Zalta's Elementary Object Theory

Qilei Xing

June 12, 2018

Department of Philosophy, PKU

Outline

Outline

These slides are based on *Abstact Objects*, D.Reidel Publishing Company, 1983

Existent or non-existent, that's the question.

Outline

Outline

Primitive Symbols

1. Primitive object terms
   1.1 Constants: $a_1, a_2, a_3 \ldots$.[1]
   1.2 Variables: $x_1, x_2, x_3 \ldots$.[2]
2. Primitive relation terms
   2.1 Constants: $P^n_1, P^n_2, P^n_3 \ldots, (n \geq 1),$[3] $=_E, E!$.
   2.2 Variables: $F^n_1, F^n_2, F^n_3, \ldots, (n \geq 1)$.[4]
3. Connectives: $\neg, \rightarrow$.
4. Quantifier: $\forall$.
5. Lambda: λ.
6. Parentheses and brackets: $(,), [,]$.

Formulas and Terms

We present a simultaneous inductive definition of **(propositional) formula**, **object term**, and ***n*-place relation term**

1. All primitive object terms are object terms and all primitive n-place relation terms are n-place relation terms.
2. Atomic exemplification: If ρ^n is any n-place relation term, and $o_1, \ldots, o_n$ are any object terms, $\rho^n o_1 \ldots o_n$ is a (propositional) formula.
3. Atomic encoding: If ρ^1 is any one-place relation term, $o\rho^1$ is a formula.
4. Molecular: If ϕ and ψ are any (propositional) formulas, then $(\neg\phi)$ and $(\phi \rightarrow \psi)$ are (propositional) formulas.

Formulas and Terms (cont.)

5. Quantified: If ϕ is any (propositional) formula, and α is any (object) variable, then $(\forall\alpha)\phi$ is a (propositional) formula.
6. Complex n-place relation terms: If ϕ is any propositional formula with n-free object variables $v_1, \ldots, v_n$, then $[\lambda v_1 \ldots v_n \phi]$ is an n-place relation term.

Some Notations and Definitions

- $(A!x)$ x is **abstract** $=_{df} [\lambda y \neg E!y]x$
- A formula ϕ is **propositional** iff ϕ has no encoding subformulas and ϕ has no subformulas with quantifiers binding relation variables.
- Rewrite $=_E o_1 o_2$ as $o_1 =_E o_2$
- parentheses
- $\wedge, \vee, \leftrightarrow, \exists$

Some Notations and Definitions (cont.)

- τ is a **term** iff τ is an object term or there is an *n* such that τ is an *n*-place relation term.
- All and only formulas and terms are **well-formed expressions**.
- An occurrence of a variable α in a well-formed expression is **bound (free)** iff it lies (does not lie) with a formula of the form $(\forall\alpha)\phi$ or a term of the form $[\lambda v_1 \ldots \alpha \ldots v_n \phi]$ within the expression.
- A variable is **free (bound)** in an expression iff it does (does not) have a free occurrence in the expression.
- A sentence is a formula having no free variables.

Some Notations and Definitions (cont.)

- A term is **substitutable for** a variable α **in** a formula ϕ iff for every variable β free in τ, no free occurrence of α in ϕ occurs either in a subformula of the form $(\forall\beta)\psi$ in ϕ or in a term $[\lambda v_1 \ldots \beta \ldots v_n \psi]$ in ϕ.
- We write $\phi(\alpha_1, \ldots, \alpha_n)$ to designate a formula which may or may not have $\alpha_1, \ldots, \alpha_n$ occuring free.
- We write $\phi^{\tau_1,\ldots,\tau_n}_{\alpha_1,\ldots,\alpha_n}$ to designate the formula which results when, for each $i, 1 \leq i \leq n, \tau_i$ is substituted for each free occurrence of α_i in ϕ.

Outline

Interpretations

An interpretation, of our language is any 6-tuple $\langle \mathscr{D}, \mathscr{R}, ext_{\mathscr{R}}, \mathscr{L}, ext_{\mathscr{A}}, F \rangle$.

- $\mathscr{D}$ is an non-empty set. It is called the **domain of objects**. ($o \in \mathscr{D}$)
- $\mathscr{R}$ is an non-empty set. It is called the **domain of relations** and it is the union of a sequence of non-empty set $\mathscr{R}_1, \mathscr{R}_2, \mathscr{R}_3, \ldots$; i.e, $\mathscr{R} = \bigcup_{n \geq 1} \mathscr{R}_n$. ($\imath^n \in \mathscr{R}_n$)
- $ext_{\mathscr{R}} : \mathscr{R}_n \to \mathcal{P}(\mathscr{D}^n)$. We call $ext_{\mathscr{R}}(\imath^n)$ the **exemplification extension** of $\imath^n$.

Interpretations (cont.)

- $\mathscr{L}$ is a class of logical functions which operate on the members of $\mathscr{R}^n$ and $\mathscr{D}$ to produce the complex relations which serve as the denotations for the λ-expressions.

1. $\mathscr{PLUG}_1$ maps $(\mathscr{R}_2 \cup \mathscr{R}_3 \cup \ldots) \times \mathscr{D}$ into $(\mathscr{R}_1 \cup \mathscr{R}_2 \cup \ldots)$. $\mathscr{PLUG}_j$, for each $j > 1$, maps $(\mathscr{R}_j \cup \mathscr{R}_{j+1} \cup \ldots) \times \mathscr{D}$ into $(\mathscr{R}_{j-1} \cup \mathscr{R}_j \cup \ldots)$.
   $\mathscr{PLUG}_i$ is subject to the following conditions:
   $ext_{\mathscr{R}}(\mathscr{PLUG}_i(\imath^n, \mathbf{o})) = \{\langle \mathbf{o}_1, \ldots, \mathbf{o}_{i-1}, \mathbf{o}_{i+1}, \ldots, \mathbf{o}_n \rangle \mid \langle \mathbf{o}_1, \ldots, \mathbf{o}_{i-1}, \mathbf{o}, \mathbf{o}_{i+1}, \ldots, \mathbf{o}_n \rangle \in ext_{\mathscr{R}}(\imath^n)\}$
2. $\mathscr{UNIV}_1$ maps $(\mathscr{R}_2 \cup \mathscr{R}_3 \cup \ldots)$ into $(\mathscr{R}_1 \cup \mathscr{R}_2 \cup \ldots)$. $\mathscr{UNIV}_j$, for each $j > 1$, maps $(\mathscr{R}_j \cup \mathscr{R}_{j+1} \cup \ldots)$ into $(\mathscr{R}_{j-1} \cup \mathscr{R}_j \cup \ldots)$. $\mathscr{UNIV}_i$ is subject to the condition:
   $ext_{\mathscr{R}}(\mathscr{UNIV}_i(\imath^n)) = \{\langle \mathbf{o}_1, \ldots, \mathbf{o}_{i-1}, \mathbf{o}_{i+1}, \ldots, \mathbf{o}_n \rangle \mid (\forall \mathbf{o})(\langle \mathbf{o}_1, \ldots, \mathbf{o}_{i-1}, \mathbf{o}, \mathbf{o}_{i+1}, \ldots, \mathbf{o}_n \rangle \in ext_{\mathscr{R}}(\imath^n))\}$

3. $\mathscr{CONV}_{i,j}$, for each $i, j, 1 \leq i \leq j$, is a function mapping $(\mathscr{R}_j \cup \mathscr{R}_{j+1} \cup \ldots)$ into $(\mathscr{R}_j \cup \mathscr{R}_{j+1} \cup \ldots)$ subject to the condition: $ext_{\mathscr{R}}(\mathscr{CONV}_{i,j}(\imath^n)) = \{\langle \mathbf{o}_1, \ldots, \mathbf{o}_{i-1}, \mathbf{o}_j, \mathbf{o}_{i+1}, \ldots, \mathbf{o}_{j-1}, \mathbf{o}_i, \mathbf{o}_{j+1}, \ldots, \mathbf{o}_n \rangle \mid \langle \mathbf{o}_1, \ldots, \mathbf{o}_i, \ldots, \mathbf{o}_j, \ldots, \mathbf{o}_n \rangle \in ext_{\mathscr{R}}(\imath^n)\}$.
4. $\mathscr{REFL}_{i,j}$ for each $i, j, 1 \leq i < j$, is a function mapping $(\mathscr{R}_j \cup \mathscr{R}_{j+1} \cup \ldots)$ into $(\mathscr{R}_{j-1} \cup \mathscr{R} \cup \ldots)$ subject to the condition: $ext_{\mathscr{R}}(\mathscr{REFL}_{i,j}(\imath^n)) = \{\langle \mathbf{o}_1, \ldots, \mathbf{o}_i, \ldots, \mathbf{o}_{j-1}, \mathbf{o}_{j+1}, \ldots, \mathbf{o}_n \rangle \mid \langle \mathbf{o}_1, \ldots, \mathbf{o}_i, \ldots, \mathbf{o}_j, \ldots, \mathbf{o}_n \rangle \in ext_{\mathscr{R}}(\imath^n) \text{and } \mathbf{o}_i = \mathbf{o}_j\}$
5. $\mathscr{COND}$ is a function from $(\mathscr{R}_1 \cup \mathscr{R}_2 \cup \ldots) \times (\mathscr{R}_1 \cup \mathscr{R}_2 \cup \ldots)$ into $(\mathscr{R}_2 \cup \mathscr{R}_3 \cup \ldots)$ subject to the condition: $ext_{\mathscr{R}}(\mathscr{COND}(\imath^n, \jmath^m)) = \{\langle \mathbf{o}_1, \ldots, \mathbf{o}_n, \mathbf{o}'_1, \ldots, \mathbf{o}'_m \rangle \mid \langle \mathbf{o}_1, \ldots, \mathbf{o}_m \rangle \notin ext_{\mathscr{R}}(\imath^n) \text{ or } \langle \mathbf{o}'_1, \ldots, \mathbf{o}'_M \rangle \in ext_{\mathscr{R}}(\jmath^m)\}$.

Interpretations (cont.)

6. $\mathscr{NEG}$ is a function from $(\mathscr{R}_1 \cup \mathscr{R}_2 \cup)$ into $(\mathscr{R}_1 \cup \mathscr{R}_2 \cup \ldots)$ subject to the condition:
$ext_{\mathscr{R}}(\mathscr{NEG}(\imath^n)) = \{\langle \mathsf{o}_1, \ldots, \mathsf{o}_n \rangle \mid \langle \mathsf{o}_1, \ldots, \mathsf{o}_n \rangle \notin ext_{\mathscr{R}}(\imath^n)\}$

- $ext_{\mathscr{A}} : \mathscr{R} \to \mathcal{P}(\mathscr{D})$. We call $ext_{\mathscr{A}}(\imath^1)$ the **encoding extension** of $\imath^1$
- $\mathscr{F}$ maps the simple names of the language to elements of the appropriate domain. For each object name $\kappa, \mathscr{F}(\kappa) \in \mathscr{D}$. For each relation name $\kappa^n, \mathscr{F}(\kappa^n) \in \mathscr{R}_n$.

Partitioning the λ-expressions (cont.)

But, we have a problem to solve about the λ-expressions.

Partitioning the λ-expressions (cont.)

But, we have a problem to solve about the λ-expressions.

We use μ, ξ, ζ as metavariables ranging over λ-expressions. Suppose μ is an arbitrary λ-expression. Then $\mu = [\lambda \nu_1 \ldots \nu_n \phi]$, for some $\phi, \nu_1, \ldots, \nu_n$.

Partitioning the λ-expressions (cont.)

1. If$(\exists i)(1 \leq i \leq n$ and ν_i is not the i^{th} free object variable in ϕ and i ixs the least such number), then where ν_j is the i^{th} free object variable in ϕ, μ is the **i,jth-conversin of**

   $$[\lambda\nu_1 \dots \nu_{i-1}\nu_j\nu_{i+1} \dots \nu_{j-1}\nu_i\nu_{j+1} \dots \nu_n\phi]$$

2. If μ is not the i, j^{th}-conversation of any λ-expression, then:
   - 2.1 if $\phi = (\neg\psi)$, μ is the **negation of** $[\lambda\nu_1 \dots \nu_n\phi]$
   - 2.2 if $\phi = (\psi \rightarrow \chi)$, and ψ and χ have no free object variable in common, then where $\nu_1, \dots, \nu_p$ are the variables in ψ and $\nu_{p+1}, \dots, \nu_n$ are the variables in χ, ν is the **conditionalization of** $[\lambda\nu_1 \dots \nu_p\psi]$ **and** $[\lambda\nu_{p+1} \dots \nu_n\chi]$
   - 2.3 if $\phi = (\forall\nu)\psi$, and ν is the i^{th} free object variable in ϕ, then μ is the **i$^{\text{th}}$-universalization of** $[\lambda\nu_1 \dots \nu_{i-1}\nu\nu_i\nu_{i+1} \dots \nu_n\psi]$.

Partitioning the λ-expressions (cont.)

3. If μ is none of the above, then if $(\exists i)(1 \leq i \leq n$ and ν_i occurs free in more than one place in ϕ and i is the least such number), then where:
   3.1 k is the number of free object variables between the first and second occurrences of ν_i,
   3.2 ϕ' is the result of replacing the second occurrences of ν_i with a new variable ν, and
   3.3 $j = i + k + 1$,

   μ is the **i,jth-reflection of** $[\lambda\nu_1 \ldots \nu_{i+k}\nu\nu_j \ldots \nu_n\phi']$

Partitioning the λ-expressions (cont.)

4. If μ is none of the above, then if o is the left most object term occurring in ϕ, then where:
   4.1 j is the number of free variables occurring before o.
   4.2 ϕ' is the result of replacing the first occurrence of o by a new variable ν, and
   4.3 $i = j + 1$,

   μ is the **i$^{\text{th}}$-plugging of** $[\lambda\nu_1 \ldots \nu_j\nu\nu_{j+1} \ldots \nu_n\phi']$ **by** o

5. If μ is none of the above, then
   5.1 ϕ is atomic
   5.2 $\nu_1, \ldots, \nu_n$ is the order in which these variables first occur in ϕ
   5.3 $\mu = [\lambda\nu_1 \ldots \nu_n\rho^n\nu_1 \ldots \nu_n]$, for some relation term ρ^n,

   and μ is called **elementary**

If given an interpretation $\mathscr{I}$ of our language, an $\mathscr{I}$-**assignment**, **f**, will be any function defined on the primitive variables of the language which satisfies the following two conditions:

1. where ν is any object variable, $\mathsf{f}(\nu) \in \mathscr{D}$
2. where π^n is any relation variable, $\mathsf{f}(\pi^n) \in \mathscr{R}_n$

Denotations

If given an interpretation $\mathscr{I}$ of our language, and an $\mathscr{I}$-assignment f, we recursively define **the denotation of** term π **with respect to** interpretation $\mathscr{I}$ **and** $\mathscr{I}$-assignment f("$\mathsf{d}_{\mathscr{I},\mathsf{f}}(\pi)$") as follows:

1. where κ is any primitive name, $\mathsf{d}_{\mathscr{I},\mathsf{f}}(\kappa) = \mathscr{F}_{\mathscr{I}}(\kappa)$
2. where ν is any object variable, $\mathsf{d}_{\mathscr{I},\mathsf{f}}(\nu) = \mathsf{f}(\nu)$
3. where π^n is any relation variable, $\mathsf{d}_{\mathscr{I},\mathsf{f}}(\pi^n) = \mathsf{f}(\pi^n)$
4. where $[\lambda\nu_1 \ldots \nu_n \rho^n \nu_1 \ldots \nu_n]$ is any elementary λ-expression, $\mathsf{d}_{\mathscr{I},\mathsf{f}}([\lambda\nu_1 \ldots \nu_n \rho^n \nu_1 \ldots \nu_n]) = \mathsf{d}_{\mathscr{I},\mathsf{f}}(\rho^n)$
5. where μ is the i^{th}-plugging of ξ by o, $\mathsf{d}_{\mathscr{I},\mathsf{f}}(\mu) = \mathscr{PLUG}_i(\mathsf{d}_{\mathscr{I},\mathsf{f}}(\xi), \mathsf{d}_{\mathscr{I},\mathsf{f}}(o))$
6. where μ is the i^{th}-universalization of ξ , $\mathsf{d}_{\mathscr{I},\mathsf{f}}(\mu) = \mathscr{UNIV}_i(\mathsf{d}_{\mathscr{I},\mathsf{f}}(\xi))$

Denotations (cont.)

7. where μ is the i, j^{th}-conversion of ξ, $\mathsf{d}_{\mathscr{I},\mathsf{f}}(\mu) = \mathscr{CONV}_{i,j}(\mathsf{d}_{\mathscr{I},\mathsf{f}}(\xi))$
8. where μ is the i, j^{th}-reflection of ξ, $\mathsf{d}_{\mathscr{I},\mathsf{f}}(\mu) = \mathscr{REFL}_{i,j}(\mathsf{d}_{\mathscr{I},\mathsf{f}}(\xi))$
9. where μ is the conditionalization of ξ and ζ, $\mathsf{d}_{\mathscr{I},\mathsf{f}}(\mu) = \mathscr{COND}_{i,j}(\mathsf{d}_{\mathscr{I},\mathsf{f}}(\xi), \mathsf{d}_{\mathscr{I},\mathsf{f}}(\zeta))$
10. where μ is the negation of ξ, $\mathsf{d}_{\mathscr{I},\mathsf{f}}(\mu) = \mathscr{NEG}(\mathsf{d}_{\mathscr{I},\mathsf{f}}(\xi))$

Satisfaction

We define **f satisfies** ϕ, recursively, as follows:

1. If $\phi = \rho^n o_1 \ldots o_n$, f satisfies ϕ iff $\langle \mathsf{d}_{\mathscr{I},\mathsf{f}}(o_1), \ldots, \mathsf{d}_{\mathscr{I},\mathsf{f}}(o_n) \rangle \in ext_{\mathscr{R}}(\mathsf{d}_{\mathscr{I},\mathsf{f}}(\rho^n))$
2. If $\phi = o\rho^1$, f satisfies ϕ iff $\mathsf{d}_{\mathscr{I},\mathsf{f}}(o) \in ext_{\mathscr{A}}(\mathsf{d}_{\mathscr{I},\mathsf{f}}(\rho^1))$
3. if $\phi = (\neg\psi)$, f satisfies ϕ iff f fails to satisfy ϕ
4. If $\phi = (\psi \rightarrow \chi)$, f satisfies ϕ iff f fails to satisfy ψ or f satisfies χ
5. If $\phi = (\forall\alpha)\psi$, f satisfies ϕ iff $(\forall \mathsf{f}')(\mathsf{f}'_{\bar{\alpha}}\mathsf{f} \rightarrow \mathsf{f}')$ satisfies ϕ), where: $\mathsf{f}'_{\bar{\alpha}}\mathsf{f} =_{df} \mathsf{f}'$ is an $\mathscr{I}$-assignment just like f except perhaps for what is assigns to α.

Truth under an Interpretation

ϕ is **true under** interpretation $\mathscr{I}$ iff every $\mathscr{I}$-assignment **f** satisfies ϕ. ϕ is **false under** $\mathscr{I}$ iff no $\mathscr{I}$-assignment **f** satisfies ϕ. ϕ is **valid** iff ϕ is true under all interpretations.

Outline

Outline

The Logical Axioms

1. $\phi \to (\psi \to \phi)$
2. $(\phi \to (\psi \to\to \chi)) \to ((\phi \to \psi) \to (\phi \to \chi))$
3. $(\neg\phi \to \neg\psi) \to ((\neg\phi \to \psi) \to \phi)$
4. $(\forall\alpha)\phi \to \phi^{\tau}_{\alpha}$, where τ is substitutable for α
5. $(\forall\alpha)(\phi \to \psi) \to (\phi \to (\forall\alpha)\psi)$, provided α is not free in ϕ
6. $(\forall x_1)\ldots(\forall x_n)([\lambda\nu_1\ldots\nu_n\phi]x_1\ldots x_n \leftrightarrow \phi^{x_1,\ldots,x_n}_{\nu_1\ldots,\nu_n})$
7. $[\lambda\nu_1\ldots\nu_n\rho^n\nu_1\ldots\nu_n] = \rho^n$

Rules of Inference

1. From ϕ and $\phi \to \psi$, we may infer ψ
2. (*UI*) from ϕ, we may infer $(\forall\alpha)\phi$

Outline

Proper Axioms

1. (*E−IDENTITY*) $x =_E y \leftrightarrow E!x \wedge E!y \wedge (\forall F)(Fx \leftrightarrow Fy)$
2. (*NO−CODER*) $E!x \rightarrow \neg(\exists F)xF$
3. (*IDENITITY*) $\alpha = \beta \rightarrow (\phi(\alpha, \alpha) \leftrightarrow \phi(\alpha, \beta))$, where $\phi(\alpha, \beta)$ is the result of replacing some, but not necessarily all, free occurences of α by β in $\phi(\alpha, \alpha)$, provided β is substitutable for α in the occurences of α it replaces.
4. (*A−OBJECTS*) $(\exists x)(A!x \wedge (\forall F)(xF \leftrightarrow \phi))$, for any formula ϕ where x is not free

Two Theorems

Definition

1. $x = y \ =_{df} x = y \vee (A!x \wedge A!y \wedge (\forall F)(xF \leftrightarrow yF))$

2. $F^1 = G^1 =_{df} (\forall x)(xF^1 \leftrightarrow xG^1)$

3. $F^n = G^n =_{df} (\forall x_1 \ldots x_{n-1})([\lambda y F^n y x_1 \ldots x_{n-1}] = [\lambda y G^n y x_1 \ldots x_{n-1}] \wedge [\lambda y F^n x_1 y x_2 \ldots x_{n-1}] = [\lambda y G^n x_1 y x_2 \ldots x_{n-1}] \wedge \ldots \wedge [\lambda y F^n x_1 \ldots x_{n-1} y] = [\lambda y G^n x_1 \ldots x_{n-1} y])$(where $n > 1$)

Theorem (IDENTITY INTRODUCTION, =I)

$\alpha = \alpha$, where α is any variable.

Proof.

(sketch) If α is an object variable x and $E!x$, then since we have $(\forall F)(Fx \leftrightarrow Fx)$ from propositional logic and *UI*, we may use (*E−IDENTITY*) to prove $x =_E x$. So $x = x$, by Definition 1. If $\neg E!x \ldots$ If α is $F^1 \ldots$ If α is $F^n \ldots$ □

Let us use the standard notation $(\exists! x)\psi$ (there is a unique *x* such that ψ) to abbreviate $(\exists x)(\psi \wedge (\forall y)(\psi^{y}_{x} \rightarrow y = x))$.

Theorem (UNIQUENESS)

$(\exists! x)(A!x \wedge (\forall F)(xF \leftrightarrow \phi))$ *for any formula* ϕ *where x is not free.*

Proof.

Firstly, we prove the existence by the *A—OBJECTS*. And then we prove the uniqueness by contradiction. There could not be distinct such objects since we cannot give a formula ϕ which give us two different conditions about properties. □

Outline

An Expansion of the Theory

- Let us stipulate that where ϕ is any formula with one free x-variable, $(\iota x)\phi$("the object x such that ϕ") is to be a complex object term of our language. Semantically, we interpret descriptions $(\iota x)\phi$ as denoting the unique object which satisfies ϕ, if there is one, and as not denoting anything if there is not one.
- Axiom(*DESCRIPTIONS*)

$$\psi_{\upsilon}^{(\iota x)\phi} \leftrightarrow (\exists! y)\phi_x^y \wedge (\exists y)(\phi_x^y \wedge \psi_v^y)$$

where ψ is any atomic formula or defined object identity formula with one free variable υ.

Definition (Form(x, G))

x is a Form of $G =_{df} A!x \wedge (\forall F)(xF \leftrightarrow F = G)$

Theorem (1)

$(\forall G)(\exists x)Form(x, G)$

Proof.

By A-OBJECTS and UI. □

Theorem (2)

$(\forall G)(\exists! x)Form(x, G)$

Proof.

By UNIQUENESS and UI. □

So now we know that the description $(\iota x)(A!x \wedge (\forall F)(xF \leftrightarrow F = G))$(the Form of G) always has a denotation. For convenience, let us use "Φ_G" to abbreviate it.

Theorem (3)

$\Phi_G G$*(the Form of G encodes G)*

Proof.

By DESCRIPTIONS, $\Phi_G G \leftrightarrow (\exists! y)(A!y \wedge (\forall F)(yF \leftrightarrow F = G)) \wedge (\exists y)(A!y \wedge (\forall F)(yF \leftrightarrow F = G) \wedge yG)$. The right side of this biconditional is easily obtainable from Theorem(2). □

Definition (Part(y, x))

y participates in $x =_{df} (\exists F)(xF \wedge Fy)$

Theorem (4)

$x \neq y \wedge Fx \wedge Fy \rightarrow (\exists u)(u = \Phi_F \wedge Part(x, u) \wedge Part(y, u))$

Proof.

Assume $a \neq b, Pa$, and Pb, where a, b are arbitrary objects and P is an arbitrary property. By =I, we have $\Phi_P = \Phi_P$. By Theorem(3) and the above assumptions, we have $\Phi_P P \wedge Pa$. So $(\exists G)(\Phi_P G \wedge G_a)$, i.e., Part($a, \Phi_P$). Similarly, Part($b, \Phi_P$). So $\Phi_P = \Phi_P \wedge Part(a, \Phi_P) \wedge Part(b, \Phi_P)$. So $(\exists u)(u = \Phi_P \wedge \text{Part}(a, u) \wedge \text{Part}(b, u))$ □

Theorem (5)

$Fx \leftrightarrow Part(x, \Phi_F)$

Proof.

($\rightarrow$) Assume Fx. By Theorem (3), Part(x, Φ_F).
($\leftarrow$) Assume Part (x, Φ_F). Call the property Φ_F encodes G and x exemplified G. Since Φ_F encodes just F, it must be that $G = F$. So Fx. □

Definition

We call the property $[\lambda x \neg E!x]$ **Platonic existence** and the notation is $\bar{E}!$

Theorem (6)

$(\forall x)(\exists F)(x = \Phi_F) \rightarrow \bar{E}!x)$.

Proof.

By the definition of Forms, We have known that Φ_F is abstract, and by the definition of the abstracts we get the theorem. □

We can call $\Phi_{\bar{E}}$ **Platonic Being**, or **Reality**. From Theorem(5) and (6) it follows that:

Theorem (7)

$(\forall x)((\exists F)(x = \Phi_F) \rightarrow \textbf{\textit{Part}}(x, \Phi_{\bar{E}}))$

Reicher, Maria
Nonexistent Objects
The Stanford Encyclopedia of Philosophy (Winter 2016 Edition), Edward N. Zalta (ed.), URL = <https://plato.stanford.edu/archives/win2016/entries/nonexisten
objects/>.

Alan McMichael & Edward N. Zalta
An Alternative Theory of Nonexistent Objects
Journal of Philosophical Logic, Vol. 9, No. 3 (Aug., 1980), pp. 297-313

Edward N. Zalta
Abstract Objects
D.Reidel Publishing Company, 1983

Thanks!