man]], [[Human]] : Type$
- IV: talk
  - $talk' : e \to t$
  - $[[talk]] : [[Human]] \to Prop$
- Adj: man, handsome
  - $handsome' : (e \to t) \to (e \to t)$
  - $[[handsome]] : [[Man]] \to Prop$
- MCN: handsome man
  - $handsome'(man') : (e \to t)$
  - $\Sigma x : [[Man]].[[handsome]](x) : Type$
- TP: A man talks

MG vs. MTTs

- CN: man, human
  - $man', human' : e \to t$
  - $[[Man]], [[Human]] : Type$
- IV: talk
  - $talk' : e \to t$
  - $[[talk]] : [[Human]] \to Prop$
- Adj: man, handsome
  - $handsome' : (e \to t) \to (e \to t)$
  - $[[handsome]] : [[Man]] \to Prop$
- MCN: handsome man
  - $handsome'(man') : (e \to t)$
  - $\Sigma x : [[Man]].[[handsome]](x) : Type$
- TP: A man talks
  - $\exists x : e\ [man'(x) \wedge talk'(x)] : t$

MG vs. MTTs

- CN: man, human
  - $man', human' : e \to t$
  - $[[Man]], [[Human]] : Type$
- IV: talk
  - $talk' : e \to t$
  - $[[talk]] : [[Human]] \to Prop$
- Adj: man, handsome
  - $handsome' : (e \to t) \to (e \to t)$
  - $[[handsome]] : [[Man]] \to Prop$
- MCN: handsome man
  - $handsome'(man') : (e \to t)$
  - $\Sigma x : [[Man]].[[handsome]](x) : Type$
- TP: A man talks
  - $\exists x : e\ [man'(x) \wedge talk'(x)] : t$
  - $\exists x : [[Man]].[[talk]](x) : Prop$

John is a man vs. *John is happy*

John is a man vs. *John is happy*

	MG	MTTs
John is a man	$man'(j) : t$	$j : [[Man]] : Prop$
John is happy	$happy'(j) : t$	$(j, p) : \Sigma x : [[Man]].[[happy]](x) : Prop$

Problems and open questions

Problems and open questions

- $||John\ is\ happy|| = ||John\ is\ a\ happy\ man||$

Problems and open questions

- $||John\ is\ happy|| = ||John\ is\ a\ happy\ man||$
- Negation

Problems and open questions

- $||John\ is\ happy|| = ||John\ is\ a\ happy\ man||$
- Negation
  - *John is not a dog.*

Problems and open questions

- $||John\ is\ happy|| = ||John\ is\ a\ happy\ man||$
- Negation
  - *John is not a dog.*
  - $\nvdash j : [[Dog]] : Prop$

Problems and open questions

- $||John\ is\ happy|| = ||John\ is\ a\ happy\ man||$
- Negation
  - *John is not a dog.*
  - $\nvdash j : [[Dog]] : Prop$
  - (Chatzikyriakidis & Luo 2017):
    $NOT : \Pi A : CN.(A \to Prop) \to (Obj \to Prop)$

Problems and open questions

- $||John\ is\ happy|| = ||John\ is\ a\ happy\ man||$
- Negation
  - *John is not a dog.*
  - $\nvdash j : [[Dog]] : Prop$
  - (Chatzikyriakidis & Luo 2017):
    $NOT : \Pi A : CN.(A \to Prop) \to (Obj \to Prop)$
- Individual accidents?

References

- Angelelli I. *Studies on Gottlob Frege and Traditional Philosophy.* Dordrecht: Reidel, 1967.
- Luo Z. Formal Semantics in Modern Type Theories with Coercive Subtyping. *Linguistics and Philosophy* 35(6), 2012, pp. 491–513.
- Martin-Löf. P. *Intuitionistic Type Theory.* Napoli: Bibliopolis, 1984.
- *Modern Perspectives in Type-Theoretical Semantics.* Eds. by Stergios Chatzikyriakidis and Zhaohui Luo. Dordrecht: Springer, 2017.
- Ranta A. *Type-Theoretical Grammar.* Oxford: Oxford University Press, 1994.
- Smith B. *Against Fantology* in *Experience and Analysis*. Eds. by Johann C. Marek and Maria E. Reicher. Vienna: Öbv&Hpt, 2005, pp. 153–170.
- Schneider L. The Logic of the Ontological Square. *Studia Logica.* 91(1), 2009, pp. 25–51.