K + K = 120: Papers dedicated to L. Kálmán & A. Kornai on the occasion of their 60th birthdays, pp. 425–443

Indefinite descriptions in typed lambda calculus

ZOLTÁN MOLNÁR
Budapest University of Technology and Economics
mozow@math.bme.hu

KEYWORDS

lambda calculus
epsilon symbol
indefinite descriptions

ABSTRACT

The epsilon calculus seems to be an appropriate environment for modelling the meaning of definite and indefinite descriptions in a natural language. A philosopher of language may ask whether Russell's meaning theory on descriptions is applicable in this language or not. Or more precisely, in what circumstances a sentence (containing an epsilon-expression) has a contextual meaning, and what its logically equivalent quantified reformulation is. The question was answered for first order languages earlier, but the conditions were full of technical complications and the construction applied difficult semantics. In this paper, the question is answered for a typed lambda calculus, in an easier way and by a simpler semantics.

1. Introduction

1.1. Hilbert's epsilon, descriptions and FOL

The first-order language (FOL) extended by the Hilbertian variable binding operator ε is possibly a good choice as an environment modelling a couple of formal linguistic and language philosophical phenomena concerning descriptions.[1] The term

(1) $(\varepsilon x)\varphi$

where φ is a FOL formula and x is a variable, has the following intuitive meaning:

(2) "an F, if there is any F at all"

[1] See Slater (2007); Kneebone (1963, 100).

where predicate F is the intended meaning or the natural language translation of the formula φ. (Here, the notion of translation is due to Tarski. In the present case, the object-language is FOL and the meta-language is the natural language.[2]) The intuition above follows straightforward from the *first epsilon (or transfinite) axiom* introduced by David Hilbert, which is the following formula scheme

$$(\exists x)\varphi(x) \to \varphi((\varepsilon x)\varphi(x))$$

The intended meaning of the epsilon term shows that $(\varepsilon x)\varphi$ can be called a *conditional indefinite description,* since "an F" alone is an indefinite description, with the addition of the conditional clause "there is any F at all" it becomes a different linguistic entity with, perhaps, a different meaning. Obviously, I do not have to mention that the meaning of the phrase "an F" is itself a problematic one. Therefore, the problem of the semantic difference between "an F" and "an F, if there is any F at all" is also a tricky one. In the paper I am committed to the standpoint that these phrases have the same meaning.

In order to show an application of Hilbert's symbol let me provide a formal reconstruction and analysis of the sentence

(3) The man drinking a martini is interesting-looking.

in FOL extended by ε (this extended language is denoted by FOL+ε).[3] Since, FOL+ε does not contain definite descriptions, the phrase *the man drinking a martini* can be seen as a special case of the use of ε. A possible solution is to add a uniqueness clause to the following formula: "$\mathsf{man\ drinking\ a\ martini}(x)$" (the formula in FOL+$\varepsilon$ expressing the natural language predicate *...is a man and drinks a martini*; see Slater 2009, 417):

$$(\varepsilon x)(\mathsf{man\ drinking\ a\ martini}(x)\ \&\ (\forall y)(\mathsf{man\ drinking\ a\ martini}(y)\ \equiv\ (x = y))$$

Let us denote the term above by

(4) $(\varepsilon_{\mathrm{D}} x)\mathsf{man\ drinking\ a\ martini}(x)$

Then sentence (3) is formulated as follows:

(5) $\mathsf{interesting\text{-}looking}((\varepsilon_{\mathrm{D}} x)(\mathsf{man\ drinking\ a\ martini}(x)))$

[2] Cf. the notion of translation as applied in Convention T in (Tarski 1956, 188).

[3] The original sentence can be found in Donnellan (1966).

Let me remark again, that the claim that the phrase *the man drinking a martini* can be expressed by $(\varepsilon_{\mathrm{D}}x)\mathsf{man\ drinking\ a\ martini}(x)$ is not an obvious one, however a possibly good enough working hypothesis. Without a man holding martini in his hand, the meaning of *the man drinking a martini* is as vague as the meaning of the phrase *the man drinking a martini, if anybody at all.*

Accepting the hypothesis above, by sentence (3) one can refer to the interesting-looking person in question, even if he holds a glass of water in his hand. In this case, the semantic value of the term (4) is a person – not drinking a martini – who seems to be interesting.

The problem of sentence (5) reminds one of Russell's Theory of Descriptions (RTD). In Russell's *On Denoting* or in Whitehead & Russell (1910/1967) it is proposed that descriptions must not be treated as proper names, but as incomplete parts of quatified sentences.

> "Thus we must either provide a denotation in cases in which it is at first sight absent, or we must abandon the view that the denotation is what is concerned in propositions which contain denoting phrases." (Russell 1905, 484)

> "According to the view I advocate, a denoting phrase is essentially *part* of a sentence, and does not, as like most single words, have any significance on its own account." (*ibid.*, 488)

According to RTD, a description D, as a denoting phrase, is not interchangeable by an other individual name N which is identical to D, since D is meaningless in separation, and has only contextual meaning. Russell in *On Denoting* (Russell 1905) gives a FOL reformulation for sentences of the form (3), but in the general case, when the natural language sentence contains more than one descriptions or a lot of logical operators the FOL reformulation can be carried out along different lines. One must mind the scope of logical operators and descriptions. Hence, in the general case RTD is appears to be a FOL reformulation program, in the spirit of the treatment of the simple case described in Russell (1905). At this point, a bit naive question arises.

(6) Is the closed formula (5) equivalent to a plain, quantified one?

If it is in general, then RTD, or rather its quantificational program, is applicable to FOL+ε, in the sense that an epsilon-term, containing a closed formula, can be considered as incomplete part of a quantified reformulation.[4] If it is not equivalent in general, then for FOL+ε Donnellan's proposal holds (i.e., sometimes descriptions have separate meaning, too; see Donnellan 1966). The answer to the question seems to be the latter. The term $(\varepsilon x)\varphi$ is a referring one (its semantic value is always defined) and its semantics is unproblematic, even if there is no φ, at least if the reference of $(\varepsilon x)\varphi$ is not a φ. Nevertheless, note that RTD is a strategy proposed to solved the problem 'how to deal with descriptions' and not a (mathematical) thesis. In FOL+ε, $(\varepsilon x)\varphi$ is a proper name (in the sense of Russell), hence the question must be rewritten in a weaker form. But what will be this weakened question?

It is well-known that if the truth value of the sentence $\psi((\varepsilon x)\varphi)$ in any model does not depend on the semantic value of $(\varepsilon x)\varphi$, then there is a FOL reformulation of $\psi((\varepsilon x)\varphi)$.[5] Hence, if $\psi((\varepsilon x)\varphi)$ is an epsilon-invariant formula (its semantic value is independent of the value of the containing epsilon-term) then the term $(\varepsilon x)\varphi$ can be eliminated from $\psi((\varepsilon x)\varphi)$ by a logically equivalent reformulation. The problem is that this plain FOL reformulation is not an explicit or transparent one. The proof of the theorem applies Craig's Interpolation Theorem, which is a pure existence theorem not giving the needed explicit formula. Hence, in the light of the above considerations, the relevant question is the following.

> "Is there an explicit, transparent, well-explainable FOL reformulation of $\psi((\varepsilon x)\varphi)$, provided that $\psi((\varepsilon x)\varphi)$ is epsilon-invariant (in some model)?"

For FOL+ε, the question has been positively answered in Molnár (2013), however with the application of a lot of technical conditions. When one changes FOL to lambda calculus the picture becomes much more clear. The point is that, in FOL the substitution $\psi[x/(\varepsilon x)\varphi]$ is only a meta-language operation, but in the lambda-calculus it is encoded into the object-language via the application MN, where M is an expression of the lambda-language and N is an epsilon-term of the form $(\varepsilon x)P$.

[4] The crucial point in the tradition of RTD is not that what the FOL reformulation is, but wheter there is any such reformulation. For instance, as Zvolenszky puts the question: "Initially, at issue was the meaning of a specific, rather narrow class of expressions, incomplete definite descriptions: are they devices of reference or of quantification?" (Zvolenszky 2007, 1).

[5] One can call it *Caicedo's Theorem* or the *Blass–Gurevich Theorem*. Its proof first presented in Caicedo (1995), but Blass and Gurevich (2000) claim before the theorem (Prop. 3.2.) that the proposition is a folklore and "it is mentioned in Caicedo (1995) without a reference".

1.2. Hilbert's epsilon and the lambda operator

In section 2, a syntax and semantics will be given for the epsilon symbol in the context of typed lambda calculus (TL). The syntactic notions will be the well-known ones, but in the definitions different way will be followed, based on labeled, ordered trees.[6] Since, by the Curry–Howard Correspondence, TL is closely related to the proof theory of the natural deduction system of propositional logic, we make use of the possibility to define the TL notions of TL syntax the same style as proofs. The form of the definitions will fit this doctrine and a tree-based method will be applied.

In section 3, it will be seen that in TL the result can be reached much more faster than in FOL. There is no need to refer to the so-called intensional and substitutional epsilon semantics.[7] The strategy will be the following. The typed lambda language extended by Hilbert's epsilon ($\mathcal{L}_\lambda^{\forall\varepsilon}$) will be considered as a formal model of the fragment of the natural language containing descriptions. Then, if it is possible, the epsilon expressions will be eliminated and the sentences containing them will be mapped, in an explicit way, to the epsilon-free quantified reduct $\mathcal{L}_\lambda^{\forall}$ of $\mathcal{L}_\lambda^{\forall\varepsilon}$. The plain lambda language reformulation will keep the logical truth in the model. Giving Montague-semantics to the extended language and to the plain epsilon-free language as models (the $(\mathfrak{M}, f)$-s and the $\mathfrak{M}$-s below, respectively) the construction will be unproblematic.

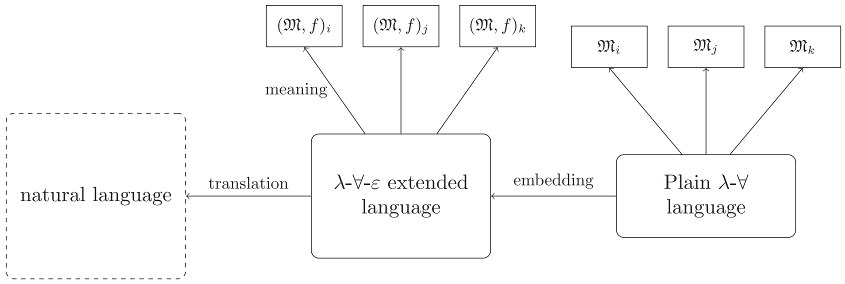


Figure 1: Chain of fragments. The natural language, the λ-$\forall$-ε expressible fragment and the λ-$\forall$ expressible fragment.

[6] It is not easy to refer to a single book or paper, but the book Simmons (2000) (with the programmatic subtitle "Taking the Curry–Howard Correspondence Seriously") surely uses the tree technique that I follow.

[7] Note that, Ahrendt and Giese introduced several types of epsilon semantics. See (Ahrendt & Giese 1999, Def. 4,5). In Molnár (2013) the substitutional semantics was applied. Now, in TL the extensional semantics (see Molnár 2013, 821 or Monk 1976, Def. 29.23) will be enough.

In section 4, it will be pointed out that the result is not less effective than the RTD proposed by Russell.

2. Syntax and semantics of typed lambda system with epsilon

2.1. Syntax

For building the syntax a tree-based method is chosen (parsing or construction trees), which is much more transparent than the old-fashioned character sequence technique. One thing to note is that here the trees grop upward, as those used by by linguists in Combinatory Categorial Grammar, or, what the main motivation is, in proof theory of the style used in natural deduction.

The definitions below are basically combinations of the well-known ones from Troelstra & Schwichtenberg (2000, Sec. 1) and from Sørensen & Urzyczyn (1998).

The so called *typeability relation* ($\vdash$) is a pure syntactic relation that joins the expressions of the lambda calculus to types with respect to a fixed set of typed variables called *context.* Of course, the relation $\vdash$ plays a fundamental role in the Curry–Howard Isomorphism, which links the lambda expressions to proof trees of the natural deduction system of the implicational logic.

Definition 1. The *language of types* is the tuple $\mathcal{L}_{\mathrm{Typ}} = \langle \iota, o, (,), [,] \rangle$. The set of its strings $\mathrm{Str}(\mathcal{L}_{\mathrm{Typ}})$ contains the finite sequences of the characters from $\{\iota, o, (,), [,]\}$. A *construction tree* Π of the string $\gamma \in \mathrm{Str}(\mathcal{L}_{\mathrm{Typ}})$ is a finite, labeled, ordered tree such that the labels of Π are from $\mathrm{Str}(\mathcal{L}_{\mathrm{Typ}})$ and

1. the labels of the leaves of Π come from the set $\{\iota, o\}$,
2. the branch nodes of Π (these are not leaves) and their labels are of the form

$$\begin{tikzcd}
\alpha \arrow[dr, no head] & & \beta \arrow[dl, no head] \\
& {[\alpha(\beta)]} &
\end{tikzcd}$$

3. the root of Π is γ.

If there is a tree Π such that Π is a construction tree of $\alpha \in \mathrm{Str}(\mathcal{L}_{\mathrm{Typ}})$, then α is said to be a *type (expression)* in $\mathcal{L}_{\mathrm{Typ}}$. The set of all types in $\mathcal{L}_{\mathrm{Typ}}$ is denoted by $\mathrm{Exp}(\mathcal{L}_{\mathrm{Typ}})$. (Cf. Troelstra & Schwichtenberg 2000, Def. 1.2.1 (p. 9); Def. 1.1.7. (p. 7).)

Note that the construction tree of a type is unique. The construction tree of the type α is denoted by $\mathrm{Tree}(\alpha)$. The reference to brackets $[,]$ is avoided when a type α is well-known and its construction tree can be completely reconstructed without them.

Intuitively, ι is the type of individuals and o is the type of sentences. The compound type $o(\iota)$ is, for example, the grammatical type of the single-variable predicates.

Definition 2. A *lambda language* is a tuple $\mathcal{L}_\lambda = \langle V, C, (,), \lambda, [,]\rangle$, where V is an infinite and C is non-empty set and V is disjoint to C. $\mathrm{Str}(\mathcal{L}_\lambda)$ contains the finite sequences from $V \cup C \cup \{\lambda, (,), [,]\}$. A *construction tree* Π of the $M \in \mathrm{Str}(\mathcal{L}_\lambda)$ is a finite, labeled, ordered tree such that the labels of Π are from $\mathrm{Str}(\mathcal{L}_\lambda)$ and

1. the labels of the leaves of Π come from the set $V \cup C$,
2. the branch nodes of Π and their labels are of the form

$$\begin{tikzcd}
P \arrow[dr, no head] & & Q \arrow[dl, no head] & & P \arrow[d, no head] \\
& {[P(Q)]} & & & {[(\lambda x)P]}
\end{tikzcd}$$

3. the root of Π is M.

If there is a tree Π such that Π is a construction tree of $M \in \mathrm{Str}(\mathcal{L}_\lambda)$, then M is said to be an *expression* in $\mathcal{L}_\lambda$. The set of all expression in $\mathcal{L}_\lambda$ is denoted by $\mathrm{Exp}(\mathcal{L}_\lambda)$.

The elements of V are called the variables of $\mathcal{L}_\lambda$ and V is denoted by $\mathrm{Var}(\mathcal{L}_\lambda)$. The elements of C are the constants of $\mathcal{L}_\lambda$ and C is denoted by $\mathrm{Const}(\mathcal{L}_\lambda)$. (Cf. Troelstra & Schwichtenberg 2000, Def. 1.2.2 (p. 9); Def. 1.1.7. (p. 7).)

Note that the construction tree of an expression is unique. The construction tree of the expression M is denoted by $\mathrm{Tree}(M)$. The *height* of $\mathrm{Tree}(M)$ is defined by the well-known manner and is denoted by $|\mathrm{Tree}(M)|$.

Referring to brackets $[,]$ is avoided when an expression M is known and its construction tree can be completely reconstructed without them.

Definition 3. Let $\langle V, C, (,), \lambda, [,]\rangle$ be a lambda language. The tuple $\mathcal{L}_\lambda = \langle V, C, (,), \lambda, [,], Z\rangle$ is a *typed lambda language,* if $Z : C \to \mathrm{Exp}(\mathcal{L}_{\mathrm{Typ}})$. The function Z is denoted by $\mathrm{CnstTp}(\mathcal{L}_\lambda)$.

Definition 4. Let $\mathcal{L}_\lambda$ be a lambda language and let $\Xi \subseteq \mathrm{Var}(\mathcal{L}_\lambda)$ be a non-empty finite set. A function $f : \Xi \to \mathrm{Exp}(\mathcal{L}_{\mathrm{Typ}})$ is called a *context,* and the set of all contexts is denoted by $\mathrm{Cont}(\mathcal{L}_\lambda)$. $(\Xi : \Gamma) \in \mathrm{Cont}(\mathcal{L}_\lambda)$ denotes a function f with domain Ξ and range Γ. If $f = (\Xi : \Gamma)$ is a context, and $x \in \Xi$ then $(x : \gamma)$ denotes $f(x) = \gamma$.

For a typed lambda language $\mathcal{L}_\lambda$ the sets of variables, expressions, contexts etc. defined and denoted by the same manner as for a lambda languages.

Definition 5. Let $\mathcal{L}_\lambda$ be a typed lambda language. By induction on the height of the construction tree of the expressions, relation

$$(\Xi : \Gamma) \vdash M : \varphi$$

will be defined as follows for every context $(\Xi : \Gamma) \in \mathrm{Cont}(\mathcal{L}_\lambda)$, expression $M \in \mathrm{Exp}(\mathcal{L}_\lambda)$ and type φ. $\vdash$ is called the *typeability relation.*

1. Let $|\mathrm{Tree}(M)| = 1$.

   a. If $c \in \mathrm{Const}(\mathcal{L}_\lambda)$ and $(\Xi : \Gamma)$ is a context, then $(\Xi : \Gamma) \vdash c : \varphi$, if $\varphi = \mathrm{CnstTp}(\mathcal{L}_\lambda)(c)$.

   b. If $x \in \mathrm{Var}(\mathcal{L}_\lambda)$ and $(\Xi : \Gamma)$ is a context, then $(\Xi : \Gamma) \vdash x : \varphi$, if $(x : \varphi) \in (\Xi : \Gamma)$.

2. Let us suppose that $n > 1$ and for every $(\Upsilon : \Delta)$ context, type ψ and expression N with $|\mathrm{Tree}(N)| < n$, the relation $(\Upsilon : \Delta) \vdash N : \psi$ is defined. Let $(\Xi : \Gamma)$ be a context, φ a type and M an expression such that $|\mathrm{Tree}(M)| = n$.

   a. Let $M = P(Q)$. Then $(\Xi : \Gamma) \vdash M : \varphi$, if $(\Xi : \Gamma) \vdash Q : \beta$ and $(\Xi : \Gamma) \vdash P : \alpha(\beta)$ and $\varphi = \alpha$.

   b. Let $M = (\lambda x)P$. Then $(\Xi : \Gamma) \vdash M : \varphi$, if $(\Upsilon : \Delta) \vdash P : \alpha$, $\varphi = \alpha(\beta)$ and $(\Xi : \Gamma) = (\Upsilon : \Delta) \setminus \{(x : \beta)\}$.[8],

For some examples, see Troelstra & Schwichtenberg (2000, 10).

[8] Cf. Sørensen & Urzyczyn (1998, 41, def. 3.1.1.).

2.2. Montague-semantics

Definition 6. Let $M \neq \emptyset$. By induction on $|\mathrm{Tree}(\varphi)|$, the domain set $D_M(\varphi)$ of type $\varphi \in \mathcal{L}_{\mathrm{Typ}}$ is defined as follows.

1. $D_M(o) = \{\mathsf{T}, \mathsf{F}\}, \qquad D_M(\iota) = M$
2. If $D_M(\alpha)$ and $D_M(\beta)$ is defined earlier, then

$$D_M(\alpha(\beta)) = {}^{D_M(\beta)}D_M(\alpha)$$

where ${}^{D_M(\beta)}D_M(\alpha)$ is the set $\{f : D_M(\beta) \to D_M(\alpha)\}$.

If M is fixed, then $D(\varphi)$ is written instead.

Definition 7. If $M \neq \emptyset$, $\mathcal{L}_\lambda$ is a lambda-language and $(\Xi : \Gamma)$ is a context, then a function $a : \mathrm{Var}(\mathcal{L}_\lambda) \to \cup_{\varphi \in \mathcal{L}_{\mathrm{Typ}}} D_M(\varphi)$ is an *assignation* of the variables. The assignation a is an *assignation of the type* $(\Xi : \Gamma)$*,* if for every $x \in \Xi$, $a(x) \in D_M(\alpha)$ whenever $(x : \alpha) \in (\Xi : \Gamma)$.

Definition 8. Let $\mathcal{L}_\lambda$ be a typed lambda-language, $M \neq \emptyset$. The tuple $\mathfrak{M} = \langle M, \mathrm{Ip}^{\mathfrak{M}} \rangle$ is a *model* over the language $\mathcal{L}_\lambda$, if $\mathrm{Ip}^{\mathfrak{M}} : C \to \cup_{\varphi \in \mathcal{L}_{\mathrm{Typ}}} D(\varphi)$ such that $\mathrm{Ip}^{\mathfrak{M}}(c) \in D(\mathrm{CnstTp}(\mathcal{L}_\lambda)(c))$.

Definition 9. Let $\mathcal{L}_\lambda$ be a typed lambda-language, $\mathfrak{M} = \langle M, \mathrm{Ip}^{\mathfrak{M}} \rangle$ a model over the language $\mathcal{L}_\lambda$, $(\Xi : \Gamma)$ a context and a an assignation of the type $(\Xi : \Gamma)$. Suppose that for $N \in \mathrm{Exp}(\mathcal{L}_\lambda)$ there is a type φ such that $(\Xi : \Gamma) \vdash N : \varphi$. By induction on $|\mathrm{Tree}(N)|$ the semantic value $[\![N]\!]_a^{\mathfrak{M}}$ in context $(\Xi : \Gamma)$ is defined as follows.

1. If $N = c \in \mathrm{Const}(\mathcal{L}_\lambda)$, then

$$[\![c]\!]_a^{\mathfrak{M}} = \mathrm{Ip}^{\mathfrak{M}}(c).$$

2. If $N = x \in \mathrm{Var}(\mathcal{L}_\lambda)$, then

$$[\![x]\!]_a^{\mathfrak{M}} = a(x).$$

3. Let $N = P(Q)$, then

$$[\![P(Q)]\!]_a^{\mathfrak{M}} = [\![P]\!]_a^{\mathfrak{M}}([\![Q]\!]_a^{\mathfrak{M}}).$$

4. Let $N = (\lambda x)P$ and let the assignment $a[x \to \xi]$ be the following:

$a[x \to \xi](y) = a(y)$ for every variable $y \neq x$, and $a(x) = \xi$.

Then

$$[\![(\lambda x)P]\!]^{\mathfrak{M}}_{a} : D(\alpha) \to D(\beta)\ ; \xi \mapsto [\![P]\!]^{\mathfrak{M}}_{a[x\to\xi]}$$

where $(\Xi : \Gamma) \vdash x : \alpha$ and $(\Xi : \Gamma) \vdash P : \beta$.

Note that if N is not typeable in a context $(\Xi : \Gamma)$, i.e., there is no type φ such that

$$(\Xi : \Gamma) \vdash N : \varphi$$

then N has no semantic value in an assignment of the type of the context. For example, let the type of the constant c be $o(\iota)$ and the context $(\Xi : \Gamma) = \{(x : o)\}$. Then the expression $c(x)$ is not typeable from the context $\{(x : o)\}$, since the argument of c must be an expression of the type ι. However, $c(x)$ is a well-defined expression, it has no semantic value in the context $\{(x : o)\}$.

2.3. Logical and epsilon extensions

The logical operators will be defined as constants of certain types. If $\mathcal{L}_\lambda$ is a typed lambda language, then it could be extended by the following constants.

1. $\neg : o(o)$ $\quad\mathrm{Ip}^{\mathfrak{M}}(\neg) : \mathsf{T} \mapsto \mathsf{F}, \mathsf{F} \mapsto \mathsf{T}$ in a model $\mathfrak{M}$,
2. $\vee : o(o(o))$ $\quad\mathrm{Ip}^{\mathfrak{M}}(\vee) : (\mathsf{F}, \mathsf{F}) \mapsto \mathsf{F},$ and T otherwise in a model $\mathfrak{M}$,
3. $\forall : o(o(\iota))$ $\quad\mathrm{Ip}^{\mathfrak{M}}(\forall) : \{\mathsf{T}, \mathsf{F}\}^M \to \{\mathsf{T}, \mathsf{F}\},\ (M \to \{\mathsf{T}, \mathsf{F}\}; \xi \mapsto \mathsf{T}) \mapsto \mathsf{T},$ and F otherwise in a model $\mathfrak{M}$,
4. $\varepsilon : \iota(o(\iota))$ $\quad\mathrm{Ip}^{\mathfrak{M}}(\varepsilon) : \{\mathsf{T}, \mathsf{F}\}^M \to M : f \mapsto g(\{\xi \in M \mid f(\xi) = \mathsf{T}\})$, where g is a fixed *choice function* $\mathcal{P}(M) \to M$ such that $g(S) \in S$, if $S \neq \emptyset$ and $g(S) \in M$, if $S = \emptyset$, in a model $\mathfrak{M}$.

In what follows, two specific extensions will be made use of, the *plain extension*

$$\mathcal{L}^{\forall}_{\lambda} \text{ with } \mathrm{Const}(\mathcal{L}^{\forall}_{\lambda}) = \mathrm{Const}(\mathcal{L}_\lambda) \cup \{\neg, \vee, \forall\}$$

and the *epsilon extension*

$$\mathcal{L}^{\forall\varepsilon}_{\lambda} \text{ with } \mathrm{Const}(\mathcal{L}^{\forall\varepsilon}_{\lambda}) = \mathrm{Const}(\mathcal{L}_\lambda) \cup \{\neg, \vee, \forall, \varepsilon\}.$$

If $\mathfrak{M}$ is a model of $\mathcal{L}^{\forall}_{\lambda}$, then $(\mathfrak{M}, g)$ will denote the (expanded) model of the $\mathcal{L}^{\forall\varepsilon}_{\lambda}$ extension with a choice function g described above.[9]

[9] Actually, epsilon-terms are a special kind of Skolem functions; it is pointed out in Monk (1976, 481) and in Mints (1996, sec. 2).

Some further (classical) notations will also be used:

$$P \to Q = \vee([\neg(P)](Q)), P\&Q = \neg(\vee([\neg(P)](\neg(Q)))), (\forall x)P = \forall((\lambda x)P)$$

$$(\varepsilon x)P = \varepsilon((\lambda x)P).$$

For further purposes the language $\mathcal{L}_\lambda^{\forall\varepsilon=}$ using *identity* of individuals is also introduced and the meaning of $=$ is defined as

5. $=: (o(\iota))(\iota) \quad \mathrm{Ip}^{\mathfrak{M}}(=) : M^2 \to \{\mathsf{T}, \mathsf{F}\}, (x, y) \mapsto \mathsf{T}$, if $x = y$ and F otherwise in a model $\mathfrak{M}$.

2.4. Examples

Proposition 1. Let x be a variable and $(\mathfrak{M}, g)$ be a model over the language $\mathcal{L}_\lambda^{\forall\varepsilon=}$. Then

1. $\vdash (\forall x)(x = x) : o$
2. $\vdash (\varepsilon x)(x \neq x) = (\varepsilon x)(x \neq x) : o$
3. $[\![(\forall x)(x = x)]\!]^{(\mathfrak{M},g)} = [\![(\varepsilon x)(x \neq x) = (\varepsilon x)(x \neq x)]\!]^{(\mathfrak{M},g)} = \mathsf{T}$

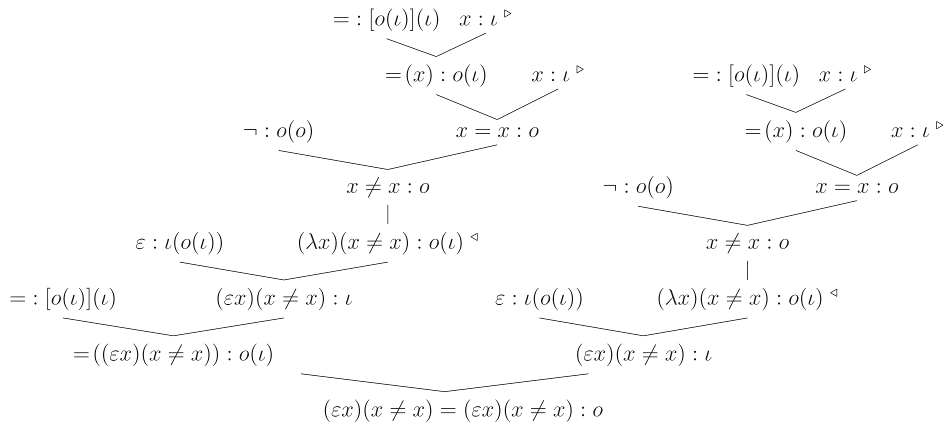
Proof. (1)

$$\begin{tikzcd}
 & & {=: [o(\iota)](\iota)} \arrow[dr, no head] & & {x : \iota^{\,\triangleright}} \arrow[dl, no head] & & \\
 & & & {=(x) : o(\iota)} \arrow[drr, no head] & & & {x : \iota^{\,\triangleright}} \arrow[dl, no head] \\
 & & & & & {x = x : o} \arrow[d, no head] & \\
{\forall : o(o(\iota))} \arrow[drr, no head] & & & & & {(\lambda x)(x = x) : o(\iota)^{\,\triangleleft}} \arrow[dlll, no head] & \\
 & & {\forall((\lambda x)(x = x)) : o} & & & &
\end{tikzcd}$$

Here $(= (x))(x)$ is denoted by $x = x$. The proof tree above shows that the expression $\forall((\lambda x)(x = x))$, which is the same as $(\forall x)(x = x)$, is typeabe by the type o. The labels $^{\triangleright}$ on the left sides of the leaves mark the places which are called the "dischargeable premises" in proof theory. $^{\triangleleft}$ marks the node where they are abandoned. According to part (2b) of Definition 5, both the $x : \iota$-s are discharged by the node $(\lambda x)([= (x)](x)) : o(\iota)$, i.e.,

$(x : \iota)$ can be canceled from the context, which is now an empty set. Note that, the use of the labels $^{\triangleleft}$ and $^{\triangleright}$ is completely unnecessary, since the role of the variable x is exactly that of the triangles. The variable x in the leaves marks the "dischargeable premises" and the symbol (λx) marks the node discharging the premises labeled by the free variable x, after which x becomes a bound variable.

(2)


$= : [o(\iota)](\iota)$ $x : \iota^{\triangleright}$
$=(x) : o(\iota)$ $x : \iota^{\triangleright}$
$= : [o(\iota)](\iota)$ $x : \iota^{\triangleright}$
$\neg : o(o)$ $x = x : o$
$=(x) : o(\iota)$ $x : \iota^{\triangleright}$
$x \neq x : o$
$\neg : o(o)$ $x = x : o$
$\varepsilon : \iota(o(\iota))$ $(\lambda x)(x \neq x) : o(\iota)^{\triangleleft}$
$x \neq x : o$
$= : [o(\iota)](\iota)$ $(\varepsilon x)(x \neq x) : \iota$
$\varepsilon : \iota(o(\iota))$ $(\lambda x)(x \neq x) : o(\iota)^{\triangleleft}$
$=((\varepsilon x)(x \neq x)) : o(\iota)$
$(\varepsilon x)(x \neq x) : \iota$
$(\varepsilon x)(x \neq x) = (\varepsilon x)(x \neq x) : o$

Here, according to the definitions of ε and $=$ as constants above, $\varepsilon((\lambda x)(x \neq x))$ is denoted by $(\varepsilon x)(x \neq x)$ and $\neg(x = x)$ is denoted by $x \neq x$. The above proof tree proves that $(\varepsilon x)(x \neq x) = (\varepsilon x)(x \neq x)$ is typeable by o.

(3)

$$\begin{aligned}
[\![(\forall x)(x = x)]\!]_a^{(\mathfrak{M},g)} &= [\![\forall((\lambda x)(x = x)]\!]_a^{(\mathfrak{M},g)} \\
&= [\![\forall]\!]_a^{(\mathfrak{M},g)}([\![(\lambda x)(x = x)]\!]^{(\mathfrak{M},g)})_a \\
&= [\![\forall]\!]_a^{(\mathfrak{M},g)}(\xi \mapsto [\![= (x)(x)]\!]_{a[x\to\xi]}^{(\mathfrak{M},g)}) \\
&= [\![\forall]\!]_a^{(\mathfrak{M},g)}(\xi \mapsto [\![= (x)]\!]_{a[x\to\xi]}^{(\mathfrak{M},g)}(\xi)) \\
&= [\![\forall]\!]_a^{(\mathfrak{M},g)}(\xi \mapsto [\![=]\!]_{a[x\to\xi]}^{(\mathfrak{M},g)}(\xi)(\xi)) \\
&= [\![\forall]\!]_a^{(\mathfrak{M},g)}(\xi \mapsto \mathsf{T}) \\
&= \mathsf{T}
\end{aligned}$$

The second expression's semantic value is trivial:

$$\begin{aligned}
[\![(\varepsilon x)(x \neq x) = (\varepsilon x)(x \neq x)]\!]_a^{(\mathfrak{M},g)} &= [\![=]\!]_a^{(\mathfrak{M},g)}([\![(\varepsilon x)(x \neq x)]\!]_a^{(\mathfrak{M},g)})([\![(\varepsilon x)(x \neq x)]\!]_a^{(\mathfrak{M},g)}) \\
&= \mathsf{T}
\end{aligned}$$

below, we determine it:

$$\begin{aligned}
[\![(\varepsilon x)(x \neq x)]\!]_a^{(\mathfrak{M},g)} &= [\![\varepsilon((\lambda x)(x \neq x))]\!]_a^{(\mathfrak{M},g)} \\
&= [\![\varepsilon]\!]_a^{(\mathfrak{M},g)}([\![(\lambda x)(x \neq x)]\!]_a^{(\mathfrak{M},g)}) \\
&= [\![\varepsilon]\!]_a^{(\mathfrak{M},g)}(\xi \mapsto [\![x \neq x]\!]_{a[x\to\xi]}^{(\mathfrak{M},g)}) \\
&= g(\{\xi \in M \mid [\![x \neq x]\!]_{a[x\to\xi]}^{(\mathfrak{M},g)} = \mathsf{T}\}) = g(\emptyset) \\
&= g(\{\xi \in M \mid [\![\neg]\!]_{a[x\to\xi]}^{(\mathfrak{M},g)}([\![=]\!]_{a[x\to\xi]}^{(\mathfrak{M},g)}(\xi)(\xi)) = \mathsf{T}\}) = g(\emptyset)
\end{aligned}$$

□

2.5. Epsilon-invariant expressions

Definition 10. Let $N \in \mathrm{Exp}(\mathcal{L}_\lambda^{\forall\varepsilon})$ be such that for a context $(\Xi : \Gamma)$ the relation $(\Xi : \Gamma) \vdash N : \varphi$ holds for a type φ and let $\mathfrak{M}$ be a $\mathcal{L}_\lambda^{\forall}$ model. N is said to be *epsilon-invariant over the model* $\mathfrak{M}$, if for every assignment a of type $(\Xi : \Gamma)$ and choice functions $g_1, g_2 : \mathcal{P}(M) \to M$ it holds that

$$[\![N]\!]_a^{(\mathfrak{M},g_1)} = [\![N]\!]_a^{(\mathfrak{M},g_2)}.$$

The notion above is a symbolic formulation of the intuitive term "epsilon-independent". In FOL this concept was applied to show that "epsilon-independent" sentences can be reformulated into an epsilon-free one, provided the sentence is independent over *every* model (see Blass & Gurevich 2000).

3. Epsilon and application

Theorem 1. Let $P, Q \in \mathrm{Exp}(\mathcal{L}_\lambda^{\forall\varepsilon})$, $\mathfrak{M}$ be a model of $\mathcal{L}_\lambda^{\forall}$, $(\Xi : \Gamma)$ a context, $(\Xi : \Gamma) \vdash P : o$, $(\Xi : \Gamma) \vdash Q : o$ and $x \in \mathrm{Var}(\mathcal{L}_\lambda^{\forall\varepsilon})$, furthermore, let $[(\lambda x)P]((\varepsilon x)Q)$, P and Q be epsilon-invariant over the model $\mathfrak{M}$. Then for every assignment a of type $(\Xi : \Gamma)$ and choice function $g : \mathcal{P}(M) \to M$:

$$[\![[(\lambda x)P]((\varepsilon x)Q)]\!]_a^{(\mathfrak{M},g)} = [\![((\forall x)(\neg Q)\&(\forall x)P) \vee (((\exists x)Q)\&(\forall x)(Q \to P))]\!]_a^{(\mathfrak{M},g)}.$$

Proof. (1) Let the right hand side be T.
First case: $[\![((\forall x)(\neg Q)\&(\forall x)P)]\!]_a^{(\mathfrak{M},g)} = \mathsf{T}$. Then $[\![(\forall x)P)]\!]_a^{(\mathfrak{M},g)} = \mathsf{T}$ holds and let $m = [\![(\varepsilon x)Q]\!]_a^{(\mathfrak{M},g)} \in M$. Hence, by definition

$$\mathsf{T} = [\![(\forall x)P)]\!]_a^{(\mathfrak{M},g)} = [\![\forall((\lambda x)P)]\!]_a^{(\mathfrak{M},g)}$$

that is

$$[\![(\lambda x)P]\!]_a^{(\mathfrak{M},g)} = \left(\xi \mapsto [\![P]\!]_{a[x\to\xi]}^{(\mathfrak{M},g)}\right) \equiv \mathsf{T}.$$

Hence

$$[\![[(\lambda x)P]((\varepsilon x)Q)]\!]_a^{(\mathfrak{M},g)} = [\![[(\lambda x)P]]\!]_a^{(\mathfrak{M},g)}(m) = [\![P]\!]_{a[x\to m]}^{(\mathfrak{M},g)} = \mathsf{T}.$$

Second case: $[\![(((\exists x)Q)\&(\forall x)(Q \to P))]\!]_a^{(\mathfrak{M},g)} = \mathsf{T}$. Then

$$[\![(\lambda x)\neg Q]\!]_a^{(\mathfrak{M},g)} = \left(\xi \mapsto [\![\neg Q]\!]_{a[x\to\xi]}^{(\mathfrak{M},g)}\right) \not\equiv \mathsf{T}$$

hence for a $\xi \in M$ $[\![Q]\!]_{a[x\to\xi]}^{(\mathfrak{M},g)} = \mathsf{T}$. Therefore, if $[\![\varepsilon((\lambda x)Q)]\!]_a^{(\mathfrak{M},g)} = m$ then $[\![(\lambda x)Q]\!]_a^{(\mathfrak{M},g)}(m) = \mathsf{T}$. But from $[\![(\forall x)(Q \to P))]\!]_a^{(\mathfrak{M},g)} = \mathsf{T}$ it follows that $[\![P]\!]_{a[x\to m]}^{(\mathfrak{M},g)} = \mathsf{T}$, since $[\![Q]\!]_{a[x\to m]}^{(\mathfrak{M},g)} = \mathsf{T}$. Hence, $[\![(\lambda x)P]\!]_a^{(\mathfrak{M},g)}(m) = [\![[(\lambda x)P]((\varepsilon x)Q)]\!]_a^{(\mathfrak{M},g)} = \mathsf{T}$.

(2) Suppose the left hand side is T. First case: let $[\![((\forall x)(\neg Q)]\!]_a^{(\mathfrak{M},g)} = \mathsf{T}$, $m \in M$ arbitrary and g' is the choice function such that $g'(\emptyset) = m$. Hence, by the epsilon-invariance of P and $[(\lambda x)P]((\varepsilon x)Q)$ it follows that

$$\mathsf{T} = [\![[(\lambda x)P]((\varepsilon x)Q)]\!]_a^{(\mathfrak{M},g)} = [\![[(\lambda x)P]((\varepsilon x)Q)]\!]_a^{(\mathfrak{M},g')} = [\![P]\!]_{a[x\to m]}^{(\mathfrak{M},g')} = [\![P]\!]_{a[x\to m]}^{(\mathfrak{M},g)}$$

therefore $[\![(\forall x)P]\!]_a^{(\mathfrak{M},g)} = \mathsf{T}$. Second case: let $[\![((\exists x)Q]\!]_a^{(\mathfrak{M},g)} = \mathsf{T}$, $m \in M$ arbitrary such that $[\![Q]\!]_{a[x\to m]}^{(\mathfrak{M},g)} = \mathsf{T}$ and g' is the choice function such that $g'(\{\xi \in M \mid [\![Q]\!]_{a[x\to\xi]}^{(\mathfrak{M},g)} = \mathsf{T}\}) = m$. Then by the epsilon-invariance of P, Q and $[(\lambda x)P]((\varepsilon x)Q)$ it follows that

$$\mathsf{T} = [\![[(\lambda x)P]((\varepsilon x)Q)]\!]_a^{(\mathfrak{M},g)} = [\![[(\lambda x)P]((\varepsilon x)Q)]\!]_a^{(\mathfrak{M},g')} = [\![P]\!]_{a[x\to m]}^{(\mathfrak{M},g')} = [\![P]\!]_{a[x\to m]}^{(\mathfrak{M},g)}$$

for every m such that $[\![Q]\!]_{a[x\to m]}^{(\mathfrak{M},g)} = \mathsf{T}$. Hence, $[\![(\forall x)(Q \to P)]\!]_a^{(\mathfrak{M},g)} = \mathsf{T}$

$\square$

4. Morning Star and King of France tests

The concluding facts can be stated in two claims:

1. In the formal language $\mathcal{L}_{\lambda}^{\forall\varepsilon}$ (which is supposed to model the behaviour of descriptions) the (closed) term $(\varepsilon x)Q$ has *referential meaning* in the sense that a fixed model $(\mathfrak{M}, g)$ singles out an individual $[\![(\varepsilon x)Q]\!]^{(\mathfrak{M},g)} \in M$ for $(\varepsilon x)Q$ as semantic value.
2. In some cases, when $(\varepsilon x)Q$ is part of a compound sentence $[(\lambda x)P]((\varepsilon x)Q)$, with all its components being epsilon-invariant, the $(\varepsilon x)Q$ has a *contextual meaning,* such that the sentence $[(\lambda x)P]((\varepsilon x)Q)$ has an equivalent epsilon-free reformulation using quantified expressions from the plain language $\mathcal{L}_{\lambda}^{\forall}$.

We do not intend to set up a weaker theory than Russell's Theory of Descriptions. A new theory must serve at least as many solutions as far as Russell's proposal was able to solve. An appropriate indicator is to look at the two problems that the Theory of Descriptions solved and examine what the new model porposes. The first one is the problem of Hesperus and Phosphorus (below it will be called Morning Star Test), the second one is the problem of the empty names (the King of France Test).

4.1. Morning Star Test

In 1905, Russell gave a FOL-based solution of the so-called Frege Puzzle in terms of RTD, understandably, without mentioning the intensional tools of possible world semantics, which is a much later development. Here, I would like to show briefly that even the exposition of the puzzle is so widely criticized, that the RTD result of the test is rather irrelevant to us.

"Gottlob thinks that the Morning Star is illuminated by the Sun."
"The Evening Star is the Morning Star."
—
"Gottlob thinks that the Evening Star is illuminated by the Sun".
(Cf. Frege 1892/1990.)

First of all, I would like to point out that several scholars are committed to the assumption that the names such as *the Morning Star* or *the Evening Star* are understood tacitly as definite descriptions. For Russell, these names abbreviate descriptions, hence they are denoting phrases too (Dummett 1973, 97). The problem is that, according to Leiniz's Rule, since the Evening Star is the Morning Star, the two phrases are interchangeable. However, the above inference does not seem to be valid, since it is possible that Gottlob thinks that the Morning Star is illuminated by the Sun, but he does not necessarily know this fact about the Evening Star, even if in reality the two planets are the same, which is the case. Russell's solution was that the phrases *the Morning Star* and *the Evening Star* are not proper names, they only have contextual meanings, hence they are not interchangeable due to formal reasons.[10]

In the epsilon language $\mathcal{L}_{\lambda}^{\forall\varepsilon}$, the definite descriptions are proper names, they are manifested as epsilon terms on the object language level, hence the modelling in terms of the epsilon-language fails the Morning Star Test, and it does not explain the puzzle. Fortunately, hitherto, the Frege Puzzle and the semantic status of the expressions like *the Morning star* are not completely solved. If the phrase *the Morning star* is a rigid designator, as it is done in Kripke's proposal, then the Puzzle is solved. Here, temporarily, not having modal context, *rigid* means that the model designates a single individual in one step, and does not select first a set, then a member of it, by a choice function.[11] Then the puzzle only says that, if planet Venus is illuminated by the Sun, then planet Venus is illuminated by the Sun. According to Kripke's approach, the problematic case is the sentence *The Evening Star is the Morning Star*. It is a necessary truth, but it may be problematic from an epistemological point of view.[12] For the epsilon model, the solution is the same. According to Monk, the closed epsilon terms are constants, therefore they are rigid designators in accordance with the Kripke doctrine. However, as Fitting pointed out, an epsilon term, being description-like, can neither be a constant, nor a variable. It is a complex flexible designator (see Fitting 1972). Here, if *the Morning star* is a complex demonstrative (selected by a descriptive term in the actual world), then it is a rigid designator (see Kaplan 1989). Clearly, now, I do not have to deal with the modal context of epsilon terms, knowing that the highly applicable tool of demonstratives might make the modal approach much more complex, and might not add essentially more to the above consideration.

[10] See Russell (1905) and Whitehead & Russell (1910/1967).

[11] Of course, it is a rough simplification. Picking an individual means direct reference, rigid means the term has the same semantic value along the possible worlds. What is more, the notion of "rigid" above is understandable, but mathematically vague.

[12] See Kripke (1972, 102). The whole story can be found in Zvolenszky (2007).

4.2. The King of France Test

Consider the following two sentences

> "The present King of France is bald."
> "The present King of France is not bald."

In order to determine the truth value of the first one, let us imagine the set of all bald people. Since the present King of France is not in this set, the first sentence is false. But, the same reasoning leads to the fact that the second sentence is false too. Which is a contradiction. Hence, the phrase *the present King of France* is not a proper name, it cannot have a meaning in isolation, rather it only has a contextual meaning and the sentences containing such phrases are quantified formulas. This is Russell's solution. In the epsilon calculus the semantic values of the epsilon terms are defined in all cases. The two sentences above are unproblematic, they assign to the phrase *the present King of France* an existing individual as reference. And it is either bald or not bald. According to Theorem 1 of the present paper, sentences *may* possess contextual meaning too, where the truth value is also well-defined. Of course, the reference of *the present King of France* in the epsilon calculus is not the present King of France. Approaching the situation on a more formal level, let us consider the symbolic sentence

$$(\varepsilon x)(x \neq x) = (\varepsilon x)(x \neq x)$$

This is a sentence containing terms which are ill-defined as descriptions: $x \neq x$ is an empty predicate. However, the semantic value of $(\varepsilon x)(x \neq x)$, in a given model, is well-defined. Moreover, $(\varepsilon x)(x \neq x) = (\varepsilon x)(x \neq x)$ is an epsilon invariant sentence, since, it is true in any given epsilon semantics. And indeed, there are epsilon semantics (for example the Bourbaki group's formal systems), where $(\varepsilon x)(x \neq x) = (\varepsilon x)(x \neq x)$ is syntactically identical to the sentence $(\forall x)(x = x)$. $(\varepsilon x)(x \neq x) = (\varepsilon x)(x \neq x)$ is an epsilon-invariant sentence, which has contextual meaning too: it is equivalent to the fact that every individual is identical to itself.

The situation is very similar to the problem of the interesting-looking man holding a martini. In this case, the *the present King of France* is rather a person who is, in fact, bald, but not the present King of France, and $(\varepsilon x)(x \neq x)$ is an existing individual, which is identical to itself, but of course, it does not hold that it is not the same as itself.

Acknowledgements

I would like to thank Endre Latabár, András Simon and an anonymous reviewer for their helpful comments concerning specific parts of the paper or the whole one.

References

Ahrendt, W. and M. Giese. 1999. Hilbert's epsilon-terms in automated theorem proving. In N. V. Murray (ed.) Automated Reasoning with Analytic Tableaux and Related Methods, International Conference, TABLEAUX '99. 171–185.

Blass, A. and Y. Gurevich. 2000. The logic of choice. The Journal of Symbolic Logic 65. 1264–1310.

Caicedo, X. 1995. Hilbert's ε symbol in the presence of generalized quantifiers. In M. Krynicki, M. Mostowski and L. W. Szczerba (eds.) Quantifiers: Logics, models and computation. Volume two: Contributions. Berlin & New York: Springer. 63–78.

Donnellan, K. 1966. Reference and definite descriptions. Philosophical Review 75. 281–304.

Dummett, M. 1973. Frege: Philosophy of language. London: Duckworth.

Fitting, M. 1972. An epsilon-calculus system for first order S4. In W. Hodges (ed.) Conference in Mathematical Logic – London '70. Berlin & New York: Springer. 103–110.

Frege, G. 1892/1990. On sense and reference. In P. Geach and M. Black (eds.) Translations from the philosophical writings of Gottlob Frege. Oxford: Basil Blackwell. 56–78.

Kaplan, D. 1989. Demonstratives. In J. Almog, J. Perry and H. Wettstein (eds.) Themes from Kaplan. Oxford: Oxford University Press. 481–563.

Kneebone, G. T. 1963. Mathematical logic and the foundation of mathematics. London: Van Nostrand.

Kripke, S. A. 1972. Naming and necessity. In D. Davidson and G. Harman (eds.) Semantics of natural language. Dordrecht: Reidel. 251–355.

Mints, G. 1996. Thoralf Skolem and the epsilon substitution method for predicate logic. Nordic Journal of Philosophical Logic 1. 133–146.

Molnár, Z. 2013. Epsilon-invariant substitutions and indefinite descriptions. Logic Journal of the IGPL 21. 812–829.

Monk, J. D. 1976. Mathematical logic. Berlin & New York: Springer.

Russell, B. 1905. On denoting. Mind 14. 479–493. (In: B. Russell: Logic and knowledge. London: Routledge. 39–56).

Simmons, H. 2000. Derivation and computation: Taking the Curry–Howard correspondence seriously. Cambridge: Cambridge University Press.

Slater, H. 2007. Completing Russell's logic. Journal of Bertrand Russell Studies 27. Article 15.

Slater, H. 2009. Hilbert's epsilon calculus and its successors. In D. M. Gabbay and J. Woods (eds.) Handbook of the history of logic. Logic from Russell to Church, Vol. 5. Amsterdam: North-Holland. 385–448.

Sørensen, M. H. B. and P. Urzyczyn. 1998. Lectures on the Curry–Howard Isomorphism. Ms. http://disi.unitn.it/~bernardi/RSISE11/Papers/curry-howard.pdf.

Tarski, A. 1956. Logic semantics, metamathematics papers from 1923 to 1938. Oxford: Clarendon Press.

Troelstra, A. S. and H. Schwichtenberg. 2000. Basic proof theory (Second edition). Cambridge: Cambridge University Press.

Whitehead, A. N. and B. Russell. 1910/1967. Incomplete symbols: Descriptions. In J. van Heijenoort (ed.) From Frege to Gödel: A source book in mathematical logic 1879–1931. Cambridge, MA: Harvard University Press. 216–223.

Zvolenszky, Zs. 2007. Modality, names and descriptions. Doctoral dissertation. New York University.