Key Metalogical Propositions on a Variant of Hilbert's Epsilon-Calculus

Domenico Cantone[1], Marianna Nicolosi-Asmundo[1,*] and Eugenio G. Omodeo[2]

[1]Dept. of Mathematics and Computer Science, University of Catania, Italy
[2]Dept. of Mathematics, Informatics and Geosciences, University of Trieste, Italy

Abstract

Hilbert's ε-operator, a foundational device for forming indefinite descriptions, has long been overshadowed by standard quantifiers in first-order logic. However, its capacity to eliminate quantifiers and reframe logical derivations makes it a compelling tool for alternative proof strategies and automated reasoning. This paper revisits the ε-calculus, offering a streamlined proof of completeness adapted from Hasenjaeger's 1953 approach. Building on earlier work by Leisenring, Davis, and Fechter, we present a variant of the ε-calculus that omits all predicate symbols aside from equality. The development follows the conventional structure of logical systems—syntax, semantics, and deductive calculus—culminating in a soundness and completeness result. The aim is to reaffirm the ε-operator's relevance in the foundations of logic through a simplified and accessible formal treatment.
MS Classification 2020: 03B10, 68V15

Keywords

Free variable FOL calculus, Hilbert's epsilon-symbol, Skolem completion

Introduction

Hilbert's ε-symbol, despite the key role it played in the historical Hilbert–Bernays treatise [1] on the foundations of mathematics and in other classical monographs, never achieved the widespread adoption enjoyed by the existential and universal quantifiers in first-order logic.

It is used to form indefinite descriptions of the form $\varepsilon\, x.\varphi$ (read: "one x such that φ"), where φ is a formula and x a variable. Each such expression is a term denoting an entity—if any exists—that satisfies φ; when no such entity exists, nonetheless $\varepsilon\, x.\varphi$ designates some element of the domain of discourse. As shown, e.g., in [2], availability of ε-descriptions allows quantifiers to be entirely eliminated from first-order logic. This enables a different approach to logical derivations and the development of tools for automated deduction framed in less conventional terms—potentially yielding better performance in certain contexts. Moreover, the use of ε-descriptions in formal proof systems such as free-variable semantic tableaux may enhance reasoning performance by enabling the construction of shorter proofs [3, 4, 5]. These results largely depend on the syntactic structure of ε-terms, which differs significantly from that of terms generated via standard Skolemization techniques. Taken together, these considerations motivate the authors to bring the ε-operator back into focus.

The completeness proof for first-order predicate logic has progressively simplified from Gödel's original (ca. 1930) to Leon Henkin's (ca. 1949) and later to Gisbert Hasenjaeger's (1953). We adapt the third of these to Hilbert's ε-calculus, further streamlining the path to this fundamental metalogical result. Our approach to the ε-calculus builds on the work of Albert Leisenring [6], and of Martin Davis and Ronald Fechter [2], but, without loss of generality, sets aside predicate symbols other than equality.

The material is organized according to the standard structure of most logical formalisms: syntax (Sec. 1), semantics (Sec. 2), calculus (understood as a system of deduction, Sec. 3), and soundness and

CILC 2025: 40th Italian Conference on Computational Logic, June 25–27, 2025, Alghero, Italy
*Corresponding author.
domenico.cantone@unict.it (D. Cantone); marianna.nicolosiasmundo@unict.it (M. Nicolosi-Asmundo); eomodeo@units.it (E. G. Omodeo)
https://www.dmi.unict.it/cantone (D. Cantone); https://sites.units.it/eomodeo/index.html/ (E. G. Omodeo)
0000-0002-1306-1166 (D. Cantone); 0000-0003-4456-5110 (M. Nicolosi-Asmundo); 0000-0003-3917-1942 (E. G. Omodeo)

completeness of the proposed calculus (Sec. 4). The concluding Sec. 5 offers a commentary on the work presented.

1. Cumulative Signature of a First-order Logic Language

A first-order language $\mathcal{L}(\Lambda)$ is identified by a signature

$$\Lambda \;=\; (\mathcal{F}, \mathbf{d}), \quad \text{where} \quad \mathbf{d} : \mathcal{F} \longrightarrow \mathbb{N}\,,$$

consisting of a set $\mathcal{F}$ of symbols called *functors* paired with a function $\mathbf{d}$, called *degree*, sending each functor to its expected number of arguments—a nonnegative integer.

The other primitive symbols of $\mathcal{L}(\Lambda)$ are: the *equality* relator '$=$', the denumerable infinity $\nu_0, \nu_1, \ldots$ of (free) individual *variables*, the propositional *connectives* '$\mathtt{f}$', '$\rightarrow$', and the punctuation marks ')', '(', ','.

1.1. Terms, formulae, and uniform substitutions

The *terms* of $\mathcal{L}(\Lambda)$ are strings of symbols from $\mathcal{L}(\Lambda)$, defined recursively as follows:

1. Each individual variable ν_i is a term.
2. A string of the form $g(\tau_1, \ldots, \tau_{\mathbf{d}(g)})$, where $g \in \mathcal{F}$ and $\tau_1, \ldots, \tau_{\mathbf{d}(g)}$ are terms, is also a term. (Here, parentheses are omitted when g is a *constant*, that is, when $\mathbf{d}(g) = 0$.)

The *formulae* of $\mathcal{L}(\Lambda)$ are strings of symbols from $\mathcal{L}(\Lambda)$, defined recursively as follows:

Atomic formulae are:
- Strings of the form $\tau_1 = \tau_2$, where τ_1 and τ_2 are terms of $\mathcal{L}(\Lambda)$.
- The propositional constant $\mathtt{f}$.

Compound formulae are of the form $(\varphi \rightarrow \psi)$, where φ and ψ are formulae, atomic or compound. (Quite often, the outermost parentheses of a compound formula will be dropped.)

The familiar *propositional connectives* $\neg$, $\vee$, $\&$, and $\leftrightarrow$, as well as the inequality relator $\neq$, are treated here as derived constructs:

$$\begin{array}{lll} (\neg\,\varphi) \;:=\; (\varphi \rightarrow \mathtt{f}), & (\varphi \vee \psi) \;:=\; ((\neg\,\varphi) \rightarrow \psi), & (\varphi \,\&\, \psi) \;:=\; \neg\,(\varphi \rightarrow (\neg\,\psi)), \\ \tau_1 \neq \tau_2 \;:=\; \neg\,(\tau_1 = \tau_2), & (\varphi \leftrightarrow \psi) \;:=\; ((\varphi \rightarrow \psi) \,\&\, (\psi \rightarrow \varphi)). & \end{array}$$

Later, we will introduce quantifiers by means of somewhat more engaging abbreviations.

Definition 1.1 (**n-adic formula**)**.** A formula φ of $\mathcal{L}(\Lambda)$ is said to be n*-adic*, where $n \in \mathbb{N}$, if the only variables occurring in φ are $\nu_0, \nu_1, \nu_2, \ldots, \nu_n$ (namely, the first $n+1$ variables in the standard list), with the only possible exception of ν_0.

Then, $\varphi(\sigma_0, \sigma_1, \ldots, \sigma_n)$ denotes the formula obtained from φ by simultaneously replacing each ν_i with the corresponding term σ_i. $\dashv$

1.2. Key formulae, Skolem completion, descriptors, and quantifiers

Definition 1.2 (Key formula). An n-adic formula χ of $\mathcal{L}(\Lambda)$ is called a *key formula* if:

1. Every term appearing in χ, if not a variable, contains at least one occurrence of ν_0.
2. Each of the variables $\nu_1, \ldots, \nu_n$ occurs in χ exactly once.
3. The variables $\nu_1, \ldots, \nu_n$ appear in χ from left to right in exactly that order. $\dashv$

Example 1.1. Set $\chi_0 := \nu_0 = \nu_0$ and $\chi_{i+1} := (\chi_i \,\&\, \nu_0 \neq \nu_{i+1})$. Then χ_n is a key formula for each n. $\dashv$

Theorem 1.1 (From [2], pp.435–436). *Let x be a variable and φ a formula of $\mathcal{L}(\Lambda)$. Then there exist unique: an integer $n \geqslant 0$, an n-adic key formula $\varphi^{[x]}$, and terms $\tau_1, \ldots, \tau_n$ in which x does not occur, such that*

$$\varphi \equiv \varphi^{[x]}(x, \tau_1, \ldots, \tau_n)\,,$$

meaning that φ and $\varphi^{[x]}(x, \tau_1, \ldots, \tau_n)$ are syntactically identical,[1] *and such that n is the least integer for which such a decomposition exists.*

Proof. The following algorithm determines (in the only possible way) the sought $n, \tau_1, \ldots, \tau_n$, and $\varphi^{[x]}$.

Initially, place a cursor j immediately before the first symbol of the first occurrence of a term in φ, and set $n \coloneqq 0$.

As long as, in some occurrence after j within φ, there is a term (possibly equal to previous ones) that does not involve x, proceed as follows:

- Locate the first occurrence (after j) of such a term τ in φ.
- Set $n \coloneqq n + 1$ and record $\tau_n \coloneqq \tau$ along with the position π_n of this occurrence.
- Advance the cursor j to the end of this occurrence of τ in φ.

Obtain $\varphi^{[x]}$ from φ by simultaneously:

- replacing x with ν_0,
- and, for each $i = 1, \ldots, n$, replacing, at position π_i, the occurrence of τ_i with the variable ν_i.

This yields the desired decomposition, where n is minimal. Moreover, the triple $\big(n, \varphi^{[x]}, (\tau_1, \ldots, \tau_n)\big)$ is uniquely determined by φ and x, due to the syntactic nature of the construction. $\square$

Example 1.2. Given $\varphi \equiv \big(x = f(x, x) \to f(z, y) = g(y, z)\big)$, where x, y, z stand for distinct variables and f, g are functors of degree 2, the algorithm just seen produces

$$\begin{aligned}
&\varphi^{[x]} \equiv \big(\nu_0 = f(\nu_0, \nu_0) \;\to\; \nu_1 = \nu_2\big), && \text{so that} \quad \varphi \equiv \varphi^{[x]}\big(x, f(z, y), g(y, z)\big),\\
&\varphi^{[y]} \equiv \big(\nu_1 = \nu_2 \;\to\; f(\nu_3, \nu_0) = g(\nu_0, \nu_4)\big), && \text{so that} \quad \varphi \equiv \varphi^{[y]}\big(y, x, f(x, x), z, z\big),\\
&\varphi^{[z]} \equiv \big(\nu_1 = \nu_2 \;\to\; f(\nu_0, \nu_3) = g(\nu_4, \nu_0)\big), && \text{so that} \quad \varphi \equiv \varphi^{[z]}\big(z, x, f(x, x), y, y\big), \quad \text{and}\\
&\varphi^{[w]} \equiv \big(\nu_1 = \nu_2 \;\to\; \nu_3 = \nu_4\big), && \text{so that} \quad \varphi \equiv \varphi^{[w]}\big(w, x, f(x, x), f(z, y), g(y, z)\big),
\end{aligned}$$

where w is any variable distinct from x, y, z. Thus, $\varphi^{[x]}$ is a 2-adic key formula, while $\varphi^{[y]}$, $\varphi^{[z]}$, and $\varphi^{[w]}$ are 4-adic formulae. Note also that, for example, $\varphi^{[y]}$ is obtained from φ by replacing both occurrences of y with ν_0, the first occurrence of x with ν_1, the term $f(x, x)$ with ν_2, and the first and second occurrences of z by ν_3 and ν_4, respectively. This clearly shows that the construction of key formulae is based on the positions of terms within the syntactic structure of φ, as shown in the proof of Thm. 1.1. $\dashv$

Definition 1.3. Let φ be a formula, x a variable, and σ a term. We denote by φ^x_σ the formula obtained from φ by simultaneously replacing all occurrences of x with σ. $\dashv$

Remark 1. The identity $\varphi^x_\sigma \equiv \varphi^{[x]}(\sigma, \tau_1, \ldots, \tau_n)$ is clear—where $\tau_1, \ldots, \tau_n$ are as described above. $\dashv$

Remark 2. Let $\chi_1 \equiv \varphi^{[x]}$ and $\chi_2 \equiv \psi^{[x]}$ be an n-adic key formula and an m-adic key formula, respectively resulting, in the manner described above, from x, φ and from x, ψ . Then we have the syntactical identities $(\neg\, \varphi)^{[x]} \;\equiv\; \neg\,\big(\varphi^{[x]}\big)$ and $(\varphi \to \psi)^{[x]} \;\equiv\; \big(\varphi^{[x]}(\nu_0, \nu_1, \ldots, \nu_n) \to \psi^{[x]}(\nu_0, \nu_{n+1}, \ldots, \nu_{n+m})\big)$. Moreover, $(\varphi \leftrightarrow \psi)^{[x]}$ is a $(2\,n + 2\,m)$-adic key formula χ such that the syntactical identity

$$\chi(\sigma, \tau_1, \ldots, \tau_n, \sigma_1, \ldots, \sigma_m, \sigma_1, \ldots, \sigma_m, \tau_1, \ldots, \tau_n) \;\equiv\; \big(\varphi^{[x]}(\sigma, \tau_1, \ldots, \tau_n) \;\leftrightarrow\; \psi^{[x]}(\sigma, \sigma_1, \ldots, \sigma_m)\big),$$

holds for every $(1 + n + m)$-tuple $(\sigma, \tau_1, \ldots, \tau_n, \sigma_1, \ldots, \sigma_m)$ of terms. $\dashv$

[1]'Syntactical identity' refers to the condition of being exactly the same in syntactic form—that is, the symbols and their arrangement are identical. However, to claim that $\varphi \equiv \psi$, one must first remove any derived constructs (i.e., shortcuts, or abbreviations defined in terms of more basic symbols) by expanding both φ and ψ into their full definitions.

Skolem expansions and completion of a signature. Our next goal is to complete, in a minimal way, any initial signature Λ_0 with functors $\mathcal{F}_0$ to a signature Λ_∞ with functors $\mathcal{F}_\infty$, where $\mathcal{F}_0 \subseteq \mathcal{F}_\infty$, so that every key formula χ of $\mathcal{L}(\Lambda_\infty)$ is associated, in an injective manner, with a new functor h_χ of the same degree as χ. We achieve this by defining an increasing sequence of signatures $(\Lambda_\ell)_{\ell\in\mathbb{N}}$, where each $\Lambda_{\ell+1}$ is obtained from Λ_ℓ as follows:

1. For every n-adic key formula χ of $\mathcal{L}(\Lambda_\ell)$ that does not belong to $\mathcal{L}(\Lambda_{\ell-1})$ (if $\ell > 0$), associate a new functor h_χ of degree $\mathbf{d}(h_\chi) = n$. Each such functor is called a *Skolem functor*.
2. Define $\mathcal{F}_{\ell+1}$ by adding these Skolem functors h_χ to $\mathcal{F}_\ell$, and extend $\mathbf{d}$ accordingly.

Finally, let the *Skolem completion* Λ_∞ of the initial signature Λ_0 be the signature whose set of functors $\mathcal{F}_\infty$ is given by

$$\mathcal{F}_\infty \quad \coloneqq \quad \bigcup_{\ell\in\mathbb{N}} \mathcal{F}_\ell\,. \qquad \dashv$$

Remark 3. Based on Example 1.1, we observe that the expansion of Λ_0 to Λ_1 introduces infinitely many new functors. A similar construction, replacing each variable ν_{i+1} with a term of the form $h_\chi(\nu_{i+1})$, shows that $\Lambda_{\ell+2}$ is endowed with an infinitely richer collection of functors than $\Lambda_{\ell+1}$. $\dashv$

Descriptors and quantifiers. We will now introduce Hilbert's *descriptors* $\boldsymbol{\varepsilon}\,x$, Peano's descriptors $\boldsymbol{\imath}\,x$, and the usual *quantifiers* $\exists\,x$ and $\forall\,x$, where x is an individual variable. These can be read, respectively, as: "an x such that", "the x such that", "there is an x such that", and "for every x, it holds that".

Consider a formula φ of $\mathcal{L}(\Lambda_\infty)$. Moreover, let χ be the key formula $\varphi^{[x]}$ determined—along with n and $\tau_1, \ldots, \tau_n$—as in Thm. 1.1. We introduce the said derived constructs through the following abbreviating rules:

$$\begin{array}{rcll}
\boldsymbol{\varepsilon}\,x.\varphi & \coloneqq & h_\chi(\tau_1,\ldots,\tau_n) & \text{and, accordingly, } \ \boldsymbol{\varepsilon}\,x.(\neg\,\varphi) \ \coloneqq \ h_{\neg\,\chi}(\tau_1,\ldots,\tau_n); \\
(\exists\,x.\,\varphi) & \coloneqq & \varphi^x_{\boldsymbol{\varepsilon}\,x.\varphi}\,, & \text{i.e., } \ (\exists\,x.\,\varphi) \ \coloneqq \ \chi\big(h_\chi(\tau_1,\ldots,\tau_n),\tau_1,\ldots,\tau_n\big); \\
(\forall\,x.\,\varphi) & \coloneqq & \varphi^x_{\boldsymbol{\varepsilon}\,x.(\neg\,\varphi)}\,, & \text{i.e., } \ (\forall\,x.\,\varphi) \ \coloneqq \ \chi\big(h_{\neg\chi}(\tau_1,\ldots,\tau_n),\tau_1,\ldots,\tau_n\big); \\
\boldsymbol{\imath}\,x.\varphi & \coloneqq & \boldsymbol{\varepsilon}\,y.\big(\forall\,x.(\varphi \rightarrow x = y)\big), & \text{where } y \text{ is a variable distinct from } x. \qquad \dashv
\end{array}$$

The formulae α of $\mathcal{L}(\Lambda_\infty)$ in which no variables occur are called *sentences*. If a given formula α involves descriptors or quantifiers, it is unnecessary to unravel these constructs to establish that it is a sentence. It suffices to check that each apparent occurrence of a variable x within it appears within a descriptor $\boldsymbol{\varepsilon}\,x.\varphi$ or $\boldsymbol{\imath}\,x.\varphi$, or within a subformula of the form $(\exists\,x.\,\varphi)$ or $(\forall\,x.\,\varphi)$. In fact, applying a quantifier $\exists\,x$ or $\forall\,x$ to a formula φ in which x occurs decreases the number of variables by 1. This holds because, despite appearances, x occurs neither in $\boldsymbol{\varepsilon}\,x.\,\varphi$ nor in $\boldsymbol{\varepsilon}\,x.\big(\neg\,\varphi\big)$.

2. Interpreting Structures and Their Augmentations

The formulae of a language $\mathcal{L}(\Lambda)$ are interpreted using a *structure* $\mathfrak{I} = (\mathfrak{D}, \mathfrak{F})$, whose

- *domain of discourse*, $\mathfrak{D}$, is a set consisting of at least two so-called *individuals*; and whose
- *functor interpretation*, $\mathfrak{F}$, belongs to $\prod_{g\in\mathcal{F}} \mathfrak{D}^{\mathfrak{D}^{\mathbf{d}(g)}}$. This means that, for every functor g,

  $$\mathfrak{F}(g) \colon \mathfrak{D}^{\mathbf{d}(g)} \longrightarrow \mathfrak{D}$$

  is a function sending each $\mathbf{d}(g)$-tuple of individuals to an individual, for every functor g.

We now provide recursive rules for evaluating the terms and formulae of $\mathcal{L}(\Lambda)$ in such a structure. Along with $\mathfrak{I}$, since terms may involve variables ν_i about which the structure itself says nothing, we must also consider a denumerable sequence $\mathfrak{N} = (\boldsymbol{n}_0, \boldsymbol{n}_1, \dots)$ of individuals drawn from $\mathfrak{D}$, called a *(variable) assignment*. This assignment associates to each variable ν_i the individual $\boldsymbol{n}_i$ in $\mathfrak{D}$. As a preliminar, we introduce the following definition.

Definition 2.1 (Truth-value assignment). Denote by $\mathbf{f}, \mathbf{t}$ falsehood and truth, respectively. A *truth value assignment* for $\mathcal{L}(\Lambda)$ is a function $\mathfrak{T}\colon \;\{\text{formulae of } \mathcal{L}(\Lambda)\} \;\longrightarrow\; \{\mathbf{f}, \mathbf{t}\}$ such that

- $\mathfrak{T}(\mathtt{f}) \;=\; \mathbf{f}$;
- $\mathfrak{T}\big((\varphi \to \psi)\big) \;=\; \textbf{if } \mathfrak{T}(\varphi) = \mathbf{t} \textbf{ then } \mathfrak{T}(\psi) \textbf{ else } \mathbf{t}\,,$ for all formulae φ, ψ. ⊣

Definition 2.2 (Evaluation rules). Let $\mathfrak{I} = (\mathfrak{D}, \mathfrak{F})$ be a structure and $\mathfrak{N}$ an assignment. The evaluation function $\mathsf{val}_{\mathfrak{I},\mathfrak{N}}(\cdot)$ assigns values to syntactic expressions of $\mathcal{L}(\Lambda)$ as follows:

(a) Terms.

- $\mathsf{val}_{\mathfrak{I},\mathfrak{N}}(\nu_i) \coloneqq \boldsymbol{n}_i$, for each variable ν_i;
- $\mathsf{val}_{\mathfrak{I},\mathfrak{N}}\big(g(\tau_1, \dots, \tau_{\mathbf{d}(g)})\big) \coloneqq \mathfrak{F}(g)\big(\mathsf{val}_{\mathfrak{I},\mathfrak{N}}(\tau_1)\,, \dots, \mathsf{val}_{\mathfrak{I},\mathfrak{N}}(\tau_{\mathbf{d}(g)})\,\big)$,
  for each functor g and every $\mathbf{d}(g)$-tuple of terms $(\tau_1, \dots, \tau_{\mathbf{d}(g)})$.

(b) Atomic formulae.

- $\mathsf{val}_{\mathfrak{I},\mathfrak{N}}(\mathtt{f}) \coloneqq \mathbf{f}$;
- $\mathsf{val}_{\mathfrak{I},\mathfrak{N}}(\tau = \sigma) \coloneqq \textbf{if } \mathsf{val}_{\mathfrak{I},\mathfrak{N}}(\tau) = \mathsf{val}_{\mathfrak{I},\mathfrak{N}}(\sigma) \textbf{ then } \mathbf{t} \textbf{ else } \mathbf{f}$, for all terms τ, σ.

(c) Compound formulae.
The evaluation function is extended to compound formulae by structural induction, following the semantics of truth-value assignments given in Definition 2.1. Specifically, we define:

$$\mathsf{val}_{\mathfrak{I},\mathfrak{N}}((\varphi \to \psi)) \coloneqq \textbf{if } \mathsf{val}_{\mathfrak{I},\mathfrak{N}}(\varphi) = \mathbf{t} \textbf{ then } \mathsf{val}_{\mathfrak{I},\mathfrak{N}}(\psi) \textbf{ else } \mathbf{t},$$

for each compound formula of $\mathcal{L}(\Lambda)$ of the form $(\varphi \to \psi)$, where φ and ψ are formulae (atomic or compound). ⊣

Remark 4. The evaluation function $\mathsf{val}_{\mathfrak{I},\mathfrak{N}}(\cdot)$ defined above is total on the syntactic expressions of $\mathcal{L}(\Lambda)$, returning domain elements for terms and truth values for formulae. Its restriction to formulae yields a truth-value assignment in the sense of Definition 2.1. ⊣

The fact that $(\mathfrak{I}, \mathfrak{N})$ *models* a formula φ, in the sense that $\mathsf{val}_{\mathfrak{I},\mathfrak{N}}(\varphi) = \mathbf{t}$, is also denoted by

$$(\mathfrak{I}, \mathfrak{N}) \;\models\; \varphi.$$

Moreover, if $\mathcal{A} \cup \{\alpha\}$ is a collection of formulae, the notation $\mathcal{A} \models \alpha$ (read: "α is a *logical consequence* of $\mathcal{A}$") indicates that $(\mathfrak{I}, \mathfrak{N}) \;\models\; \alpha$ holds for all pairs $(\mathfrak{I}, \mathfrak{N})$ that model each formula φ in $\mathcal{A}$. We say that α is *valid*, written as

$$\models \alpha,$$

if $(\mathfrak{I}, \mathfrak{N}) \;\models\; \alpha$ holds for all pairs $(\mathfrak{I}, \mathfrak{N})$. Clearly, this is the case if $\mathfrak{T}(\alpha) = \mathbf{t}$ holds in every truth-value assignment $\mathfrak{T}$; in the latter case, α is called a *tautology* .

Let $\mathcal{L}(\Lambda)$ be a language interpreted by a structure $\mathfrak{I} = (\mathfrak{D}, \mathfrak{F})$, and let φ be any formula of $\mathcal{L}(\Lambda)$. Also, let n be an integer that bounds the indices of all variables occurring in φ, that is, every ν_i in φ satisfies $i \leqslant n$. Then, for all $\mathfrak{D}$-valued assignments $\mathfrak{N} = (\boldsymbol{n}_0, \boldsymbol{n}_1, \dots)$ and $\mathfrak{N}' = (\boldsymbol{n}'_0, \boldsymbol{n}'_1, \dots)$ such that $\boldsymbol{n}_i = \boldsymbol{n}'_i$ for $i = 0, 1, \dots, n$, we have:

$$(\mathfrak{I}, \mathfrak{N}) \models \varphi \quad\Longleftrightarrow\quad (\mathfrak{I}, \mathfrak{N}') \models \varphi.$$

(In particular, if φ is a sentence, then its evaluation is independent of any $\mathfrak{D}$-valued assignment.)

This observation justifies the following definition.

Definition 2.3 (Satisfaction by a tuple). Let $\mathcal{L}(\Lambda)$ be a language interpreted by a structure $\mathfrak{I} = (\mathfrak{D}, \mathfrak{F})$, and let φ be any formula of $\mathcal{L}(\Lambda)$. Let n be an integer that bounds the indices of all variables occurring in φ. For any $(n{+}1)$-tuple $\vec{a} \coloneqq (\boldsymbol{a}_0, \boldsymbol{a}_1, \dots, \boldsymbol{a}_n)$ in $\mathfrak{D}^{n+1}$, we write

$$(\mathfrak{I}, \vec{a}) \models \varphi$$

and say that $\vec{a}$ *satisfies* φ *in* $\mathfrak{I}$ to mean that for every assignment $\mathfrak{N} = (\boldsymbol{n}_0, \boldsymbol{n}_1, \dots)$ over $\mathfrak{D}$ such that $\boldsymbol{n}_i = \boldsymbol{a}_i$ for all $i = 0, \dots, n$, it holds that $(\mathfrak{I}, \mathfrak{N}) \models \varphi$. ⊣

Extending the evaluation rules canonically to the Skolem completion $\mathcal{L}(\Lambda_\infty)$ of $\mathcal{L}(\Lambda_0)$ requires an enrichment of the structure $\mathfrak{I}$, as explained next.

2.1. Selective structures

As a preliminary step, we equip any structure $\mathfrak{I} = (\mathfrak{D}, \mathfrak{F})$ with a function

$$\mathfrak{c} \colon \mathscr{P}(\mathfrak{D}) \quad \longrightarrow \quad \mathfrak{D}\,,$$

where $\mathscr{P}(\mathfrak{D})$ denotes the power set of $\mathfrak{D}$, such that:

- for every nonempty subset $S \subseteq \mathfrak{D}$, we have $\mathfrak{c}(S) \in S$;
- and $\mathfrak{c}(\emptyset) \neq \mathfrak{c}(\mathfrak{D})$.

This can be done by a plain application of the Axiom of Choice. The second condition, while arbitrary, ensures that $\mathfrak{c}$ distinguishes between the empty set and the entire domain—this will be useful in what follows. We call the resulting triple $\mathfrak{I}^{(\mathfrak{c})} = (\mathfrak{D}, \mathfrak{F}, \mathfrak{c})$ a *selective structure*, and refer to $\mathfrak{c}(\cdot)$ as the associated *selection function*.

Remark 5 (Why do we need no relator other than equality?). Renouncing structures whose domain of discourse has cardinality 1, as we have done, entails no drawbacks; on the contrary, it offers an advantage: if we define

$$\mathfrak{t} \coloneqq \boldsymbol{\varepsilon}\, x .\, x = x \quad \text{and} \quad \mathfrak{f} \coloneqq \boldsymbol{\varepsilon}\, x .\, x \neq x\,,$$

our semantics will ensure that $\mathfrak{t} \neq \mathfrak{f}$.[2] After that, we can surrogate every relation symbol R, except for equality, by a functor g_R of the same degree a as R, constrained to satisfy

$$\Big(\forall\, x_1 . \cdots \Big(\forall x_a . \big(g_R(x_1 \, \dots, x_a) = \mathfrak{t} \,\vee\, g_R(x_1 \, \dots, x_a) = \mathfrak{f}\big)\Big) \cdots \Big)\,.$$

Thus, we can use $g_R(\tau_1, \dots, \tau_a) = \mathfrak{t}$ instead of the atomic formula $R(\tau_1, \dots, \tau_a)$. ⊣

2.2. Completing an interpretation using a selection function

Next we expand step by step a selective structure $\mathfrak{I}_0^{(\mathfrak{c})} \coloneqq (\mathfrak{D}, \mathfrak{F}_0, \mathfrak{c})$, based on our initial signature $\Lambda_0 \coloneqq \Lambda$, to its Skolem completion Λ_∞. For each integer $\ell \geqslant 0$, obtain $\mathfrak{I}_{\ell+1}^{(\mathfrak{c})} \coloneqq (\mathfrak{D}, \mathfrak{F}_{\ell+1}, \mathfrak{c})$ from $\mathfrak{I}_\ell^{(\mathfrak{c})}$ as described below.

Embed $\mathfrak{F}_\ell$ in $\mathfrak{F}_{\ell+1}$, i.e., put $\mathfrak{F}_{\ell+1}(g) \coloneqq \mathfrak{F}_\ell(g)$ for every functor g in $\mathcal{F}_\ell$. Then, for every functor h_χ in $\mathcal{F}_{\ell+1} \setminus \mathcal{F}_\ell$:

- Determine the number n such that the key formula χ is n-adic (and hence $\mathbf{d}(h_\chi) = n$).
- For each n-tuple $\vec{a} = (\boldsymbol{a}_1, \dots, \boldsymbol{a}_n)$ in $\mathfrak{D}^n$, consider the (possibly empty) set $S_\chi^{\vec{a}}$ of those elements $\boldsymbol{s} \in \mathfrak{D}$ such that
$$\big(\mathfrak{I}_\ell, (\boldsymbol{s}, \boldsymbol{a}_1, \dots, \boldsymbol{a}_n)\big) \models \chi\,.$$

[2] We will see in Sec. 2.2 how to evaluate this sentence. Nonetheless, the interpretation we are about to define allows for the evaluation of every formula of $\mathcal{L}(\Lambda_\infty)$.

- pick as the image $\big(\mathfrak{F}_{\ell+1}(h_\chi)\big)(\vec{a})$ of $\vec{a}$ under the interpretation of h_χ the representative $\mathfrak{c}(S_\chi^{\vec{a}})$ of $S_\chi^{\vec{a}}$, namely, set
$$\big(\mathfrak{F}_{\ell+1}(h_\chi)\big)(\vec{a}) \coloneqq \mathfrak{c}(S_\chi^{\vec{a}}).$$

Finally, define $\mathfrak{I}_\infty = (\mathfrak{D}, \mathfrak{F}_\infty)$, where $\mathfrak{F}_\infty \coloneqq \bigcup_{\ell \in \mathbb{N}} \mathfrak{F}_\ell$.

It is straightforward that for each term or formula ϑ in the language $\mathcal{L}(\Lambda_\infty)$, there exists a number ℓ such that $\mathsf{val}_{\mathfrak{I}_{\ell+m},\mathfrak{N}}(\vartheta) = \mathsf{val}_{\mathfrak{I}_\infty,\mathfrak{N}}(\vartheta)$ holds for each $\mathfrak{D}$-valued sequence $\mathfrak{N}$ and for all $m \in \mathbb{N}$.

Remark 6 (Why does $\mathfrak{c}$ disappear in the completed interpretation $\mathfrak{I}_\infty$?). It would be pointless to write $\mathfrak{I}_\infty^{(\mathfrak{c})}$ instead of simply $\mathfrak{I}_\infty$, because $\mathfrak{c}$ only serves to enable the extension of each $\mathfrak{I}_\ell$ from the signature Λ_ℓ to the next signature $\Lambda_{\ell+1}$. Once the signature reaches the plateau, this auxiliary role of $\mathfrak{c}$ becomes redundant, because all sentences of $\mathcal{L}(\Lambda_\infty)$ can be evaluated in the interpretation $(\mathfrak{D}, \mathfrak{F}_\infty)$. ⊣

Remark 7. Let $\mathfrak{I}_\infty = (\mathfrak{D}, \mathfrak{F}_\infty)$ be the canonical completion of a selective structure $(\mathfrak{D}, \mathfrak{F}, \mathfrak{c})$, as above. Let χ be any n-adic key formula, for some $n \in \mathbb{N}$, so that $\neg\,\chi$ is also an n-adic key formula. For any n-tuple $\vec{a} = (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n)$ in $\mathfrak{D}^n$, we claim that
$$\big(\mathfrak{F}_\infty(h_\chi)\big)(\vec{a}) \neq \big(\mathfrak{F}_\infty(h_{\neg\,\chi})\big)(\vec{a}).$$
Indeed, since
$$S_\chi^{\vec{a}} \cap S_{\neg\,\chi}^{\vec{a}} = \emptyset \qquad \text{and} \qquad S_\chi^{\vec{a}} \cup S_{\neg\,\chi}^{\vec{a}} = \mathfrak{D},$$
it follows that
$$\big(\mathfrak{F}_\infty(h_\chi)\big)(\vec{a}) = \mathfrak{c}(S_\chi^{\vec{a}}) \neq \mathfrak{c}(S_{\neg\,\chi}^{\vec{a}}) = \big(\mathfrak{F}_\infty(h_{\neg\,\chi})\big)(\vec{a})$$
as claimed. ⊣

Remark 8. We show, in some detail, that for every formula φ of $\mathcal{L}(\Lambda_\infty)$—where Λ_∞ is the Skolem completion of an initial signature Λ—the formula $\varepsilon\, x.\, \varphi \neq \varepsilon\, x.\, \neg\,\varphi$ is valid. In particular, when $\varphi \equiv (x = x)$, we recover the validity of $\mathfrak{t} \neq \mathfrak{f}$, where $\mathfrak{t} \coloneqq \varepsilon\, x.\, x = x$ and $\mathfrak{f} \coloneqq \varepsilon\, x.\, x \neq x$, as stated in Remark 5.

Let $\mathfrak{I}_\infty = (\mathfrak{D}, \mathfrak{F}_\infty)$ be the canonical completion of an arbitrary selective structure $\mathfrak{I}^{(\mathfrak{c})} \coloneqq (\mathfrak{D}, \mathfrak{F}, \mathfrak{c})$, and let $\mathfrak{N}$ be an arbitrary variable assignment over $\mathfrak{D}$. It suffices to show that
$$(\mathfrak{I}_\infty, \mathfrak{N}) \models \varepsilon\, x.\, \varphi \neq \varepsilon\, x.\, \neg\,\varphi\,.$$
Let χ be the n-adic key formula $\varphi^{[x]}$ determined—along with n and the terms $\tau_1, \ldots, \tau_n$, none of which involves x—as in Thm. 1.1. Then we have $\varepsilon\, x.\varphi \equiv h_\chi(\tau_1, \ldots, \tau_n)$ and $\varepsilon\, x.(\neg\,\varphi) \equiv h_{\neg\,\chi}(\tau_1, \ldots, \tau_n)$. Define $\vec{a} \coloneqq (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n) \in \mathfrak{D}^n$, where $\boldsymbol{a}_i \coloneqq \mathsf{val}_{\mathfrak{I}_\infty,\mathfrak{N}}(\tau_i)$ for $i = 1, \ldots, n$. Then:
$$\begin{aligned}\mathsf{val}_{\mathfrak{I}_\infty,\mathfrak{N}}(h_\chi(\tau_1, \ldots, \tau_n)) &= \mathfrak{F}_\infty(h_\chi)(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n) \\ &\neq \mathfrak{F}_\infty(h_{\neg\,\chi})(\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n) = \mathsf{val}_{\mathfrak{I}_\infty,\mathfrak{N}}(h_{\neg\,\chi}(\tau_1, \ldots, \tau_n))\,.\end{aligned}$$
Thus, $(\mathfrak{I}_\infty, \mathfrak{N}) \models h_\chi(\tau_1, \ldots, \tau_n) \neq h_{\neg\,\chi}(\tau_1, \ldots, \tau_n)$, namely $(\mathfrak{I}_\infty, \mathfrak{N}) \models \varepsilon\, x.\, \varphi \neq \varepsilon\, x.\, \neg\,\varphi$. Since both the selective structure $\mathfrak{I}^{(\mathfrak{c})}$ and its canonical completion $\mathfrak{I}_\infty$, as well as the assignment $\mathfrak{N}$, were arbitrary, it follows that the formula $\varepsilon\, x.\, \varphi \neq \varepsilon\, x.\, \neg\,\varphi$ is valid. ⊣

3. Logical Laws, *Modus Ponens* Rule, and Derivations in the ε-calculus

We will explain how to properly enchain lists of formulae of $\mathcal{L}(\Lambda_\infty)$, where Λ_∞ originates from an initial signature Λ_0 in the manner discussed above, so that such a list can be regarded as a *derivation* in the Λ_0-based ε-calculus. The components of a derivation are called its *steps*, each of which is either a logical axiom, a proper axiom, or the immediate consequence of preceding steps. We will group logical axioms under a small number of schemes, referred to as *logical laws*. The proper axioms pertain to a specific intended use of our logical machinery. A single, historically significant inference rule, known as *modus ponens*, will suffice for our needs.

3.1. Logical laws of the ε-calculus

Here is a somewhat redundant selection of *logical laws* drawn from the valid formulae of $\mathcal{L}(\Lambda_\infty)$:

0. *Tautologies* belonging to $\mathcal{L}(\Lambda_\infty)$ — see p. 5.

1. All instances of the *equivalential properties* of equality: $\tau = \tau$ and $\tau = \tau' \to (\tau' = \sigma \to \sigma = \tau)$, where τ, τ', and σ are arbitrary terms of $\mathcal{L}(\Lambda_\infty)$.

2. All instances

$$\tau_0 = \sigma_0 \to \big(\tau_1 = \sigma_1 \to \big(\cdots \to \big(\tau_n = \sigma_n \to g(\tau_0, \tau_1, \ldots, t_n) = g(\sigma_0, \sigma_1, \ldots, \sigma_n)\big) \cdots\big)\big)$$

   of the *congruence property* of equality. Here $\tau_0, \sigma_0, \tau_1, \sigma_1, \ldots, \tau_n, \sigma_n$, and g, are terms and a functor of $\mathcal{L}(\Lambda_\infty)$; moreover, $\mathbf{d}(g) = n + 1$.

3. *Exclusion formulae.* These have the form

$$\boldsymbol{\varepsilon}\, x.\neg\, \varphi \neq \boldsymbol{\varepsilon}\, x.\varphi\,.$$

4. *Epsilon formulae.* These have the form $\varphi^x_\sigma \to \varphi^x_{\boldsymbol{\varepsilon}\, x.\varphi}$ (see Remark 1); that is, more explicitly,

$$\varphi^{[x]}(\sigma, \tau_1, \ldots, \tau_n) \quad \to \quad \varphi^{[x]}(\boldsymbol{\varepsilon}\, x.\varphi, \tau_1, \ldots, \tau_n)\,,$$

   or, even more explicitly,

$$\varphi^{[x]}(\sigma, \tau_1, \ldots, \tau_n) \quad \to \quad \varphi^{[x]}(h_\chi(\tau_1, \ldots, \tau_n), \tau_1, \ldots, \tau_n)\,.$$

   Here, σ is a term and φ a formula of $\mathcal{L}(\Lambda_\infty)$. Moreover, $\varphi^{[x]}$ is the key formula of degree n, $\tau_1, \ldots, \tau_n$ are the terms—none of which involves x—whose existence and uniqueness are established in Thm. 1.1, such that $\varphi \equiv \varphi^{[x]}(x, \tau_1, \ldots, \tau_n)$. The functor h_χ is the one uniquely associated with χ during the Skolem completion of the signature.

5. *Leisenring's axioms* (cf. [6, p. 13 and p. 40]. These have the form

$$\big(\forall\, z.\, (\varphi^x_z \leftrightarrow \psi^y_z)\,\big) \quad \to \quad \boldsymbol{\varepsilon}\, x.\varphi = \boldsymbol{\varepsilon}\, y.\psi\,,$$

   where x, y, z are variables, and φ and ψ are formulae of $\mathcal{L}(\Lambda_\infty)$, neither of which involves z.

Definition 3.1 (Derivability). Consider a collection $\mathcal{A} \cup \{\alpha\}$ of formulae of $\mathcal{L}(\Lambda_\infty)$. A finite sequence of formulae of $\mathcal{L}(\Lambda_\infty)$

$$\boldsymbol{\delta} \quad = \quad (\delta_0, \delta_1, \ldots, \delta_\ell)$$

is called a *derivation* of α from the set $\mathcal{A}$ of *premises* when $\delta_\ell \equiv \alpha$ and each δ_i (for $i = 0, \ldots, \ell$) satisfies one of the following conditions :

Logical axiom: δ_i is an instance of one of the logical laws listed above.

Premise (or proper axiom): δ_i belongs to $\mathcal{A}$.

Modus Ponens: There exist indices M and m such that $M < i$, $m < i$, and $\delta_M \equiv (\delta_m \to \delta_i)$.

The existence of such a list is indicated by the notation $\mathcal{A} \vdash \alpha$ —or simply $\vdash \alpha$ when $\mathcal{A} = \emptyset$. $\dashv$

In Sec. 4, we will establish that $\mathcal{A} \vdash \alpha$ implies $\mathcal{A} \models \alpha$ (*soundness*) and, conversely, that $\mathcal{A} \models \alpha$ implies $\mathcal{A} \vdash \alpha$ (*completeness*).

4. Soundness and Completeness of the ε-Calculus

4.1. Soundness of the ε-calculus

The ε-calculus, as introduced in the previous section, enjoys the following crucial metalogical property:

Theorem 4.1 (Soundness). *Let $\mathcal{A} \cup \{\alpha\}$ be a set of formulae of $\mathcal{L}(\Lambda_\infty)$. If $\mathcal{A} \vdash \alpha$, then $\mathcal{A} \models \alpha$.*

This claim is proved by induction on the length of a derivation of α from $\mathcal{A}$. To justify the *modus ponens* rule, observe that when $\mathfrak{T}\big((\varphi \to \psi)\big) = \mathbf{t}$ and $\mathfrak{T}(\varphi) = \mathbf{t}$ hold in a truth-value assignment $\mathfrak{T}$, then clearly $\mathfrak{T}(\psi) = \mathbf{t}$. It follows that $\mathcal{A} \models (\varphi \to \psi)$ and $\mathcal{A} \models \varphi$ imply $\mathcal{A} \models \psi$. Soundness will therefore follow directly from the validity of every logical axiom.

Lemma 4.1. *Every logical axiom is valid.*

Proof. Let the structure $\mathfrak{I}_\infty = (\mathfrak{D}, \mathfrak{F}_\infty)$ stem from a selective structure $\mathfrak{I}_0^{(\mathfrak{c})} = (\mathfrak{D}, \mathfrak{F}_0, \mathfrak{c})$ as discussed in Sec. 2.2, and let the sequence $\mathfrak{N}$ bind the variables ν_i to values in $\mathfrak{D}$ as described right before Def. 2.1. We must prove that $(\mathfrak{I}_\infty, \mathfrak{N}) \models \psi$ holds for every instance ψ of a logical law, where $\mathfrak{N}$ is any $\mathfrak{D}$-valued sequence. This is readily verified for laws 0, 1, and 2. As for law 3, its validity has already been established in Remark 8. The remaining two laws will be addressed next.

Epsilon formulae. A formula ψ falls under this law if and only if it can be written as

$$\psi \quad \equiv \quad \Big(\chi(\sigma, \tau_1, \ldots, \tau_n) \quad \to \quad \chi\big(h_\chi(\tau_1, \ldots, \tau_n), \tau_1, \ldots, \tau_n\big)\Big),$$

where $\chi \equiv \chi(\nu_0, \nu_1, \ldots, \nu_n)$ is an n-adic key formula, $\sigma, \tau_1, \ldots, \tau_n$ are terms of $\mathcal{L}(\Lambda_\infty)$, and h_χ is the functor uniquely associated with χ in the Skolem completion of the signature.

To show that $(\mathfrak{I}_\infty, \mathfrak{N}) \models \psi$, observe that the implication holds trivially unless

$$(\mathfrak{I}_\infty, \mathfrak{N}) \models \chi(\sigma, \tau_1, \ldots, \tau_n).$$

Assume this is the case. Let $\boldsymbol{a}_0 := \mathsf{val}_{\mathfrak{I}_\infty, \mathfrak{N}}(\sigma)$ and $\boldsymbol{a}_i := \mathsf{val}_{\mathfrak{I}_\infty, \mathfrak{N}}(\tau_i)$ for $i = 1, \ldots, n$, and define the tuples $\vec{a} := (\boldsymbol{a}_0, \boldsymbol{a}_1, \ldots, \boldsymbol{a}_n)$ and $\vec{a}^- := (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n)$. Then, by Definition 2.3, we have

$$(\mathfrak{I}_\infty, \vec{a}) \models \chi \quad \text{and} \quad \boldsymbol{a}_0 \in S_\chi^{\vec{a}^-},$$

so that $S_\chi^{\vec{a}^-} \neq \emptyset$. Thus, by construction of the canonical completion (see Remark 8), the element $\mathfrak{c}(S_\chi^{\vec{a}^-})$ belongs to $S_\chi^{\vec{a}^-}$, and so replacing $\boldsymbol{a}_0$ with $\mathfrak{c}(S_\chi^{\vec{a}^-})$ in $\vec{a}$ yields another tuple $\vec{a}^*$ such that

$$(\mathfrak{I}_\infty, \vec{a}^*) \models \chi.$$

But $\vec{a}^* = \big(\mathsf{val}_{\mathfrak{I}_\infty, \mathfrak{N}}(h_\chi(\tau_1, \ldots, \tau_n)), \boldsymbol{a}_1, \ldots, \boldsymbol{a}_n\big)$ by definition of the interpretation of h_χ, and therefore

$$(\mathfrak{I}_\infty, \mathfrak{N}) \models \chi(h_\chi(\tau_1, \ldots, \tau_n), \tau_1, \ldots, \tau_n).$$

Hence, $(\mathfrak{I}_\infty, \mathfrak{N}) \models \psi$, as required.

Leisenring's formulae. Written explicitly, Leisenring's axioms take the form

$$\Big(\chi_1\big(\varepsilon\, x.(\neg\, \chi), \tau_1, \ldots, \tau_n\big) \;\leftrightarrow\; \chi_2\big(\varepsilon\, x.(\neg\, \chi), \sigma_1, \ldots, \sigma_m\big)\Big) \quad \to \\ h_{\chi_1}(\tau_1, \ldots, \tau_n) = h_{\chi_2}(\sigma_1, \ldots, \sigma_m)\,,$$

where:

- χ_1, χ_2, and χ are key formulae of $\mathcal{L}(\Lambda_\infty)$, with χ_1 being n-adic and χ_2 being m-adic;
- $\tau_1, \ldots, \tau_n, \sigma_1, \ldots, \sigma_m$ are terms of $\mathcal{L}(\Lambda_\infty)$; and

- χ is the $(2\,n+2\,m)$-adic key formula associated, relative to the variable z, with the formula

$$\chi_1(z,\tau_1,\dots,\tau_n) \quad\leftrightarrow\quad \chi_2(z,\sigma_1,\dots,\sigma_m)$$

(Clue: In light of the more compact earlier formulation of these axioms, the guiding idea is that χ_1 and χ_2 stand for $(\varphi^x_z)^{[z]}$ and $(\psi^y_z)^{[z]}$, respectively — see Remark 2.)

Assuming that

$$(\mathfrak{I}_\infty,\mathfrak{N}) \;\models\; \chi_1\big(\varepsilon\,x.(\neg\,\chi),\tau_1,\dots,\tau_n\big) \leftrightarrow \chi_2\big(\varepsilon\,x.(\neg\,\chi),\sigma_1,\dots,\sigma_m\big)$$

holds, we proceed to verify that

$$(\mathfrak{I}_\infty,\mathfrak{N}) \;\models\; h_{\chi_1}(\tau_1,\dots,\tau_n) \;=\; h_{\chi_2}(\sigma_1,\dots,\sigma_m).$$

Clearly, we have $(\mathfrak{I}_\infty,\widetilde{\mathfrak{N}}) \;\models\; \chi_1 \leftrightarrow \chi_2(\nu_0,\nu_{n+1},\dots,\nu_{n+m})$, where $\widetilde{\mathfrak{N}}$ is any $\mathfrak{D}$-valued sequence whose initial segment $(\boldsymbol{a}_0,\boldsymbol{a}_1,\dots,\boldsymbol{a}_{n+m})$ consists of the values $\boldsymbol{a}_0=\mathsf{val}_{\mathfrak{I}_\infty,\mathfrak{N}}(\varepsilon\,x.(\neg\,\chi))$, $\boldsymbol{a}_i=\mathsf{val}_{\mathfrak{I}_\infty,\mathfrak{N}}(\tau_i)$ for $i=1,\dots,n$, and $\boldsymbol{a}_{n+j}=\mathsf{val}_{\mathfrak{I}_\infty,\mathfrak{N}}(\sigma_j)$, for $j=1,\dots,m$. For each $\boldsymbol{s}$ in $\mathfrak{D}$, let us denote by $\mathfrak{N}^{\boldsymbol{s}}_1$ and $\mathfrak{N}^{\boldsymbol{s}}_2$ generic $\mathfrak{D}$-valued sequences having, respectively, initial segments $(\boldsymbol{s},\boldsymbol{a}_1,\dots,\boldsymbol{a}_n)$ and $(\boldsymbol{s},\boldsymbol{a}_{n+1},\dots,\boldsymbol{a}_{n+m})$; accordingly, we have $(\mathfrak{I}_\infty,\mathfrak{N}^{\boldsymbol{a}_0}_1)\models\chi_1$ if and only if $(\mathfrak{I}_\infty,\mathfrak{N}^{\boldsymbol{a}_0}_2)\models\chi_2$.

Consider the first ℓ such that χ_1 and χ_2 are formulae, and $\tau_1,\dots,\tau_n,\sigma_1,\dots,\sigma_m$ are terms, of $\mathcal{L}(\Lambda_\ell)$. Then, by construction, the set S_{χ_1} of those $\boldsymbol{s}$ in $\mathfrak{D}$ such that $(\mathfrak{I}_\ell,\mathfrak{N}^{\boldsymbol{s}}_1)\models\chi_1$ coincides with the set S_{χ_2} of those $\boldsymbol{s}$ in $\mathfrak{D}$ such that $(\mathfrak{I}_\ell,\mathfrak{N}^{\boldsymbol{s}}_2)\models\chi_2$. It easily follows that $\mathsf{val}_{\mathfrak{I}_{\ell+1},\widetilde{\mathfrak{N}}}(h_{\chi_1}(\tau_1,\dots,\tau_n)) = \mathfrak{c}(S_{\chi_1}) = \mathfrak{c}(S_{\chi_2}) = \mathsf{val}_{\mathfrak{I}_{\ell+1},\widetilde{\mathfrak{N}}}(h_{\chi_2}(\sigma_1,\dots,\sigma_m))$ and therefore $(\mathfrak{I}_{\ell+1},\widetilde{\mathfrak{N}}) \models h_{\chi_1}(\tau_1,\dots,\tau_n) = h_{\chi_2}(\sigma_1,\dots,\sigma_m)$, whence $(\mathfrak{I}_\infty,\mathfrak{N}) \models h_{\chi_1}(\tau_1,\dots,\tau_n) = h_{\chi_2}(\sigma_1,\dots,\sigma_m)$, as desired. $\square$

4.2. The ε-calculus is a Boolean logic

In preparation for the completeness proof, let us momentarily digress to check that the ε-calculus satisfies all the properties of a Boolean logic, as intended in [7]. A benefit of this verification is that, thanks to a key theorem applicable to all Boolean logics, the completeness proof will be significantly simplified.

Here are the presupposed notions:

Definition 4.1. A *logic* is a pair $(\mathcal{S},\vdash_{\mathcal{S}})$ consisting of

- a set $\mathcal{S}\neq\emptyset$ whose elements are called *statements* and
- a relation $\vdash_{\mathcal{S}}$ included in $\mathscr{P}(\mathcal{S})\times\mathcal{S}$, called *derivability*,

which satisfies the following conditions, for all $\alpha,\beta\in\mathcal{S}$:

L1. $\{\alpha\}\vdash_{\mathcal{S}}\alpha$.

L2. **(Monotonicity)** $\mathcal{A}\vdash_{\mathcal{S}}\alpha$ implies $\mathcal{B}\vdash_{\mathcal{S}}\alpha$ when $\mathcal{A}\subseteq\mathcal{B}\subseteq\mathcal{S}$.

L3. **(Compactness)** $\mathcal{A}\vdash_{\mathcal{S}}\alpha$ implies that $\mathcal{F}\vdash_{\mathcal{S}}\alpha$ holds for some <u>finite</u> set $\mathcal{F}\subseteq\mathcal{A}$.

L4. **(Cut)** From $\mathcal{A}\vdash_{\mathcal{S}}\alpha$ and $\mathcal{B}\cup\{\alpha\}\vdash_{\mathcal{S}}\beta$ it follows that $\mathcal{A}\cup\mathcal{B}\vdash_{\mathcal{S}}\beta$. $\dashv$

Definition 4.2. A *Boolean logic* is a quadruple

$$\mathbb{L} \;=\; (\mathcal{S},\vdash_{\mathcal{S}},f,\Rightarrow)$$

consisting of

- a logic $(\mathcal{S}, \vdash_{\mathcal{S}})$ as above,
- a distinguished statement $f \in \mathcal{S}$, and
- a dyadic operation $\Rightarrow : \mathcal{S} \times \mathcal{S} \longrightarrow \mathcal{S}$,

which satisfies the following conditions, for all $\mathcal{A} \subseteq \mathcal{S}$ and $\alpha \in \mathcal{S}$:

B1. **(Deduction principle)** $\mathcal{A} \vdash_{\mathcal{S}} \alpha \Rightarrow \beta$ holds if and only if $\mathcal{A} \cup \{\alpha\} \vdash_{\mathcal{S}} \beta$.

B2. **(Double negation principle)** $\{(\alpha \Rightarrow f) \Rightarrow f\} \vdash_{\mathcal{S}} \alpha$.

The syntactic operation $\Rightarrow$ is called *implication* (just like the operation on truth-values denoted by $\rightarrow$), and its first and second operands are called its *antecedent* and *consequent*, respectively. $\dashv$

In what follows, assuming that

- $\mathcal{S}$ is the set of all formulae of $\mathcal{L}(\Lambda_\infty)$,
- $\vdash_{\mathcal{S}}$ is the derivability relation $\vdash$ introduced in Def. 3.1,
- f is the formula $\mathtt{f}$, and
- $\Rightarrow$ constructs the formula $(\varphi \rightarrow \psi)$ from a given antecedent φ and consequent ψ ,

we proceed to verify that the conditions L1.–L4., B1., and B2. are all satisfied.

In fact,

- L1., L2., and L3. are immediate;
- L4. is almost so;
- B2. is derived in three steps:
  - $(\alpha \rightarrow \mathtt{f}) \rightarrow \mathtt{f}$,
  - $((\alpha \rightarrow \mathtt{f}) \rightarrow \mathtt{f}) \rightarrow \alpha$, and
  - α,

  which correspond to the premise, a tautology, and the result of modus ponens, respectively.

As for B1., assume first that the sequence $(\delta_0, \delta_1, \ldots, \delta_\ell)$ derives $(\alpha \rightarrow \beta) \equiv \delta_\ell$ from $\mathcal{A}$ in the ε-calculus; then, by adding two more steps—α and β—at the end, we obtain a derivation of β from $\mathcal{A} \cup \{\alpha\}$. To get the converse, suppose that $(\delta_0, \delta_1, \ldots, \delta_\ell)$ is a derivation of β from $\mathcal{A} \cup \{\alpha\}$. Inductively, for each $i = 0, 1, \ldots, \ell$, we construct a derivation of $\alpha \rightarrow \delta_i$ from $\mathcal{A}$ as follows.

- If $\delta_i \equiv \alpha$, then $\alpha \rightarrow \delta_i$ is a tautology; a one-step derivation proves it.
- If δ_i is a logical axiom or belongs to $\mathcal{A}$, then a three-step derivation
  - δ_i,
  - $\delta_i \rightarrow (\alpha \rightarrow \delta_i)$,
  - $\alpha \rightarrow \delta_i$

  does the job.
- If δ_i is obtained from preceding steps δ_j and $\delta_j \rightarrow \delta_i$, we concatenate the derivations of

$$\alpha \rightarrow \delta_j \quad \text{and} \quad \alpha \rightarrow (\delta_j \rightarrow \delta_i)$$

  (which exist, by the induction hypothesis), and continue with the tautology

$$\big(\alpha \rightarrow (\delta_j \rightarrow \delta_i)\big) \rightarrow \big((\alpha \rightarrow \delta_j) \rightarrow (\alpha \rightarrow \delta_i)\big);$$

  then conclude as desired, by applying modus ponens twice.

In the special case when $i = \ell$, we obtain a derivation of $\alpha \to \beta$ from $\mathcal{A}$.

This completes the verification of B1., thereby showing that the ε-calculus is a Boolean logic.

Before recalling three important metalogical propositions, we define:

Definition 4.3. A set $\mathcal{A}$ of statements of a Boolean logic $\mathbb{L} = (\mathcal{S}, \vdash_{\mathcal{S}}, f, \Rightarrow)$ is said to be *consistent* if there exist statements α in $\mathcal{S}$ such that $\mathcal{A} \nvdash_{\mathcal{S}} \alpha$, i.e., α is not derivable from $\mathcal{A}$. $\dashv$

Here are the announced propositions, whose proofs can be found in [7]:[3]

Lemma 4.2 (Lindenbaum). *For every consistent set $\mathcal{A}$ of statements in a Boolean logic $\mathbb{L}$, there exists a* maximally consistent *set $\mathcal{M}$ of statements in $\mathbb{L}$ such that $\mathcal{A} \subseteq \mathcal{M}$. That is, $\mathcal{M}$ is consistent, and for every statement α in $\mathbb{L}$ with $\alpha \notin \mathcal{M}$, the set $\mathcal{M} \cup \{\alpha\}$ is inconsistent.*

Lemma 4.3. *If $\mathcal{A} \nvdash_{\mathcal{S}} f$ in a Boolean logic $\mathbb{L}$ as above, then there exists an assignment $\mathfrak{T} \colon \mathcal{S} \longrightarrow \{\mathbf{f}, \mathbf{t}\}$ satisfying the following conditions:*

- $\mathfrak{T}(f) = \mathbf{f}$,
- $\mathfrak{T}(\varphi \Rightarrow \psi) = \textbf{\textit{if }} \mathfrak{T}(\varphi) = \mathbf{t} \textbf{\textit{ then }} \mathfrak{T}(\psi) \textbf{\textit{ else }} \mathbf{t}$, *for all φ, ψ in $\mathcal{S}$*, *and*
- $\mathfrak{T}(\beta) = \mathbf{t}$, *for all $\beta \in \mathcal{A}$.*

Theorem 4.2 (Key theorem for Boolean logics). *Let $\mathbb{L}$ be a Boolean logic as above. Let $\mathcal{A}$ be a set of statements, and α a statement, in $\mathbb{L}$. If every assignment $\mathfrak{T} \colon \mathcal{S} \longrightarrow \{\mathbf{f}, \mathbf{t}\}$ satisfying the three conditions stated in Lemma 4.3 sends α to $\mathbf{t}$, then $\mathcal{A} \vdash_{\mathcal{S}} \alpha$ holds.*

Relying on the fact that the ε-calculus constitutes a Boolean logic—particularly its deduction principle—we conclude with two instructive derivations. These establish standard quantifier equivalences which, though often taken for granted in classical logic, require formal justification within the ε-calculus.

4.2.1. Interdeducibility of $\neg(\exists\, x.\, \neg\varphi)$ and $(\forall\, x.\, \varphi)$, and between $\neg(\forall\, x.\, \neg\varphi)$ and $(\exists\, x.\, \varphi)$

We now demonstrate how the ε-calculus supports two classical equivalences involving quantifiers and negation: the equivalence between universal quantification and the negation of an existential, and vice versa. While these relationships are well known, formalizing them within the ε-calculus requires careful manipulation of ε-terms and the use of Leisenring's axioms. These derivations underscore the expressive power and internal coherence of the ε-calculus as a Boolean logic.

It is straightforward to derive $(\forall\, x.\, \varphi)$ from the single premise $\neg(\exists\, x.(\neg\varphi))$ and, conversely, to derive $\neg\,(\exists\, x.(\neg\, \varphi))$ from the premise $(\forall\, x.\, \varphi)$.

Recall that, in our notation:

$$\neg(\exists\, x.(\neg\varphi) \equiv \neg\left(\neg\, \varphi^{x}_{\varepsilon\, x.\neg\varphi}\right) \qquad \text{and} \qquad (\forall\, x.\, \varphi) \equiv \varphi^{x}_{\varepsilon\, x.\neg\varphi}.$$

Thus, to derive $(\forall\, x.\, \varphi)$ from $\{\neg(\exists\, x.(\neg\varphi)\}$ in the ε-calculus, it suffices to prove:

$$\left\{\neg\left(\neg\, \varphi^{x}_{\varepsilon\, x.\neg\varphi}\right)\right\} \vdash \varphi^{x}_{\varepsilon\, x.\neg\varphi}\,.$$

Since the ε-calculus is a Boolean logic, by the Deduction Principle B1, this reduces to proving:

$$\vdash \left(\neg\left(\neg\, \varphi^{x}_{\varepsilon\, x.\neg\varphi}\right)\right) \to \varphi^{x}_{\varepsilon\, x.\neg\varphi}\,.$$

This implication is trivially valid, being a tautology—an axiom of type 0.

The converse direction, namely the derivation of $\neg(\exists\, x.(\neg\varphi))$ from $\{(\forall\, x.\, \varphi)\}$, follows by a symmetric argument.

[3] A fully general proof of Lindenbaum's lemma follows straightforwardly from the Zorn lemma. A more elementary proof can be given (see, e.g., [7, pp.8–9]) for the case when the set $\mathcal{S}$ is countable and effectively listable.

We now turn to the more involved equivalence between $\neg\,(\forall\, x.(\neg\,\varphi))$ and $(\exists\, x.\,\varphi)$. In our notation:

$$\neg\,(\forall\, x.(\neg\,\varphi)) \equiv \neg\,\left(\neg\,\varphi^{x}_{\boldsymbol{\varepsilon}\, x.(\neg(\neg\varphi))}\right) \qquad \text{and} \qquad (\exists\, x.\,\varphi) \equiv \varphi^{x}_{\boldsymbol{\varepsilon}\, x.\varphi}\,.$$

To derive $(\exists\, x.\,\varphi)$ from $\{\neg\,(\forall\, x.(\neg\,\varphi))\}$, we aim to prove:

$$\left\{\neg\,\left(\neg\,\varphi^{x}_{\boldsymbol{\varepsilon}\, x.(\neg(\neg\varphi))}\right)\right\} \vdash \varphi^{x}_{\boldsymbol{\varepsilon}\, x.\varphi}\,.$$

We begin by applying modus ponens to the tautology $\left(\neg\,\left(\neg\,\varphi^{x}_{\boldsymbol{\varepsilon}\, x.(\neg(\neg\varphi))}\right)\right) \to \varphi^{x}_{\boldsymbol{\varepsilon}\, x.(\neg(\neg\varphi))}$ and the premise $\neg\,\left(\neg\,\varphi^{x}_{\boldsymbol{\varepsilon}\, x.(\neg(\neg\varphi))}\right)$, obtaining:

$$\varphi^{x}_{\boldsymbol{\varepsilon}\, x.(\neg(\neg\varphi))}\,. \tag{1}$$

Now, let $\psi \coloneqq \big((\neg(\neg\varphi)) \leftrightarrow \varphi\big)$. Then, $\big(\forall x.((\neg(\neg\varphi)) \leftrightarrow \varphi)\big)$ is expressed as:

$$\big(\neg(\neg(\varphi^{x}_{\boldsymbol{\varepsilon}\, x.\neg\psi}))\big) \;\leftrightarrow\; \varphi^{x}_{\boldsymbol{\varepsilon}\, x.\neg\psi}\,. \tag{2}$$

We include in the derivation the tautology (2), along with the following instance of Leisenring's axiom:

$$\big((\neg(\neg(\varphi^{x}_{\boldsymbol{\varepsilon}\, x.\neg\psi}))) \leftrightarrow \varphi^{x}_{\boldsymbol{\varepsilon}\, x.\neg\psi}\big) \;\to\; \boldsymbol{\varepsilon}\, x.\big(\neg(\neg(\varphi))\big) = \boldsymbol{\varepsilon}\, x.\varphi\,. \tag{3}$$

Applying modus ponens to (3) and (2) yields:

$$\boldsymbol{\varepsilon}\, x.\big(\neg(\neg(\varphi))\big) = \boldsymbol{\varepsilon}\, x.\varphi\,. \tag{4}$$

We rely on the general theorem

$$\vdash \boldsymbol{\varepsilon}\, x.\big(\neg(\neg(\varphi))\big) = \boldsymbol{\varepsilon}\, x.\varphi \;\to\; \left(\varphi^{x}_{\boldsymbol{\varepsilon}\, x.(\neg(\neg\varphi))} \to \varphi^{x}_{\boldsymbol{\varepsilon}\, x.\varphi}\right)\,, \tag{5}$$

whose proof proceeds by structural induction on φ.

Applying modus ponens to (5) and (4), we obtain:

$$\varphi^{x}_{\boldsymbol{\varepsilon}\, x.(\neg(\neg\varphi))} \;\to\; \varphi^{x}_{\boldsymbol{\varepsilon}\, x.\varphi}\,. \tag{6}$$

Finally, from (1) and (6), we derive $\varphi^{x}_{\boldsymbol{\varepsilon}\, x.\varphi}$ by modus ponens, completing the proof.

A symmetric argument shows that $\neg\,(\forall\, x.\neg\,\varphi)$ can likewise be derived from the premise $(\exists\, x.\,\varphi)$.

4.3. Completeness of the ε-calculus

Theorem 4.3 (Completeness). *Consider a set $\mathcal{A} \cup \{\alpha\}$ of sentences of $\mathcal{L}(\Lambda_\infty)$. If $\mathcal{A} \models \alpha$, then $\mathcal{A} \vdash \alpha$.*

Proof. Suppose $\mathcal{A} \models \alpha$. Let $\mathcal{E}$ denote the collection of all logical axioms of $\mathcal{L}(\Lambda_\infty)$ other than tautologies. Now consider a generic truth value assignment $\mathfrak{T}$ for $\mathcal{L}(\Lambda_\infty)$, as defined in Def. 2.1, such that $\mathfrak{T}(\varphi) = \mathbf{t}$ for every φ in $\mathcal{A} \cup \mathcal{E}$. We will show that $\mathfrak{T}(\alpha) = \mathbf{t}$. It will then follow, by the key theorem for Boolean logics (as presented in Sec. 4.2 and applicable to the ε-calculus), that $\mathcal{A} \cup \mathcal{E} \vdash \alpha$. Since $\mathcal{E}$ consists of logical axioms, we can conclude that $\mathcal{A} \vdash \alpha$, as required.

A preliminary step in proving that $\mathfrak{T}(\alpha) = \mathbf{t}$ consists in constructing a selective structure $\mathfrak{I}_0^{(\mathfrak{c})} = (\mathfrak{D}, \mathfrak{F}, \mathfrak{c})$ that complies with $\mathfrak{T}$ in the sense that $\mathsf{val}_{\mathfrak{I}_\infty,\mathfrak{N}}(\beta) = \mathfrak{T}(\beta)$ for every formula β, independently of how the $\mathfrak{D}$-valued sequence $\mathfrak{N}$ is chosen.

As the domain of discourse $\mathfrak{D}$ of $\mathfrak{I}_0$ (and, consequently, of $\mathfrak{I}_\infty$), we adopt the quotient of the collection of all terms of $\mathcal{L}(\Lambda_\infty)$ with respect to the equivalence relation $\approx$, where $\tau \approx \sigma$ holds between two terms τ and σ if and only if $\mathfrak{T}(\tau = \sigma) = \mathbf{t}$. Each functor g in Λ_∞ is interpreted as the function $\mathfrak{F}(g)$ that maps every $\mathbf{d}(g)$-tuple $\big(\tau_1, \ldots, \tau_{\mathbf{d}(g)}\big)$ of terms to the term $g\big(\tau_1, \ldots, \tau_{\mathbf{d}(g)}\big)$. This definition is well-posed because

$$g\big(\tau_1, \ldots, \tau_{\mathbf{d}(g)}\big) \approx g\big(\sigma_1, \ldots, \sigma_{\mathbf{d}(g)}\big)$$

follows from

$$\tau_1 \approx \sigma_1, \ldots, \tau_{\mathbf{d}(g)} \approx \sigma_{\mathbf{d}(g)}$$

thanks to the inclusion of all instances of the congruence property of equality among the logical axioms.

The preservation of truth values—namely, the identity $\mathsf{val}_{\mathfrak{I}_\infty,\mathfrak{N}}(\beta) = \mathfrak{T}(\beta)$—is proved by a straightforward induction on the number of occurrences of '$\rightarrow$' in β. Since $\mathfrak{T}$ models $\mathcal{A}$, so does $\left(\mathfrak{I}_\infty, \mathfrak{N}\right)$; therefore $\mathsf{val}_{\mathfrak{I}_\infty,\mathfrak{N}}(\alpha) = \mathbf{t}$, thanks to hypothesis $\mathcal{A} \models \alpha$. The sought conclusion $\mathfrak{T}(\alpha) = \mathbf{t}$ follows readily. □

5. Commentary

1. The version of the ε-calculus proposed above builds on—but slightly differs from—the one presented in [2]. The main differences are:
   - The suppression of all relators except for equality, which Davis and Fechter considered an optional construct.
   - The restoration of *Leisenring's logical law*, which was missing from [2].

   A minor change is the introduction of what we dub *exclusion formulae* among the logical axioms, along with a corresponding revision to the semantics.

2. In [2], Davis and Fechter prove a conservativeness result: their version of the ε-calculus can mimic the proofs in Shoenfield's formalization of first-order predicate logic [8]. Indirectly, this establishes the completeness of the ε-calculus. Our completeness proof, however, is autonomous, relying exclusively on the proposed semantics for the ε-calculus.

3. In [2], Davis and Fechter present three elementary examples illustrating "the kinds of proof procedures which our free variable formulation should make possible." Based on this, they conclude: "there is reason to believe that they may turn out to be of interest". The authors of this paper share their expectation that the ε-calculus can effectively support predicate calculus theorem-proving.

4. The logical axioms we have indicated as the basis of the ε-calculus can be significantly simplified by adopting a few tautological schemes instead of all tautologies. Which schemes? An elegant option—one among several—was proposed by Quine [9] and it neatly separates the role of implication from that of negation. It comprises all formulae of the following four forms:

   $$\left(\alpha \rightarrow \beta\right) \rightarrow \left((\beta \rightarrow \gamma) \rightarrow (\alpha \rightarrow \gamma)\right), \qquad \left((\alpha \rightarrow \beta) \rightarrow \alpha\right) \rightarrow \alpha,$$
   $$\alpha \rightarrow (\beta \rightarrow \alpha), \qquad \mathtt{f} \rightarrow \alpha.$$

   Another simplification consists in postulating, instead of all instances of the reflexive property of equality ($\tau = \tau$, where τ can be any term), only those of the form $x = x$, where x is a variable.

5. We have formulated *exclusion formulae* as

   $$\varepsilon\, x.\neg\, \varphi \neq \varepsilon\, x.\varphi\,.$$

   However, for the purpose of ensuring completeness, it would suffice to include only restricted instances of such formulae—namely, those of the form

   $$(\forall x.\varphi) \rightarrow \varepsilon\, x.\neg\, \varphi \neq \varepsilon\, x.\varphi\,.$$

   This restricted version captures all cases needed in the completeness proof, while reducing the logical overhead.

Acknowledgments

This contribution was partially supported by the research program *PIAno di inCEntivi per la Ricerca di Ateneo 2024/2026 – Linea di Intervento I "Progetti di ricerca collaborativa"*, Università di Catania, under the project *"Semantic Web of EveryThing through Ontological Protocols" (SWETOP).*

The first author is affiliated with the National Centre for HPC, Big Data and Quantum Computing (Project CN00000013, Spoke 10), co-funded by the European Union – NextGenerationEU.

All authors are members of the GNCS (Gruppo Nazionale per il Calcolo Scientifico) of INdAM.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] D. Hilbert, P. Bernays, Grundlagen der Mathematik *I and II*, Die Grundlehren der Mathematischen Wissenschaften, 2nd ed., Verlag Julius Springer, 1968-1970. French translation. Fondements des mathématiques. 2 volumes. By F. Gaillard, E. Gaillard, and M. Guillaume. Paris: l'Harmattan, 2001.

[2] M. Davis, R. Fechter, A free variable version of the first-order predicate calculus, J. of Logic and Computation 1 (1991) 431–451.

[3] D. Cantone, M. Nicolosi Asmundo, A further and effective liberalization of the δ-rule in free variable semantic tableaux, in: R. Caferra, G. Salzer (Eds.), Automated Deduction in Classical and Non-Classical Logics, volume 1761 of *Lecture Notes in Computer Science*, Springer-Verlag, 2000, pp. 109–125.

[4] D. Cantone, M. Nicolosi Asmundo, Skolem functions and Hilbert's ϵ-terms in free variable tableau systems, in: CILC06 - Convegno italiano di Logica Computazionale - 26-27 June 2006, Bari, 2006. URL: http://cilc2006.di.uniba.it/download/camera/13_Cantone_CILC06.pdf.

[5] D. Cantone, M. Nicolosi Asmundo, A sound framework for δ-rule variants in free variable semantic tableaux, Journal of Automated Reasoning 38 (2007) 31–56.

[6] A. C. Leisenring, Mathematical Logic and Hilbert's ε-symbol, MacDonald & Co., 1969.

[7] M. Davis, Lecture notes in Logic, Courant Inst. of Mathematical Sc., New York University, 1993.

[8] J. R. Shoenfield, Mathematical Logic, Addison-Wesley, Reading, MA, 1967.

[9] W. V. Quine, Completeness of the propositional calculus, J. Symb. Logic 3 (1938) 37–40.