First-Order Predicate Calculus: Finitary Syntax, Derivation, and Models

N. I. Kazimirov

2026

Continuation of the note "The Finitary Basis of Formal Systems and a Rigorous Construction of the Propositional Calculus with Its Models" (PDF).

Contents

Introduction **1**

1 Syntax of a First-Order Language **3**
1.1 Alphabet, Variables, Templates, and Signatures 3
1.2 Terms, Relevant Lists, and Formulas of a Signature 7
1.3 Translating the Formal Canon into the Internal Representation of the Lean Proof Assistant 13

2 The Predicate Calculus **15**
2.1 First-Order Language 15
2.2 Axioms and Rules of Inference 17
2.3 Derivability in the Context of a List of Hypotheses 19
2.4 First-Order Theories 24
2.5 Definitional Extensions of the Language and Theories 27

3 Semantics and Models **32**
3.1 Structures and Tarski Truth 32
3.2 Soundness and Completeness 34
3.3 The Propositional Calculus as a Fragment of the Predicate Calculus 35

Conclusion. The Postulate of Binarity and the Spiral of Metatheories **38**

References **43**

Introduction

The present text continues the paper "The Finitary Basis of Formal Systems and a Rigorous Construction of the Propositional Calculus with Its Models" (hereafter **PC**) and constructs

the first-order predicate calculus (**FOL**) on the same finitary basis. The special metalanguage introduced in Part I of the PC paper is used unchanged: its domain of discourse is exhausted by finite words over a fixed alphabet, the basic operations and relations on them (concatenation, identity, occurrence, and word-for-word replacement), and the recursive generation of classes of words and relations between words, without appealing to infinite collections or natural numbers.

Constructing a first-order language strictly within these boundaries requires resolving two syntactic problems that did not arise in the propositional calculus.

The first problem concerns the notions of signature and arity. Unlike the propositional calculus, whose language has a single fixed vocabulary, the syntax of first-order logic is parameterized by a collection of non-logical symbols—constants, function symbols, and predicate symbols. In a set-theoretic exposition, each function and predicate symbol is assigned a natural number—its arity (number of arguments)—and the formation rules for terms and atomic formulas are written with argument lists of variable length.

Because the initial special language contains no separate class of natural numbers, function and predicate stems are generated recursively over a fixed finite alphabet in the same manner as variables, while the numerical arity function is replaced by a purely string-based construction: a string of special placeholders (slots) in parentheses, separated by commas, is adjoined to a head name. Function templates (with placeholders) are thereby formed from function stems by adjoining a string of slots, whereas function stems without attached slots simultaneously play the role of constants, realizing constants as 0-ary functions; analogously, predicate templates are formed from predicate stems by adjoining a string of slots.

All function stems (constants), function templates, and predicate templates of arbitrary numbers of slots form recursively generated decidable classes of words, while a signature is specified at the metalevel as a selection from these classes possessing the antichain property: for each head name, at most one word belongs to the signature. In order to substitute arguments into a function or predicate template all at once as a list while matching the number of argument slots without counting, comma-separated term lists are generated by simultaneous recursion synchronously with strings of placeholders. A term list generated alongside the placeholder string of a given template matches it in length (is relevant to it), and forming a compound term or an atomic formula reduces to adjoining a relevant term list to the head name of the template.

The resources of the special language (a finite alphabet, operations on words, structural recursion, and structural induction) suffice in principle to construct the natural numbers themselves as a separate class of words—both in unary notation (strings of a single symbol) and in positional notation with digits and arithmetic operations (in expressive power such a metalanguage corresponds to primitive recursive arithmetic). In effect, strings of variables and strings of placeholders already constitute a unary implementation of natural numbers, and simultaneous recursion for relevant lists implements the check that their lengths coincide. Embedding a unary counter directly into positional templates avoids introducing a separate class of numeral representations and auxiliary digit arithmetic prior to constructing the syntax of the calculus, while preserving a strictly finite object alphabet; once unique readability has been proved (§ 1.2), numerical notation for arities and subscripts is introduced in § 2.1 as an abbreviation of this canon.

The second problem concerns term substitution and quantifier binding. If one uses a single stock of individual variables both at free positions in terms and under quantifiers, then word-for-word replacement from Part I of the PC paper ceases to be a sound operation on formulas: it affects bound occurrences of the variable being replaced, and variables inside

the substituted term may fall within the scope of quantifiers.

To retain basic word-for-word replacement without introducing complex auxiliary predicates for admissibility of substitution and variable renaming, individual variables are partitioned already at the alphabet level into two disjoint classes—free and bound—while the propositional variables of the propositional calculus form a third class disjoint from both. Terms are built exclusively from free variables and constants, whereas bound variables are introduced solely by the quantifier attachment rule: in an already formed formula, all occurrences of a chosen free variable are replaced by a bound variable that does not yet occur in that formula. As a result, no term contains bound variables, variable capture by quantifiers upon term substitution is syntactically impossible, and the condition that a variable has no free occurrences in a formula reduces to elementary non-occurrence of one word in another. Furthermore, the independence of propositional variables from the FOL vocabulary makes it possible to obtain propositional axioms and tautologies of first-order logic by direct word-for-word substitution of FOL formulas for propositional variables in axioms and theorems of the propositional calculus.

The exposition is organized as follows. Section 1 is kept strictly finitary: in the canon of the special language, we define the classes of variables, non-logical names, positional templates, the notion of a signature as an antichain, and the classes of relevant term lists, terms, and formulas of a given signature; we establish unique readability and examine the analogy between this string-based canon and the internal representation of the Lean proof assistant kernel.

Section 2 passes to the more customary set-theoretic notation (§ 2.1), specifies the axioms, rules of inference, and derivability in the context of a finite list of hypotheses, and establishes the Renaming Lemma, the Deduction Theorem, and the basic properties of derivability. In §§ 2.4–2.5, non-finitary superstructures are built atop this finitary core of derivations—arbitrary sets of formulas, first-order theories, and countable chains of definitional extensions (the theorem on eliminability and conservativity). In Section 3, standard set theory is used to construct the semantics: structures and Tarski truth are introduced, the soundness of FOL is proved, Gödel's Completeness Theorem is stated (with its detailed proof deferred to the companion note "The Henkin Construction"), its principal corollaries (the Compactness Theorem and the Downward Löwenheim–Skolem Theorem) are derived, and two representations of the propositional calculus within first-order logic are compared—as the quantifier-free fragment of equality-free FOL and as the theory of Boolean algebras. The Conclusion traces the spiral of metatheoretic ascent from the pre-formal postulate of binarity, through the induction-free core ($\mathsf{TC} \equiv \mathsf{Q}$) and the finitary arithmetic stage of induction ($\mathsf{TC} + \mathrm{Ind} \equiv \mathsf{PA} \equiv \mathsf{HF}$, with the finitary minimum $\mathsf{PRA} \equiv_{\Pi_2} I\Sigma_1$ and EA), to set-theoretic semantics (ZFC).

1 Syntax of a First-Order Language

1.1 Alphabet, Variables, Templates, and Signatures

Unlike the propositional calculus, whose language has a single fixed alphabet, the syntax of first-order logic is parameterized by the choice of a signature—a collection of constants, function templates, and predicate templates. To remain within the finitary special language of Part I of the PC paper without appealing to natural numbers, the basic object alphabet is fixed once and for all: within it, all sorts of variables as well as function and predicate stems together with their positional templates are generated recursively.

A new *object alphabet* $\mathbb{A}$ consisting of 16 symbols is fixed:

$$\mathsf{p} \quad \mathsf{a} \quad \mathsf{x} \quad \mathfrak{F} \quad \mathfrak{P} \quad \lrcorner \quad (\quad) \quad , \quad = \quad \neg \quad \wedge \quad \vee \quad \to \quad \forall \quad \exists$$

All listed symbols are pairwise distinct and disjoint from the metasymbols of the special language. With the transition to the alphabet $\mathbb{A}$, the notion of a word from PC changes its extension: henceforth, *words* everywhere mean finite strings of symbols of the alphabet $\mathbb{A}$, and accordingly the range of the general word metavariables $\xi, \eta, \zeta, \theta, \tau, \omega$ from § I.1 of the PC paper shifts—they now range over all words in the alphabet $\mathbb{A}$. The basic operations and relations on words (concatenation, identity $\equiv$, occurrence, and word-for-word replacement $\xi[\zeta/\eta]$) retain their definitions from § I.1 of the PC paper, but now act on words over $\mathbb{A}$; in particular, there are no empty words.

Because the base 7-symbol alphabet of the propositional calculus (with the symbol p for propositional variables and without the external list separator) is contained in $\mathbb{A}$, and the formation rules for formulas and theorems in the PC paper use none of the remaining symbols of $\mathbb{A}$, the classes of propositional variables, formulas, axioms, and theorems of PC embed unchanged into the class of words over $\mathbb{A}$; at the same time, extending the range of the metavariables ξ, η, ζ to all of $\mathbb{A}$ allows arbitrary words of the new language to be substituted into PC formulas via the same basic replacement operation $\xi[\zeta/\eta]$.

The comma in the alphabet $\mathbb{A}$ serves to separate argument positions in templates and terms; moreover, by prefix-freeness of formulas (Lemma 1.7) and the editor property (§ I.1 of the PC paper), the very same comma symbol within the 16-symbol alphabet $\mathbb{A}$ (without extending it) unambiguously separates formulas in external lists of hypotheses and derivation chains (§ 2.3). Argument parentheses of metapredicates continue to be highlighted in blue ($(\,)$), and a compound argument of a predicate is enclosed in quotes. For compact notation of concrete words in the exposition (at the natural-language level, above the special language containing no numbers), exponential shorthand is used for consecutive repetitions of a single symbol of $\mathbb{A}$: a^2 for aa, a^3 for aaa, $\mathfrak{F}^2$ for $\mathfrak{F}\mathfrak{F}$, $\mathfrak{F}^3$ for $\mathfrak{F}\mathfrak{F}\mathfrak{F}$, $\mathfrak{P}^2$ for $\mathfrak{P}\mathfrak{P}$, etc. (the object word itself remains the expanded string of symbols without exponents).

Equality as a logical symbol. The binary predicate symbol $=$ does not belong to the signature: it counts among the logical symbols alongside connectives and quantifiers, is present in the language of every signature, and its axioms (reflexivity and congruence, group 3 of Definition 2.1) are logical axioms. In the semantics (Definition 3.2), the symbol $=$ is always interpreted as the identity relation on the domain and therefore, unlike elements of the signature, is not subject to free interpretation. Unless stated otherwise, *predicate calculus with equality* is considered. *Predicate calculus without equality* is obtained by deleting the symbol $=$ from the formula vocabulary used, the equality clause from rule **F1** of Definition 1.5, and group 3 of the axioms (assuming $\mathrm{Pred}_\Sigma \neq \emptyset$, since otherwise the language contains no atomic formulas and hence no formulas at all); all results of Sections 1–3 not involving equality carry over to it unchanged.

Definition 1.1 (Classes of variables, function stems, and predicate stems)**.** The predicates $\mathrm{PVar}(\xi)$ ("propositional variable of PC"), $\mathrm{FVar}(\xi)$ ("free individual variable of FOL"), $\mathrm{BVar}(\xi)$ ("bound individual variable of FOL"), $\mathrm{FSym}(\xi)$ ("function stem"), and $\mathrm{PSym}(\xi)$ ("predicate stem") are defined recursively:

1. *Propositional variables* (PVar):

P1. *Basis:* $\mathrm{PVar}(\text{'}(\mathsf{p})\text{'})$.

P2. *Rule:* if PVar('(ξ)'), then PVar('(ξp)').

P3. *Closure:* no other propositional variables exist.

2. *Free individual variables* (FVar):

FV1. *Basis:* FVar('(a)').

FV2. *Rule:* if FVar('(ξ)'), then FVar('(ξa)').

FV3. *Closure:* no other free variables exist.

3. *Bound individual variables* (BVar):

BV1. *Basis:* BVar('(x)').

BV2. *Rule:* if BVar('(ξ)'), then BVar('(ξx)').

BV3. *Closure:* no other bound variables exist.

4. *Function stems* (FSym):

FS1. *Basis:* FSym('($\mathfrak{F}$)').

FS2. *Rule:* if FSym('(ξ)'), then FSym('($\xi\mathfrak{F}$)').

FS3. *Closure:* no other function stems exist.

5. *Predicate stems* (PSym):

PS1. *Basis:* PSym('($\mathfrak{P}$)').

PS2. *Rule:* if PSym('(ξ)'), then PSym('($\xi\mathfrak{P}$)').

PS3. *Closure:* no other predicate stems exist.

By induction on the construction, every word of the classes PVar, FVar, BVar, FSym, and PSym has the form (p . . . p), (a . . . a), (x . . . x), ($\mathfrak{F}$ . . . $\mathfrak{F}$), and ($\mathfrak{P}$ . . . $\mathfrak{P}$), respectively, with at least one occurrence of the base symbol. Because the symbols p, a, x, $\mathfrak{F}$, $\mathfrak{P}$ are pairwise distinct, these five classes are pairwise disjoint. Moreover, the parentheses (and) occur in each such word only as its first and last symbol, so no word of any of the five classes PVar, FVar, BVar, FSym, PSym is a proper subword of another word of any of these classes; thus word-for-word replacement $\xi[\zeta/\eta]$ of any such word acts cleanly without affecting the others. Words of the class FSym serve simultaneously as head names of function templates and as constants (realizing constants as 0-ary functions), whereas words of the class PSym are used only as head names of predicate templates.

Once these classes are defined, restricted metavariables are assigned to them:

- to the class PVar—the lowercase Latin letters p, q, r (as in the PC paper);
- to the class FVar—the lowercase Latin letters from the beginning of the alphabet a, b, c, d;
- to the class BVar—the lowercase Latin letters from the end of the alphabet x, y, z, u, v, w;
- to the class FSym—the uppercase Latin letters C, D.

Next, using the classes FSym and PSym, the classes of function and predicate templates are defined within the special language.

Definition 1.2 (Function and predicate templates)**.** The classes FTpl ("function template") and PTpl ("predicate template") are defined recursively:

1. *Function templates* (FTpl):

   FT1. *Basis:* FTpl('$C(_)$').

   FT2. *Rule:* if FTpl('$C(\xi)$'), then FTpl('$C(\xi,_)$').

   FT3. *Closure:* no other function templates exist.

2. *Predicate templates* (PTpl):

   PT1. *Basis:* if PSym(η), then PTpl('$\eta(_)$').

   PT2. *Rule:* if PSym(η) and PTpl('$\eta(\xi)$'), then PTpl('$\eta(\xi,_)$').

   PT3. *Closure:* no other predicate templates exist.

By induction on the construction, the class of function templates FTpl consists of words of the form $(\mathfrak{F}\dots\mathfrak{F})(_,\dots,_)$ (one or more placeholders $_$ separated by commas inside the second pair of parentheses), e.g., $(\mathfrak{F})(_)$, $(\mathfrak{F})(_,_)$, $(\mathfrak{F}^2)(_,_,_)$, and the class of predicate templates PTpl consists of words of the form $(\mathfrak{P}\dots\mathfrak{P})(_,\dots,_)$, e.g., $(\mathfrak{P})(_)$, $(\mathfrak{P}^2)(_,_)$.

To the newly defined classes FTpl and PTpl are assigned the corresponding restricted metavariables:

- to the class FTpl—the uppercase Latin letters F, G, H (so each denotes a function template of the form $C\xi$ already together with its parenthesized string of placeholders);
- to the class PTpl—the uppercase Latin letters P, Q (denoting a predicate template of the form $\eta\xi$ already together with its parenthesized string of placeholders).

The classes FTpl and PTpl are decidable: to test whether a word belongs to FTpl (or PTpl), it suffices to split off the initial word of FSym (respectively PSym), which is uniquely determined by its first pair of symbols and the position of the first closing parenthesis, and verify that the remaining part is a string of alternating symbols $_$ and commas inside a pair of parentheses (and).

Every word of the classes FSym, FTpl, and PTpl begins with a unique word of FSym or PSym—its *head name*: for both a function stem (constant) C and a function template $C\xi$, the head name is the function stem C (built from $\mathfrak{F}$), whereas for a predicate template $\eta\xi$, the head name is the predicate stem η (built from $\mathfrak{P}$). A function stem C without attached placeholders acts as a 0-ary function template (a constant).

Definition 1.3 (Signature)**.** A *signature* Σ is a subclass (selection) of words from the classes FSym, FTpl, and PTpl possessing the *antichain property*: no two distinct words of the signature Σ share a common head name (thus, for each function stem C, the signature Σ contains either the stem C itself as a constant or at most one function template of the form $C\xi$, and for each predicate stem η, at most one predicate template of the form $\eta\xi$). Membership of a word η in the signature Σ is denoted by the predicate $\Sigma(\eta)$.

The choice of signature is made at the metalevel and requires introducing neither a new syntactic separator nor a special class of signatures: in particular, an empty signature (the pure language of equality), a finite selection of words, and (in Section 3, where finitary restrictions are lifted) an infinite antichain of words from FSym, FTpl, and PTpl are all permitted. For a finite (or decidable) selection Σ, testing the predicate $\Sigma(\eta)$ terminates in finitely many steps, which ensures the decidability of the language in Sections 1–2. The antichain property guarantees that for any head name C (or predicate stem η) occurring in the signature Σ, there is exactly one word in Σ with that head name: for example, a single signature cannot simultaneously contain the constant C and the function template $C(_)$, two function templates $C(_)$ and $C(_,_)$, or predicate templates $\eta(_)$ and $\eta(_,_)$. The layout of the basic syntactic classes in the alphabet $\mathbb{A}$ and of the signature antichain Σ is shown in Fig. 1.

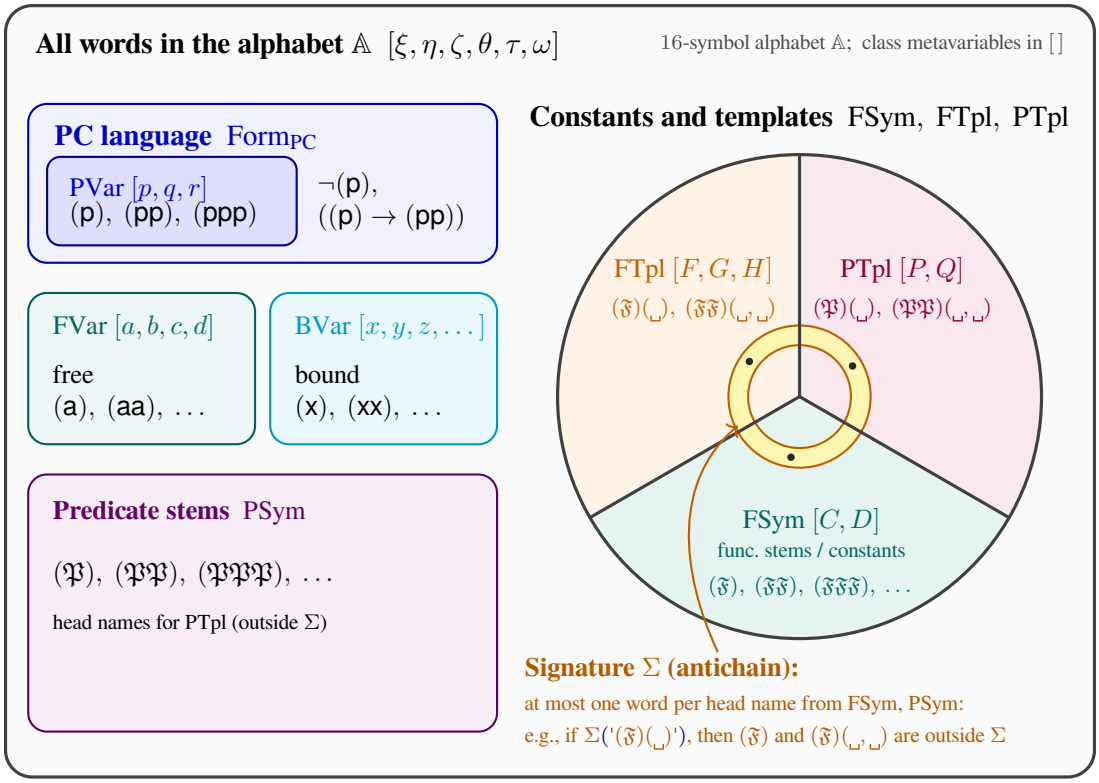


Figure 1: Inclusion of basic syntactic classes in the alphabet $\mathbb{A}$: the propositional fragment (PVar inside $\mathrm{Form}_{\mathrm{PC}}$), variables (FVar, BVar), predicate stems (PSym), the three-sector region of constants and templates (FSym, FTpl, PTpl, where function stems FSym also serve as constants), and a signature Σ (an antichain).

1.2 Terms, Relevant Lists, and Formulas of a Signature

Henceforth, let a signature Σ (in the sense of Definition 1.3) be fixed. Words of the language of this signature fall into two primary syntactic classes: *terms*, denoting objects of the domain, and *formulas*, expressing assertions about those objects. To substitute arguments into function and predicate templates all at once as a list and match its length to the string of placeholders without using numbers, term lists are generated by simultaneous recursion with placeholder strings.

Definition 1.4 (Terms and relevant term lists)**.** The predicate $\mathrm{Term}_\Sigma(\xi)$ ("the word ξ is a term of signature Σ") and the relation $\mathrm{Rel}_\Sigma(\tau;\xi)$ ("the term list τ is relevant to the placeholder string ξ") are defined by simultaneous recursion:

Tm1. *Basis of terms:* $\mathrm{Term}_\Sigma(a)$; if $\Sigma(C)$, then $\mathrm{Term}_\Sigma(C)$.

Tm2. *Single-element term list:* if $\mathrm{Term}_\Sigma(\eta)$, then $\mathrm{Rel}_\Sigma(\text{'}(\eta)\text{'}\,;\,\text{'}(_)\text{'})$.

Tm3. *Appending a term to a list:* if $\mathrm{Term}_\Sigma(\eta)$ and $\mathrm{Rel}_\Sigma(\text{'}(\tau)\text{'}\,;\,\text{'}(\xi)\text{'})$, then $\mathrm{Rel}_\Sigma(\text{'}(\eta,\tau)\text{'}\,;\,\text{'}(_,\xi)\text{'})$.

Tm4. *Applying a function template:* if $\Sigma(\text{'}C\xi\text{'})$ and $\mathrm{Rel}_\Sigma(\tau\,;\,\xi)$, then $\mathrm{Term}_\Sigma(\text{'}C\tau\text{'})$.

Tm5. *Closure:* no other terms and no other pairs in the relation Rel_Σ exist.

The restricted metavariables T, R are assigned to the class Term_Σ.

By induction on the construction (rules **Tm1**–**Tm4**), the general form of words in the class Term_Σ is established: every term of signature Σ is either a free variable of the form $(\mathsf{a}\ldots\mathsf{a})$, or a constant of the form $(\mathfrak{F}\ldots\mathfrak{F})$ from Σ, or a compound term of the form $(\mathfrak{F}\ldots\mathfrak{F})(T_1,\ldots,T_n)$, where the template $(\mathfrak{F}\ldots\mathfrak{F})(_,\ldots,_)$ with the same number of placeholders belongs to Σ and each argument $T_1,\ldots,T_n$ is itself a term of one of these three kinds; no other terms exist. An example of a parse tree of a term is shown in Fig. 2.

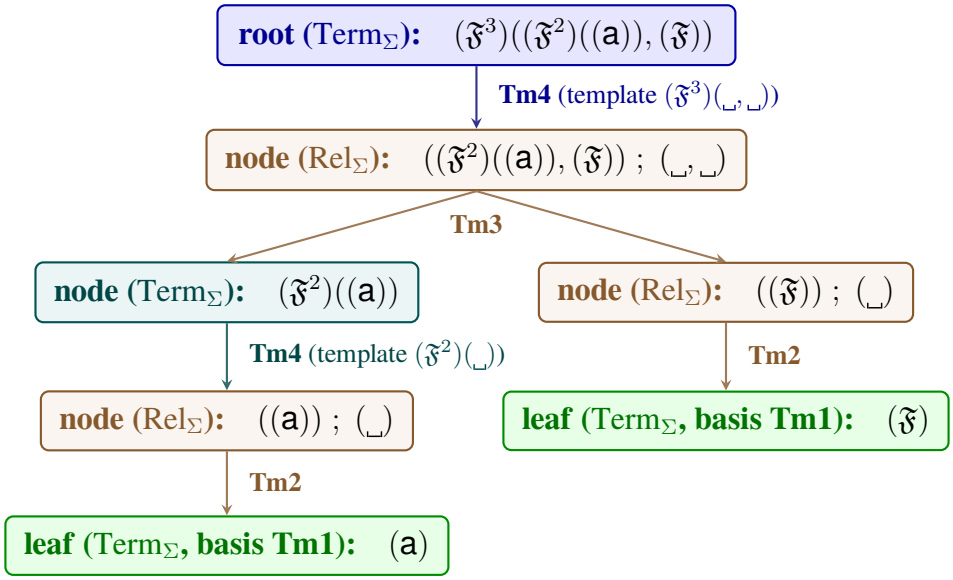


Figure 2: Parse tree of the term $(\mathfrak{F}^3)((\mathfrak{F}^2)((\mathsf{a})),(\mathfrak{F}))$ for $\Sigma(\text{'}(\mathfrak{F})\text{'})$, $\Sigma(\text{'}(\mathfrak{F}^2)(_)\text{'})$, $\Sigma(\text{'}(\mathfrak{F}^3)(_,_)\text{'})$: at the leaves stand the variable (a) and the constant $(\mathfrak{F})$ (basis **Tm1**), while intermediate nodes correspond to relevant lists (**Tm2**, **Tm3**) and applications of function templates (**Tm4**).

Definition 1.5 (Formulas of the language of signature Σ)**.** The predicate $\mathrm{Form}_\Sigma(\xi)$ ("the word ξ is a formula of signature Σ") is defined recursively:

F1. *Atomic formulas:* $\mathrm{Form}_\Sigma(\text{'}(T = R)\text{'})$; if $\mathrm{PSym}(\eta)$, $\Sigma(\text{'}\eta\xi\text{'})$, and $\mathrm{Rel}_\Sigma(\tau\,;\,\xi)$, then $\mathrm{Form}_\Sigma(\text{'}\eta\tau\text{'})$.

F2. *Negation:* if $\mathrm{Form}_\Sigma(\xi)$, then $\mathrm{Form}_\Sigma(\text{'}\neg\xi\text{'})$.

F3. *Binary connectives:* if $\mathrm{Form}_\Sigma(\xi)$ and $\mathrm{Form}_\Sigma(\eta)$, then

$$\mathrm{Form}_\Sigma(\text{'}(\xi \wedge \eta)\text{'}), \qquad \mathrm{Form}_\Sigma(\text{'}(\xi \vee \eta)\text{'}), \qquad \mathrm{Form}_\Sigma(\text{'}(\xi \to \eta)\text{'}).$$

F4. *Quantifiers:* if $\mathrm{Form}_\Sigma(\xi)$ and the variable x **does not occur** in the word ξ, then

$$\mathrm{Form}_\Sigma(\text{'}\forall x\, \xi[x/a]\text{'}), \qquad \mathrm{Form}_\Sigma(\text{'}\exists x\, \xi[x/a]\text{'}).$$

F5. *Closure:* no other formulas exist.

The restricted metavariables φ, ψ, χ are assigned to the class Form_Σ (whereas calligraphic $\mathcal{A}, \mathcal{B}, \mathcal{C}$ continue to refer to formulas of PC). The class of all formulas Form_Σ for a fixed signature Σ is called the *language of signature* Σ and is denoted by $\mathcal{L}_\Sigma$. A diagram showing the inclusion and separation of the syntactic classes of signature Σ (in the notation of Section 2) is given later in § 2.2 in Fig. 4. An example of a parse tree of a formula is shown in Fig. 3.

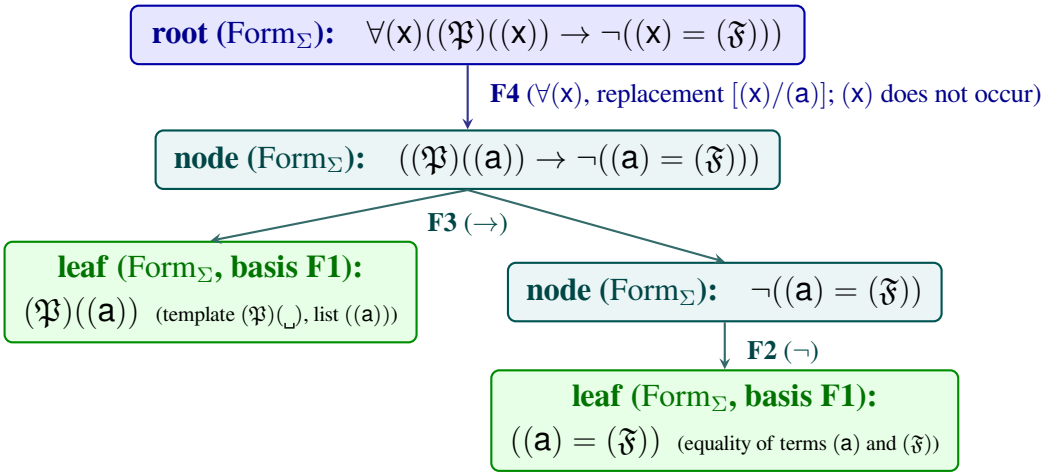


Figure 3: Parse tree of the formula $\forall(\mathsf{x})((\mathfrak{P})((\mathsf{x})) \to \neg((\mathsf{x}) = (\mathfrak{F})))$ for $\Sigma(\text{'}(\mathfrak{F})\text{'})$ and $\Sigma(\text{'}(\mathfrak{P})(_)\text{'})$: at the leaves stand atomic formulas (basis **F1**), which can in turn be parsed by the rules for terms.

Henceforth, the metasigns $\circ, \bullet$ denote an arbitrary binary connective among $\wedge, \vee, \to$, and the metasign Q denotes a quantifier among $\forall, \exists$.

To prove unique readability of terms and formulas, auxiliary classes of words are introduced that admit arbitrary individual variables in argument positions, and disjointness of classes together with prefix-freeness are established.

Definition 1.6 (Quasiterms and quasiformulas)**.** The classes of *quasiterms* Term^*_Σ and *quasiformulas* Form^*_Σ and the relation Rel^*_Σ for lists of quasiterms are defined by the same rules **Tm1**–**Tm5** and **F1**–**F5** of Definitions 1.4 and 1.5 with two relaxations: in **Tm1**, arbitrary individual variables of FVar and BVar are permitted, and in **F4**, for any $x \in \mathrm{BVar}$ and $\psi \in \mathrm{Form}^*_\Sigma$, one sets $\mathrm{Form}^*_\Sigma(\text{'}\forall x\, \psi\text{'})$ and $\mathrm{Form}^*_\Sigma(\text{'}\exists x\, \psi\text{'})$.

By induction on the construction, the classes Term^*_Σ and Form^*_Σ are closed under word-for-word replacement $\xi[u/a]$ of any free variable $a \in \mathrm{FVar}$ by an arbitrary variable $u \in \mathrm{FVar} \cup \mathrm{BVar}$ (such a replacement does not affect quantifier prefixes $\mathsf{Q}x$, where $x \in \mathrm{BVar}$). In particular, for $\psi \in \mathrm{Form}^*_\Sigma$ and $x \in \mathrm{BVar}$, one has $\psi[x/a] \in \mathrm{Form}^*_\Sigma$, whence $\mathrm{Term}_\Sigma \subseteq \mathrm{Term}^*_\Sigma$ and $\mathrm{Form}_\Sigma \subseteq \mathrm{Form}^*_\Sigma$.

Lemma 1.7 (Separation of classes and initial subwords)**.**

(a) *The classes* Term_Σ *and* Form_Σ *are disjoint.*

(b) *No word of the class* Term_Σ *or* Form_Σ *is a proper initial subword of another word of the classes* Term_Σ *and* Form_Σ.

Proof. Because $\mathrm{Term}_\Sigma \subseteq \mathrm{Term}^*_\Sigma$ and $\mathrm{Form}_\Sigma \subseteq \mathrm{Form}^*_\Sigma$, it suffices to prove both statements for Term^*_Σ and Form^*_Σ.

(a) Every word of the classes FVar, BVar, FSym, and PSym begins with the parenthesis (followed by a, x, $\mathfrak{F}$, or $\mathfrak{P}$, respectively. Therefore every quasiterm of Term^*_Σ begins with (whose second symbol is a, x, or $\mathfrak{F}$. Every quasiformula of Form^*_Σ, on the other hand, begins either with one of the symbols $\neg, \forall, \exists$, or with (whose second symbol is $\mathfrak{P}$ (for predicate formulas) or the initial symbol of a quasiterm or quasiformula (for $(T = R)$ and $(\varphi \circ \psi)$), i.e., again one of $(, \neg, \forall, \exists$. Since the first two symbols of a quasiterm and a quasiformula never coincide, the classes Term^*_Σ and Form^*_Σ are disjoint.

(b) By simultaneous structural induction on the construction of words in Term^*_Σ, Rel^*_Σ, and Form^*_Σ, we show that no word ξ of Term^*_Σ or Form^*_Σ is a proper initial subword of another word η of Term^*_Σ or Form^*_Σ, and that for a fixed placeholder string ζ, no list τ with $\mathrm{Rel}^*_\Sigma(\tau;\zeta)$ is a proper initial subword of another list θ with $\mathrm{Rel}^*_\Sigma(\theta;\zeta)$. If one of two words is a proper initial subword of the other, their first two symbols coincide; moreover, all formation rules reduce to three structural mechanisms that preserve prefix-freeness by the properties of words from § I.1 of the PC paper:

1. *Parenthesized basic atoms* (FVar, BVar, FSym, PSym): the closing parenthesis) occurs in each word of these classes only as its last symbol, so none of them can be a proper initial subword of another; in particular, this handles the basis **Tm1** for variables.

2. *Prefix composition* ($\neg\varphi$, $\mathsf{Q}x\,\varphi$, as well as C in **Tm1**, $C\tau$ in **Tm4**, and $\eta\tau$ in **F1**): if the left parts of two words belong to an already prefix-free class, then by prefix-freeness (and left-cancellation of word concatenation) those left parts coincide and cancel, transferring the comparison to the right parts. Specifically:

   - in negations $\neg\varphi$, $\neg\psi$ (**F2**), $\neg$ cancels, and in quantified words $\mathsf{Q}x\,\varphi$, $\mathsf{Q}y\,\psi$ (**F4**), after canceling Q, prefix-freeness of BVar forces $x \equiv y$, which cancels and reduces the task to the induction hypothesis for φ and ψ;
   - in words beginning with ($\mathfrak{F}$ (**Tm1**, **Tm4**) or ($\mathfrak{P}$ (**F1**), prefix-freeness of FSym and PSym forces the head names to coincide, and by the antichain property of the signature Σ (Definition 1.3), the shared head name occurs in Σ within a unique word—either the constant C (in which case both words equal C) or a unique template with placeholder string ζ, in which case canceling the head name reduces the task to the lists τ, θ with the common placeholder string ζ.

3. *Parenthesizing with a separator* (lists '(T)' and '(T,τ)' in **Tm2**–**Tm3**, equalities $(T = R)$ in **F1**, and binary quasiformulas $(\varphi \circ \psi)$ in **F3**): after canceling the outer opening parenthesis (, the first components are initial subwords of the same word and coincide by the induction hypothesis (and by part (a), a quasiterm T from $(T = R)$ cannot coincide with a quasiformula φ from $(\varphi \circ \psi)$, so both compared words are of the same kind). After canceling the first component, in a single-element list '(T)' only the closing parenthesis) remains, precluding proper extension; in equalities and binary quasiformulas,

the separators ($=$ or connective $\circ$) coincide and cancel, after which the induction hypothesis forces the second components and the final parenthesis) to coincide; finally, in compound lists $'(T,\tau)'$ and $'(R,\theta)'$ (**Tm3**), after canceling the first component and the comma, adjoining (on the left returns the comparison to the lists $'(\tau)'$ and $'(\theta)'$ for a shorter placeholder string, completing the induction. □

Proposition 1.8 (Unique readability of terms, lists, and formulas)**.**

(a) *Every term T has exactly one of the following forms: a free variable a; a constant C of signature Σ; a word $C\tau$ for a unique function template $F \equiv C\zeta$ of signature Σ and a unique term list τ satisfying* $\mathrm{Rel}_\Sigma(\tau;\zeta)$*. Moreover, for any list τ satisfying* $\mathrm{Rel}_\Sigma(\tau;\zeta)$*, the terms occurring in it are uniquely recovered: if $\zeta \equiv '(_)'$, then $\tau \equiv '(T)'$ for a unique term T; if $\zeta \equiv '(_,\omega)'$, then $\tau \equiv '(R,\theta)'$ for uniquely determined term R and list $'(\theta)'$ with* $\mathrm{Rel}_\Sigma('(\theta)';'(\omega)')$*.*

(b) *Every formula has exactly one of the following forms:* $\eta\tau$ *(*$\Sigma(P)$*,* $P \equiv \eta\zeta$*,* $\mathrm{Rel}_\Sigma(\tau;\zeta)$*);* $(T = R)$*;* $\neg\varphi$*;* $(\varphi \circ \psi)$ *(*$\circ$ *is one of* $\wedge, \vee, \to$*);* $\forall x\, \chi$*;* $\exists x\, \chi$*. Moreover, all constituents (predicate template P, relevant list τ, terms T, R, connective $\circ$, subformulas φ, ψ, bound variable x, and quantifier body χ) are uniquely determined. In the quantified case, $\chi \equiv \varphi[x/a]$ for some formula $\varphi \in \mathrm{Form}_\Sigma$ and free variable $a \in \mathrm{FVar}$ (and for any free variable b not occurring in χ, one has $\chi[b/x] \equiv \varphi[b/a]$ and $\chi \equiv (\chi[b/x])[x/b]$).*

Proof. For terms, a free variable begins with (𝖺, whereas a constant and a function term begin with (𝔉. In a word $C\tau$, the closing parenthesis of the head name C is immediately followed by the opening parenthesis (of the list τ, whereas in a constant C ($\Sigma(C)$) the word ends at that closing parenthesis (in agreement with the antichain property of Σ, which uniquely determines a word of Σ from its head name). If $'(T,\tau)' \equiv '(R,\theta)'$ with $\mathrm{Rel}_\Sigma('(T,\tau)';'(_,\zeta)')$ and $\mathrm{Rel}_\Sigma('(R,\theta)';'(_,\zeta)')$, then after canceling the initial parenthesis (, the identity $T \equiv R$ follows from Lemma 1.7 (b), after which canceling T and the comma gives $\tau \equiv \theta$, and hence $'(\tau)' \equiv '(\theta)'$. For formulas, the initial symbol pairs (𝔓 (for $\eta\tau$), ((, (¬, (∀, (∃ and the initial symbols $\neg, \forall, \exists$ immediately separate all cases except $(T = R)$ and $(\varphi \circ \psi)$ (in both of these cases the second symbol may be (; they cannot coincide, since after canceling the outer parenthesis, $(T = R)$ begins with the term T whereas $(\varphi \circ \psi)$ begins with the formula φ, which is impossible by Lemma 1.7). Uniqueness of the constituents in each case follows from Lemma 1.7 (b) (for $\forall x\, \chi$ and $\exists x\, \chi$, in the classes Term^*_Σ and Form^*_Σ). In the quantified case, by Definition 1.5 (**F4**) we have $\chi \equiv \varphi[x/a]$, where x does not occur in the formula φ; for any free variable b not occurring in χ, every occurrence of x in χ arose from an occurrence of a in φ, so $\chi[b/x] \equiv \varphi[b/a]$ and $(\chi[b/x])[x/b] \equiv \chi$. □

Lemma 1.9 (Closure of terms and formulas under substitution)**.** *For any terms $R, T \in \mathrm{Term}_\Sigma$, formula $\varphi \in \mathrm{Form}_\Sigma$, and free variable $a \in \mathrm{FVar}$, one has* $\mathrm{Term}_\Sigma('R[T/a]')$ *and* $\mathrm{Form}_\Sigma('\varphi[T/a]')$*, and the word $\varphi[T/a]$ contains the same number of occurrences of connectives and quantifiers ($\neg, \wedge, \vee, \to, \forall, \exists$) as φ.*

Proof. Note first that no term in Term_Σ contains the connective or quantifier symbols $\neg, \wedge, \vee, \to, \forall, \exists$, and words of the classes FVar, BVar, FSym, PSym are not proper subwords of one another, so word-for-word replacements of variables do not affect other symbol classes and create no spurious occurrences.

Closure of Term_Σ under substitution $[T/a]$ (together with closure of $\mathrm{Rel}_\Sigma(\tau;\xi)$ under componentwise replacement $\tau[T/a]$) is proved by simultaneous structural induction on rules **Tm1**–**Tm4**: if $R \equiv a$, then $R[T/a] \equiv T \in \mathrm{Term}_\Sigma$; if $R \not\equiv a$ and R belongs to the basis **Tm1**, then $R[T/a] \equiv R \in \mathrm{Term}_\Sigma$; rules **Tm2**–**Tm4** carry over by the induction hypothesis, since a occurs neither in $_$ nor in head names from FSym.

Closure of Form_Σ and preservation of the number of connectives and quantifiers are proved by induction on the number k of occurrences of the symbols $\neg, \wedge, \vee, \to, \forall, \exists$ in the formula φ (simultaneously for all $a \in \mathrm{FVar}$ and $T \in \mathrm{Term}_\Sigma$):

- $k = 0$ (**F1**): φ has the form $(R_1 = R_2)$ or $\eta\tau$ ($\eta \in \mathrm{PSym}$); by what has already been proved for Term_Σ and Rel_Σ, the word $\varphi[T/a]$ belongs to Form_Σ and contains no connectives or quantifiers.

- Step **F2** ($\varphi \equiv \neg\psi$) and **F3** ($\varphi \equiv (\psi_1 \circ \psi_2)$): the induction hypothesis for ψ, ψ_1, ψ_2 yields $\psi[T/a], \psi_1[T/a], \psi_2[T/a] \in \mathrm{Form}_\Sigma$ with the same number of connectives and quantifiers.

- Step **F4** ($\varphi \equiv \mathsf{Q}y\,\psi[y/b]$, where $\psi \in \mathrm{Form}_\Sigma$ contains $k-1$ occurrences of connectives and quantifiers and $y \in \mathrm{BVar}$ does not occur in ψ): if $b \equiv a$, then a does not occur in $\psi[y/a]$, so $\varphi[T/a] \equiv \varphi \in \mathrm{Form}_\Sigma$. Suppose $b \not\equiv a$.

  - If b does not occur in T, then since $y \in \mathrm{BVar}$ also does not occur in T, the word-for-word replacements $[T/a]$ and $[y/b]$ commute on ψ: $(\psi[y/b])[T/a] \equiv (\psi[T/a])[y/b]$. By the induction hypothesis, $\psi[T/a] \in \mathrm{Form}_\Sigma$ has $k-1$ connectives and quantifiers, and y does not occur in $\psi[T/a]$. By rule **F4**, $\varphi[T/a] \equiv \mathsf{Q}y\,(\psi[T/a])[y/b] \in \mathrm{Form}_\Sigma$ (with k connectives and quantifiers).
  - If b occurs in T, choose a free variable $c \in \mathrm{FVar}$ that occurs neither in ψ nor in T and is distinct from a and b. By the induction hypothesis (at $k-1$) for the substitution $[c/b]$, we have $\psi' \equiv \psi[c/b] \in \mathrm{Form}_\Sigma$ with $k-1$ connectives and quantifiers, y does not occur in ψ', and $\psi[y/b] \equiv (\psi[c/b])[y/c] \equiv \psi'[y/c]$. Because c does not occur in T and $c \not\equiv a$, the previous case applies to $\psi'[y/c]$: $(\psi'[y/c])[T/a] \equiv (\psi'[T/a])[y/c]$ and $\varphi[T/a] \equiv \mathsf{Q}y\,(\psi'[T/a])[y/c] \in \mathrm{Form}_\Sigma$. □

Unique readability and Lemma 1.9 guarantee the decidability of the classes PVar, FVar, BVar, FSym, PSym, FTpl, PTpl, and—when Σ is a decidable selection—also the decidability of Term_Σ, Rel_Σ, and Form_Σ, as well as the well-definedness of definitions of functions and predicates by structural recursion. In particular, for a quantified formula $\mathsf{Q}x\,\chi$, passing to the quantifier body can be done by replacing x with an arbitrary free variable b not occurring in χ, obtaining by Proposition 1.8 (b) and Lemma 1.9 a formula $\chi[b/x] \in \mathrm{Form}_\Sigma$ of strictly smaller complexity (with one fewer connective or quantifier).

Remark 1.10 (Bottom-up reduction and generation via word-for-word replacements)**.** Prefix-freeness and unique readability admit a transparent algorithmic interpretation via the basic operation of word-for-word replacement $\xi[\zeta/\eta]$ from § I.1 of the PC paper, if one chooses the basic variable (a) as the reference term and the propositional variable (p) as the reference atom for formulas. By successive bottom-up replacements along the construction tree, every individual variable and every constant C of signature Σ ($\Sigma(C)$) is replaced by (a); elementary function calls of the form $C((\mathsf{a}),\dots,(\mathsf{a}))$ (for templates in Σ) then collapse back to (a); elementary atomic formulas $\zeta((\mathsf{a}),\dots,(\mathsf{a}))$ and $((\mathsf{a}) = (\mathsf{a}))$ are replaced by (p); and finally the constructs $\neg(\mathsf{p})$, $\mathsf{Q}(\mathsf{a})\,(\mathsf{p})$, and $((\mathsf{p}) \circ (\mathsf{p}))$ are successively reduced to

(p). Consequently, every quasiterm reduces to (a) and every quasiformula reduces to (p). The antichain property of Σ and the closure of parenthesized constructs from the proof of Lemma 1.7 ensure that elementary subwords subject to reduction cannot overlap across the boundary between a quasiterm or quasiformula ξ and a non-empty word ω adjoined on the right: after reducing the initial subword ξ to (a) or (p), the word $\xi\omega$ leaves a word of the form $(\mathsf{a})\theta$ or $(\mathsf{p})\theta$, to which no rule of reduction to a single reference symbol (a) or (p) can apply, since compound constructs always begin with one of $(, \neg, \forall, \exists$ or with a head name from FSym or PSym.

Dually, word-for-word replacement $\xi[T/a]$ makes it possible to generate the exact same classes Term_Σ and Form_Σ without the placeholder symbol ␣ in the alphabet and without the auxiliary relation Rel_Σ: it suffices in the templates FTpl and PTpl to write, instead of the placeholder string $(\text{␣}, \ldots, \text{␣})$, the canonical list of pairwise distinct free variables $((\mathsf{a})), ((\mathsf{a}), (\mathsf{a}^2)), ((\mathsf{a}), (\mathsf{a}^2), (\mathsf{a}^3)), \ldots$ (generated recursively by adjoining the next variable $(\eta\mathsf{a})$ after the last variable (η)), include the templates belonging to Σ directly in the basis of terms and atomic formulas, and generate compound terms and predicate atomic formulas from them by the single-variable substitution rule $\xi \mapsto \xi[T/a]$ (under sequential substitution of multiple arguments, name collisions are avoided via fresh intermediate variables just as in § II.2 of the PC paper). In that representation, however, one and the same term or atom receives infinitely many distinct construction trees, whereas the construction with nameless placeholders ␣ and lists Rel_Σ assembles each compound expression from its immediate subwords in a single step and thereby immediately identifies the construction tree with the parse tree.

1.3 Translating the Formal Canon into the Internal Representation of the Lean Proof Assistant

The mechanisms that emerged in the rigorous construction of FOL syntax over a finite alphabet—the separation of free and bound variables, the prohibition of bound variables occurring free in terms, and the control of arity and unique declaration of signature names—have a close analogue in the internal architecture of modern interactive theorem provers (proof assistants), in particular in the kernel of **Lean** (Lean 4). However, Lean's kernel is typed, and the separation into `bvar` and `fvar` (the locally nameless representation) is a device of internal expression representation rather than a partition of the source text alphabet.

The key difference lies in the boundary between linear text and syntactic trees. In the present exposition, the formal system is built directly from flat strings in the alphabet $\mathbb{A}$, so all issues of separators, parentheses, and unique readability (Lemma 1.7 and Proposition 1.8) are resolved within the syntactic foundation itself. In Lean, parsing a text file with parentheses and commas is relegated to an external parser and elaborator, whereas the trusted logical kernel operates directly on already parsed trees—values of the inductive expression type `Expr`. Within this representation, the canon of Section 1 translates into Lean kernel constructs as follows.

1. Separation of FVar **and** BVar**: locally nameless representation.** In Lean's kernel type `Expr`, variables are likewise separated into two disjoint constructors (the so-called *locally nameless representation*):

- free variables FVar correspond to the constructor `Expr.fvar`, storing a unique free-variable identifier; only such variables appear in open terms and contexts of hypotheses;

- bound variables BVar correspond to the constructor `Expr.bvar` carrying a *de Bruijn index* k, indicating the number of enclosing quantifiers (counted upward in the tree from the given occurrence; e.g., `bvar` 0 refers to the nearest binding quantifier).

In a well-formed Lean expression, just as in formulas of the class Form_Σ of Definition 1.5, no unbound occurrences of `bvar` exist. Rule **F4** of quantifier attachment $\forall x\,\varphi[x/a]$ corresponds in Lean's kernel to the built-in operation `abstract`: it replaces all occurrences of a given free variable `fvar` in the tree φ by the corresponding de Bruijn index `bvar`, and wraps the result in the quantifier constructor `Expr.forallE` (for the existential quantifier $\exists x\,\varphi[x/a]$, the same `abstract` operation is used together with the λ-abstraction constructor `Expr.lam` and the predicate `Exists`). The converse passage—substituting a term T when instantiating a quantifier—corresponds to the operation `instantiate`, which replaces `bvar` 0 by the tree of the term T. Because no `bvar`-closed tree of a term T contains unbound `bvar`s, variable capture by quantifiers during substitution is syntactically impossible, and de Bruijn indices eliminate the need to choose names for bound variables.

2. Templates, relevant lists, and the antichain property: currying and the environment. In place of positional templates from $\mathrm{FTpl} \cup \mathrm{PTpl}$ and the simultaneous recursion $\mathrm{Rel}_\Sigma(\tau;\xi)$, Lean's kernel uses *currying* and a typed declaration table—the `Environment`:

- the domain of discourse is fixed as a type U (of type `Type`), and elements of the signature are declared as constants in the `Environment` (expressions `Expr.const`); the antichain property of Definition 1.3 corresponds to name uniqueness in the environment: each head name may be declared in `Environment` only once with a single type;

- templates of arbitrary length are replaced by iterated function types (built with the same constructor `Expr.forallE` without dependency): a constant $C \in \mathrm{FSym}$ (as a 0-ary function) corresponds to type U, a function template $F \in \mathrm{FTpl}$ of the form $C\xi$ with n placeholders corresponds to type $U \to \cdots \to U$ (n arrows), and a predicate template $P \in \mathrm{PTpl}$ of the form $\eta\xi$ with n placeholders corresponds to type $U \to \cdots \to U \to$ `Prop`;

- relevant term lists $\mathrm{Rel}_\Sigma(\tau;\xi)$ and their attachment to a head name ($C\tau$ for $F \equiv C\xi$) are replaced by nested binary function application `Expr.app`: the expression $C(T_1,\dots,T_n)$ is represented in memory by the tree

$$\mathtt{app}\big(\dots\,\mathtt{app}(\mathtt{app}(C,\,T_1),\,T_2)\dots,\,T_n\big),$$

  and consistency of the number of arguments with the string of placeholders (the condition $\mathrm{Rel}_\Sigma(\tau;\xi)$) is checked by the kernel's type system via the number of arrows in the declared type of the constant.

The table below compares the key constructs of the string canon of Section 1 with their analogues in the Lean 4 kernel.

String canon (Section 1)	**Lean 4 kernel (`Expr`, `Environment`)**
Word in the 16-symbol alphabet $\mathbb{A}$; Lemma 1.7 and Proposition 1.8	Value of tree type `Expr` (text parsing is delegated to an external parser)
Free variable $a \in \mathrm{FVar}$	Constructor `Expr.fvar`
Bound variable $x \in \mathrm{BVar}$	Constructor `Expr.bvar` (de Bruijn index)
Quantifier attachment $\forall x\, \varphi[x/a]$ (**F4**)	Operation `abstract` and constructor `Expr.forallE`
Quantifier instantiation $(\chi[b/x])[T/b]$ ($b \notin \chi$, Prop. 1.8 (b))	Operation `instantiate`
Signature Σ as an antichain in $\mathrm{FSym} \cup \mathrm{FTpl} \cup \mathrm{PTpl}$	Uniquely declared constants `Expr.const` in `Environment`
Number of slots $\sqcup$ in string ξ of template $F \equiv C\xi$ / $P \equiv \eta\xi$	Number of arrows in type $U \to \cdots \to U$ or $U \to \cdots \to$ `Prop`
Applying template $F \equiv C\xi$ to relevant list τ (term $C\tau$)	Chain of binary nodes `Expr.app` with type checking
Equality $(T = R)$, connectives $\neg, \wedge, \vee, \to$	Applications of built-in constants `Eq`, `Not`, `And`, `Or` and constructor `Expr.forallE`

2 The Predicate Calculus

2.1 First-Order Language

In the preceding section, following the finitary canon established in the description of the language of PC, it was shown how the language of first-order predicate calculus is specified over a finite alphabet using the small arsenal of special-language tools defined in Part I of the PC paper. For a clearer overview of the whole picture, a more customary and compact description of the language of FOL is given below, employing set-theoretic notation and freer notational conventions.

Thus, the logical apparatus of FOL is taken to include the standard logical connectives $\neg, \wedge, \vee, \to, \leftrightarrow$ (where $\varphi \leftrightarrow \psi$ abbreviates $(\varphi \to \psi) \wedge (\psi \to \varphi)$), the quantifier symbols $\forall, \exists$, and the equality symbol $=$. As in § 1.2, an arbitrary quantifier ($\forall$ or $\exists$) is denoted by the metasign Q (to distinguish it from the predicate letter Q). It is also assumed that p, q, r range over propositional variables of PC, and $\mathcal{A}, \mathcal{B}, \mathcal{C}, \ldots$ range over formulas of PC.

Further, the letters $a, b, c, \ldots$ from the beginning of the Latin alphabet are retained for arbitrary free individual variables (FVar), and the letters x, y, z and nearby letters from the end of the Latin alphabet are retained for arbitrary bound individual variables (BVar). Let Const_Σ be the set of constant symbols, Func_Σ the set of function symbols, and Pred_Σ the set of predicate symbols. Their union $\Sigma = \mathrm{Const}_\Sigma \cup \mathrm{Func}_\Sigma \cup \mathrm{Pred}_\Sigma$ constitutes the signature of the language under consideration. Contentually, these three sets are specified as selections from the classes FSym, FTpl, and PTpl, so Σ is at most countable (since FSym, FTpl, PTpl consist of words over the finite alphabet $\mathbb{A}$). Disjointness of the three sets $\mathrm{Const}_\Sigma, \mathrm{Func}_\Sigma, \mathrm{Pred}_\Sigma$ holds automatically; the substantive requirement of Definition 1.3 is the antichain condition ("one head name—one arity"): a signature Σ cannot simultaneously contain a constant and a function template, or two templates of different arities, with the same head name.

As before, the metavariables C, D are assigned to constant symbols, F, G, H to function symbols, P, Q to predicate symbols, T, R to terms, and $\varphi, \chi, \psi, \alpha, \beta$ to formulas,

with α, β used primarily for non-logical axioms (thus lowercase Latin letters always denote variables of one sort or another, uppercase Latin letters denote non-logical symbols and terms, and the listed Greek letters—disjoint from the six general word metavariables $\xi, \eta, \zeta, \theta, \tau, \omega$ of § 1.1—denote formulas). Moreover, various accents and numerical subscripts may be attached to any letter symbols to ensure an adequate supply ($a_1, b'', C_1, \hat{D}_0, F_0, G'_4, P_1, Q_8, T_1, R'$, etc.).

Function and predicate symbols have a definite arity ($n \geq 1$, or $n = 0$ for constants), which in the canon of Section 1 is specified by a string of placeholders or a list of distinct free variables in parentheses immediately following the symbol: $F(a), G(a, b), P(a, b, c)$, etc. Accordingly, the exposition employs abbreviated notation with numerical arity n and argument list $x_1, \ldots, x_n$ or $\bar{x}$, as well as the standard shorthand for quantified formulas: for a formula $\varphi(a)$ with free variable a (and in general with variables $a_1, \ldots, a_n$), the notation $\forall x\, \varphi(x)$ (respectively $\exists x\, \varphi(x)$, $Q(x_1, \ldots, x_n)$, $F(\bar{x})$) stands for the canonical formula $\forall x\, \varphi[x/a]$ (respectively $\exists x\, \varphi[x/a]$, $(Q(a_1, \ldots, a_n))[x_1/a_1] \ldots [x_n/a_n]$, $(F(a_1, \ldots, a_n))[\bar{x}/\bar{a}]$). Expressions $Q(x_1, \ldots, x_n)$, $F(\bar{x})$, $\varphi(x)$ with bound variables are not themselves formulas or terms and are used solely within the scope of a matching quantifier prefix as shorthand expanded by the indicated substitutions. In general, when the specific arity of a symbol is not fixed, notation such as $F(a, \ldots, b)$, $P(a, \ldots, b)$ is used, or simply the head name F or P is written.

The construction of the set of all terms Term_Σ broadly follows Definition 1.4, while adopting the idea of Remark 1.10, according to which one and the same term admits many different construction trees:

Tm1. every free variable and every constant is a term;

Tm2. if $F(a, \ldots, b)$ is a function symbol of signature Σ written with pairwise distinct free variables $a, \ldots, b$, then $F(a, \ldots, b)$ is a term;

Tm3. if R and T are terms and a is a free variable, then $R[T/a]$ is a term;

Tm4. no other terms exist.

The set of formulas Form_Σ in signature Σ is defined analogously:

Fm1. if T, R are terms, then $(T = R)$ is a formula;

Fm2. if $P(a, \ldots, b)$ is a predicate symbol of signature Σ written with pairwise distinct free variables $a, \ldots, b$, then $P(a, \ldots, b)$ is a formula;

Fm3. if φ is a formula, T is a term, and a is a free variable, then $\varphi[T/a]$ is a formula;

Fm4. if φ, ψ are formulas, then the following are also formulas:

$$\neg\varphi,\ (\varphi \wedge \psi),\ (\varphi \vee \psi),\ (\varphi \to \psi),\ \forall x\ \varphi[x/a],\ \exists x\ \varphi[x/a],$$

where a is an arbitrary free variable and the bound variable x does not occur in φ;

Fm5. no other formulas exist.

Formulas containing no connectives or quantifiers, i.e., equalities of terms or formulas obtained by substituting terms into predicate symbols (without applying rule **Fm4**), are called *atomic*.

A free variable *occurs* in a term or formula if it is a subword of that word. The quantifier $\forall x\,\varphi[x/a]$ "closes" the variable a: it does not occur in the resulting formula. A term or formula in which no free variable occurs is called *closed*; a closed formula is also called a *sentence*, and the class of sentences is denoted by Sent.

Because the classes FVar and BVar are disjoint, substituting a term T for a free variable a in a term R or formula φ coincides with word-for-word replacement $R[T/a]$ and $\varphi[T/a]$ and cannot cause variable capture by quantifiers. For the same reason, permitting rule **Fm3** for arbitrary rather than only atomic formulas does not enlarge the class Form_Σ (by Lemma 1.9, Form_Σ is closed under substitutions $\varphi[T/a]$): the redundancy of rules **Tm3** and **Fm3** merely produces multiple construction trees for the same expression. For example, for a binary function symbol $F(a,b)$, the term $F(C,D)$ can be obtained from $F(a,b)$ either in the order $(F(a,b)[C/a])[D/b]$ or in the order $(F(a,b)[D/b])[C/a]$ (as well as via any fresh intermediate variables or vacuous substitutions $\xi[T/c]$ when c does not occur in ξ). Similarly, the formula $(P(C)\wedge P(D))$ is built either by rule **Fm4** from the atomic formulas $P(C)$ and $P(D)$, or by rule **Fm3** from $(P(a)\wedge P(b))$ via consecutive substitutions $[C/a]$ and $[D/b]$. Meanwhile, unique readability of every formula is preserved by Proposition 1.8.

2.2 Axioms and Rules of Inference

The predicate calculus is built over formulas of a fixed signature Σ in Hilbert style: to the propositional axioms and Modus Ponens are added quantifier schemas, equality axioms, and quantifier introduction rules (Bernays' rules).

Definition 2.1 (The predicate "to be an axiom of FOL")**.** A word ξ has the property Ax (written $\mathrm{Ax}(\xi)$) and is called an *axiom of FOL* if it is a formula and is identical to one of the following formulas (for arbitrary formulas φ,ψ,χ, propositional variables p,q,r, free variable a, bound variable x, and terms T,R,T_0):

Ax1. *Propositional axiom schemas:* $\mathcal{A}[\varphi/p][\psi/q][\chi/r]$, where $\mathcal{A}$ is an axiom of PC (since PVar and Form_Σ are disjoint, replacements do not affect φ,ψ,χ, and the condition $\xi\in\mathrm{Form}_\Sigma$ ensures that all propositional variables in $\mathcal{A}$ have been replaced).

Ax2. *Quantifier axiom schemas* (under the condition that the bound variable x does not occur in φ; well-formedness $\varphi[T/a]\in\mathrm{Form}_\Sigma$ is guaranteed by Lemma 1.9):

$$\forall x\,\varphi[x/a]\to\varphi[T/a],\qquad \varphi[T/a]\to\exists x\,\varphi[x/a].$$

Ax3. *Equality axiom schemas:*

- *reflexivity:* $T=T$;
- *congruence for terms:* $(T=R)\to(T_0[T/a]=T_0[R/a])$;
- *congruence for formulas:* $(T=R)\to(\varphi[T/a]\to\varphi[R/a])$.

No other words have the property Ax. All axioms of groups Ax1–Ax3 are *logical*: they are adopted for every signature and are not included in lists of non-logical axioms of specific theories. In predicate calculus without equality, group 3 is absent.

Remark 2.2 (Atomic form of the congruence axioms)**.** In some expositions (for example, in formal arithmetic, where the axiom $a = b \to S(a) = S(b)$ is written out for the successor operation S), congruence is postulated only for atomic expressions—the function and predicate symbols of the signature (and for equality itself)—after which the general schemas for arbitrary terms T_0 and formulas φ are derived by structural induction, including induction on the number of arguments. The schemas of group 3 of Definition 2.1 are equivalent to such a specification and require no enumeration of signature symbols or their arities.

For a finite signature Σ, the class of axioms Ax is decidable: for any word ξ, parsing by Proposition 1.8 checks whether ξ is a formula, after which membership in the axioms is decided in finitely many steps. Specifically: for the propositional schemas **Ax1** and reflexivity $(T = T)$, checking reduces to direct matching of subformulas and subterms along the parse tree (Proposition 1.8); for the quantifier schemas **Ax2** (of the form $\forall x\, \chi \to \theta$ or $\theta \to \exists x\, \chi$), the formula $\varphi \equiv \chi[b/x]$ is recovered using the shortest variable $b \in \mathrm{FVar}$ not occurring in χ, whereupon if $b \notin \varphi$ one requires $\theta \equiv \varphi$, and if b occurs in φ the term T is uniquely read off at the position of the first occurrence of b; for the congruence schemas **Ax3**, the terms T, R are given by the antecedent $(T = R)$, while all candidate templates T_0 or φ (with a single fresh variable a) correspond to the finitely many subsets of occurrences of T in the left-hand side of the consequent and are checked in finitely many steps.

Definition 2.3 (The predicate “to be a theorem of FOL”)**.** The predicate $\mathrm{Thm}(\xi)$ is defined recursively:

T1. *Basis:* if $\mathrm{Ax}(\varphi)$, then $\mathrm{Thm}(\varphi)$.

T2. *Rule of Modus Ponens:* if $\mathrm{Thm}(\varphi)$ and $\mathrm{Thm}(\text{'}\varphi \to \psi\text{'})$, then $\mathrm{Thm}(\psi)$.

T3. *Bernays' rules:*

- if $\mathrm{Thm}(\text{'}\varphi \to \psi\text{'})$, a does not occur in φ, and x does not occur in ψ, then $\mathrm{Thm}(\text{'}\varphi \to \forall x\, \psi[x/a]\text{'})$;
- if $\mathrm{Thm}(\text{'}\varphi \to \psi\text{'})$, a does not occur in ψ, and x does not occur in φ, then $\mathrm{Thm}(\text{'}\exists x\, \varphi[x/a] \to \psi\text{'})$.

T4. *Closure:* no other words have the property Thm.

A construction for the predicate Thm is called a *derivation* (or *proof*) in FOL.

The mutual arrangement of the syntactic classes of a fixed signature Σ—terms (Term_Σ), bound variables (BVar), and formulas (Form_Σ), including sentences (Sent), axioms (Ax), and theorems (Thm)—is shown in Fig. 4.

Proposition 2.4 (Derived generalization rule)**.** *If* $\mathrm{Thm}(\psi)$*, then* $\mathrm{Thm}(\text{'}\forall x\, \psi[x/a]\text{'})$ *for any free variable* a *and any bound variable* x *not occurring in* ψ*.*

Proof. Choose a free variable b distinct from a and set $\psi' \equiv \psi[b/a]$; then a does not occur in ψ'. The formula $\varphi \equiv \text{'}\psi' \to (\psi' \to \psi')\text{'}$ is a propositional axiom (obtained by substituting $[\psi'/p][\psi'/q]$ into $p \to (q \to p)$), so $\mathrm{Thm}(\varphi)$ and a does not occur in φ. By affirmation of the consequent $\psi \to (\varphi \to \psi)$ and Modus Ponens, $\mathrm{Thm}(\psi)$ implies $\mathrm{Thm}(\text{'}\varphi \to \psi\text{'})$. Since a does not occur in φ and x does not occur in ψ, Bernays' rule **T3** yields $\mathrm{Thm}(\text{'}\varphi \to \forall x\, \psi[x/a]\text{'})$. Applying Modus Ponens to $\mathrm{Thm}(\varphi)$ completes the derivation of $\mathrm{Thm}(\text{'}\forall x\, \psi[x/a]\text{'})$. □

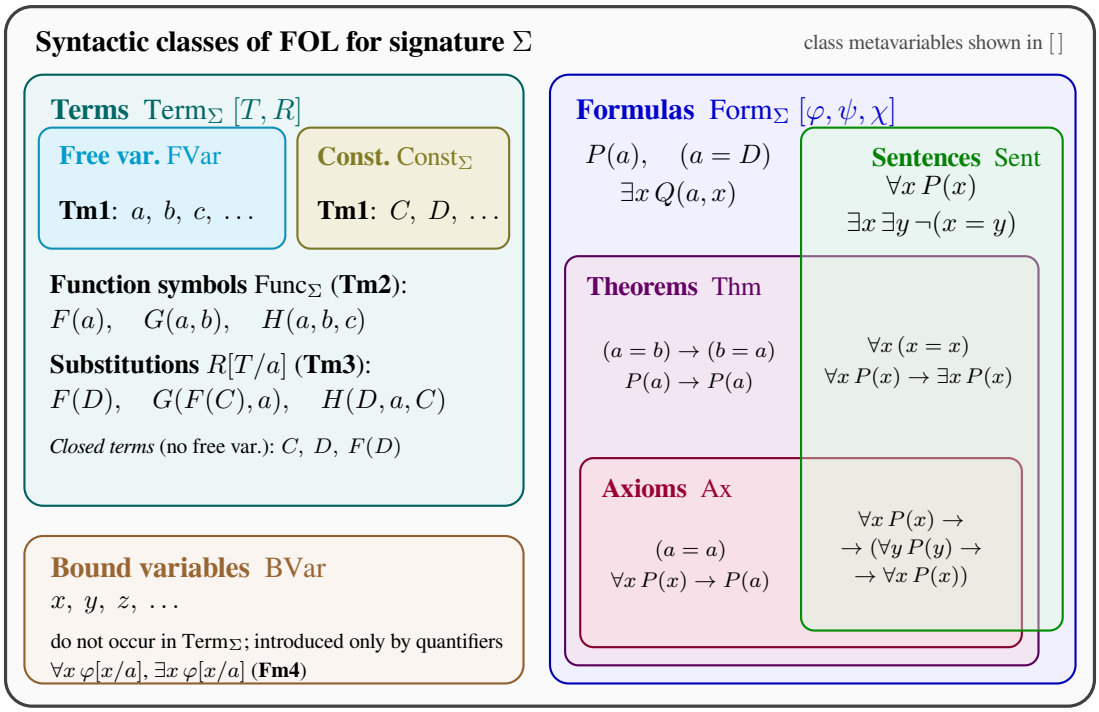


Figure 4: Syntactic classes of first-order predicate calculus for a fixed signature Σ: the disjoint classes of terms (Term_Σ), bound variables (BVar), and formulas (Form_Σ), highlighting sentences (Sent), axioms (Ax), and theorems (Thm).

Every propositional tautology into which a substitution yields a formula of signature Σ is derivable in FOL from group Ax1 and rule **T2** alone (since the propositional part coincides with PC). For instance, $\mathrm{Thm}(\text{'}\forall x\,\varphi[x/a] \to \exists x\,\varphi[x/a]\text{'})$ (where x does not occur in φ) is derived from the two quantifier axioms $\forall x\,\varphi[x/a] \to \varphi$ and $\varphi \to \exists x\,\varphi[x/a]$ (here $T \equiv a$) using the propositional theorem—the law of syllogism $((\chi_1 \to \chi_2) \to ((\chi_2 \to \chi_3) \to (\chi_1 \to \chi_3)))$—and two applications of rule **T2** (Modus Ponens).

Example of a derivation tree. As an example of a complete derivation in FOL using exclusively rules **T1**–**T3** without derived tautologies, consider the derivation tree of the theorem $\exists y\,\forall x\,P(x,y) \to \forall x\,\exists y\,P(x,y)$ (for distinct a, b and distinct x, y; Fig. 5). Unlike the derivation tree in the PC paper, it is oriented in the manner customary in logic: the axioms (leaves) stand at the top, and the theorem being proved (root) stands at the bottom.

Remark 2.5 (Undecidability of the class of theorems of FOL)**.** Unlike the propositional calculus (where the class Thm is decidable by truth tables), in the predicate calculus for a signature containing at least one binary predicate symbol (or a sufficient stock of function symbols), there exists no algorithm that determines in finitely many steps, for an arbitrary formula φ, whether $\mathrm{Thm}(\varphi)$ holds (the Church–Turing theorem). At the same time, the class Thm itself remains recursively enumerable: an exhibited derivation is checked in finitely many steps.

2.3 Derivability in the Context of a List of Hypotheses

To formalize a derivation from a finite list of hypotheses as a single word, in § II.3 of the PC paper the base 7-symbol alphabet was extended to $\mathbb{A}'$ by a fresh separator symbol—

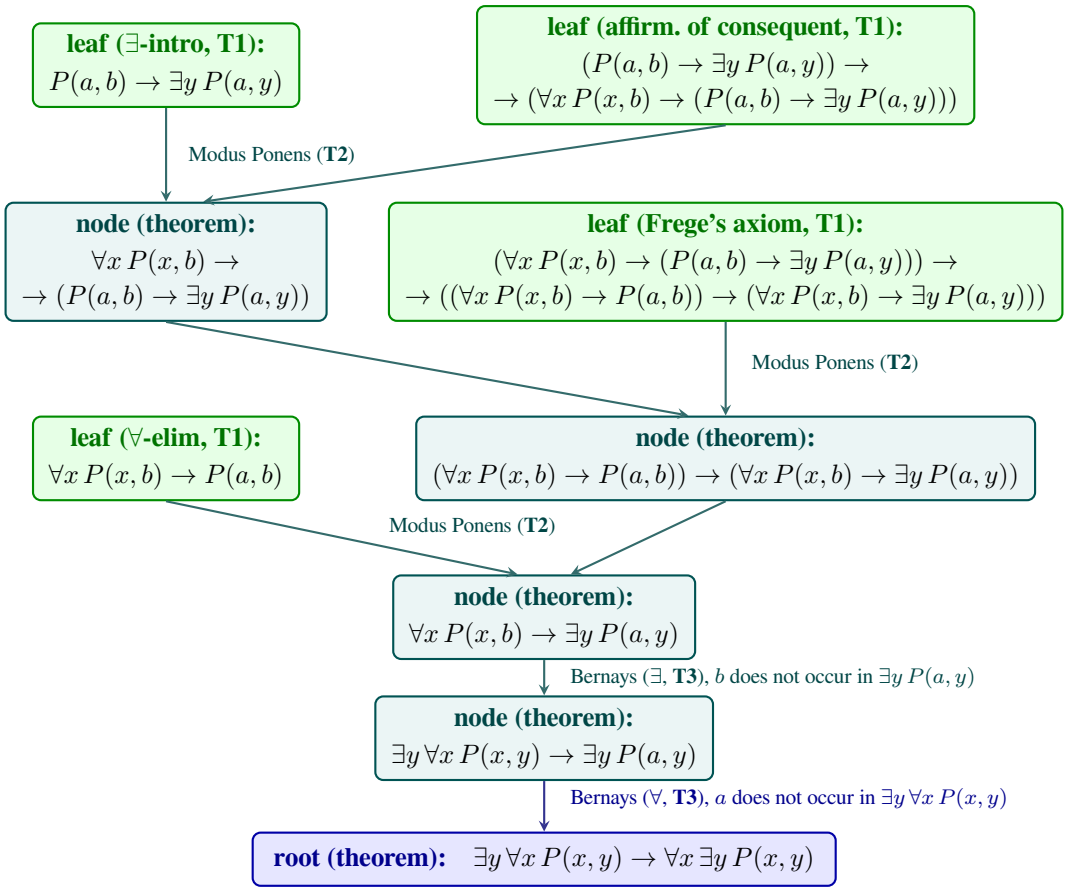


Figure 5: Derivation tree of the theorem $\exists y \forall x\, P(x,y) \to \forall x \exists y\, P(x,y)$ with the root at the bottom: at the top stand four axioms (basis **T1**), three Modus Ponens steps (**T2**) yield $\forall x\, P(x,b) \to \exists y\, P(a,y)$, and two applications of Bernays' rules (**T3**) attach the quantifiers.

the comma. In the first-order language, the comma already belongs to the 16-symbol alphabet $\mathbb{A}$ (inside terms and atomic formulas); nevertheless, by prefix-freeness of formulas (Lemma 1.7 (b)) and the editor property (§ I.1 of the PC paper), any external list of formulas $\varphi_1, \ldots, \varphi_k$ (as well as a sequence of formulas in a derivation) is uniquely parsed from left to right without introducing a 17th separator symbol. Hereafter the comma is also used to write lists of hypotheses of the form Γ, ψ (for the empty context Γ, the notation Γ, ψ stands for the single-element list ψ). Componentwise replacement $\Gamma[b/a]$ in a list Γ is defined inductively: $\psi[b/a]$ for a single-element list and $(\Gamma, \psi)[b/a] \equiv \Gamma[b/a], \psi[b/a]$ (for the empty context, $\Gamma[b/a] \equiv \Gamma$).

Definition 2.6 (List of formulas and derivability in a context)**.** The predicate $\mathrm{List}(\Gamma)$ and the relation $\varphi \trianglelefteq \Gamma$ ("formula φ occurs in list Γ") are defined by simultaneous recursion:

L1. *Single-element list:* $\mathrm{List}(\psi)$ and $\psi \trianglelefteq \psi$.

L2. *Appending a formula:* if $\mathrm{List}(\Gamma)$, then $\mathrm{List}('\Gamma, \psi')$ and $\psi \trianglelefteq \Gamma, \psi$; if moreover $\varphi \trianglelefteq \Gamma$, then $\varphi \trianglelefteq \Gamma, \psi$.

L3. *Closure:* no other lists and no other pairs of the relation $\trianglelefteq$ exist.

As in PC, the metavariables Γ, Δ range over lists of formulas and may also denote the empty context (in which no formula occurs). For a fixed context Γ, the predicate $\Gamma \vdash \varphi$ ("formula φ is derivable in context Γ") is defined recursively:

D1. *Axiom basis:* if $\mathrm{Ax}(\varphi)$, then $\Gamma \vdash \varphi$.

D2. *Hypothesis basis:* if $\varphi \trianglelefteq \Gamma$, then $\Gamma \vdash \varphi$.

D3. *Modus Ponens:* if $\Gamma \vdash \varphi$ and $\Gamma \vdash \varphi \to \psi$, then $\Gamma \vdash \psi$.

D4. *Bernays' rules in a context:*

- if $\Gamma \vdash \varphi \to \psi$, a occurs neither in any formula of Γ nor in φ, and x does not occur in ψ, then $\Gamma \vdash \varphi \to \forall x\, \psi[x/a]$;
- if $\Gamma \vdash \varphi \to \psi$, a occurs neither in any formula of Γ nor in ψ, and x does not occur in φ, then $\Gamma \vdash \exists x\, \varphi[x/a] \to \psi$.

D5. *Closure:* no other formulas are derivable in context Γ.

The condition that a does not occur in the formulas of Γ in rule **D4** is required to preserve semantic soundness: without it, from the hypothesis $P(a)$ (that is, $P(a) \vdash (b = b) \to P(a)$) one would derive $\forall x\, P(x)$, even though property P for a single element does not entail that it holds for all elements. If a does not occur in Γ and x does not occur in ψ, then $\Gamma \vdash \psi$ yields $\Gamma \vdash \forall x\, \psi[x/a]$ by the same device as in Proposition 2.4.

Lemma 2.7 (Renaming of free variables in a derivation)**.** *Let $\mathcal{D}$ be a derivation of a formula φ in context Γ ($\Gamma \vdash \varphi$), let $a \in$ FVar be an arbitrary free variable, and let the free variable $b \in$ FVar occur in no formula of $\mathcal{D}$ (in particular, not in Γ or φ). Then $\Gamma[b/a] \vdash \varphi[b/a]$, and there exists a derivation $\mathcal{D}'$ of this statement whose length does not exceed the length of $\mathcal{D}$.*

Proof. For $b \equiv a$ the statement is trivial ($\Gamma[a/a] \equiv \Gamma$, $\varphi[a/a] \equiv \varphi$). Assume $b \not\equiv a$; proceed by induction on the number of steps (length) of the derivation $\mathcal{D}$.

- *Axiom basis* (**D1**): if $\mathrm{Ax}(\varphi)$, then $\mathrm{Ax}(\text{'}\varphi[b/a]\text{'})$. Indeed:
  - for schema **Ax1** ($\varphi \equiv \mathcal{A}[\psi_1/p][\psi_2/q][\psi_3/r]$), since PVar and FVar are disjoint, we have $\varphi[b/a] \equiv \mathcal{A}[\psi_1[b/a]/p][\psi_2[b/a]/q][\psi_3[b/a]/r]$, where $\psi_i[b/a] \in \mathrm{Form}_\Sigma$ by Lemma 1.9;
  - for schema **Ax2** ($\forall x\, \psi[x/c] \to \psi[T/c]$ or $\psi[T/c] \to \exists x\, \psi[x/c]$ with $x \notin \psi$) and both congruence schemas **Ax3** ($(T = R) \to (T_0[T/c] = T_0[R/c])$ and $(T = R) \to (\psi[T/c] \to \psi[R/c])$) with parameter $c \in$ FVar, note that if $c \equiv a$, then a does not occur in $\forall x\, \psi[x/a]$ (or $\exists x\, \psi[x/a]$), while b occurs neither in ψ nor in T_0, whence $(\psi[T/a])[b/a] \equiv \psi[T[b/a]/a]$ and $(T_0[T/a])[b/a] \equiv T_0[T[b/a]/a]$ (and similarly for R), which by Lemma 1.9 again yields an axiom of the same schema. If instead $c \not\equiv a$, then without loss of generality $c \not\equiv a, b$: indeed, if $c \equiv b$, it suffices to choose a fresh variable $d \in$ FVar occurring in none of T_0, ψ, T, R and distinct from a, b, and rewrite the same axiom with parameter d, since $T_0[T/c] \equiv (T_0[d/c])[T/d]$, $\psi[T/c] \equiv (\psi[d/c])[T/d]$ (and similarly for R and $[x/c]$). When $c \not\equiv a, b$, the word-for-word replacement $[b/a]$ commutes with $[x/c]$ and satisfies $(\psi[T/c])[b/a] \equiv (\psi[b/a])[T[b/a]/c]$, $(T_0[T/c])[b/a] \equiv (T_0[b/a])[T[b/a]/c]$ (and similarly for R); by Lemma 1.9, $\psi[b/a] \in \mathrm{Form}_\Sigma$ (without x) and $T_0[b/a], T[b/a], R[b/a] \in \mathrm{Term}_\Sigma$, so $\varphi[b/a] \in \mathrm{Ax}$. Finally, $(T = T)[b/a] \equiv (T[b/a] = T[b/a]) \in \mathrm{Ax}$.
- *Hypothesis basis* (**D2**): if $\varphi \trianglelefteq \Gamma$, then by the definition of componentwise replacement $\varphi[b/a] \trianglelefteq \Gamma[b/a]$, whence $\Gamma[b/a] \vdash \varphi[b/a]$.

- *Modus Ponens* (**D3**): if the step passes from χ and $\chi \to \varphi$ to φ, then by the induction hypothesis $\Gamma[b/a] \vdash \chi[b/a]$ and $\Gamma[b/a] \vdash \chi[b/a] \to \varphi[b/a]$, whence rule **D3** yields $\Gamma[b/a] \vdash \varphi[b/a]$.

- *Bernays' rules* (**D4**): consider a step $\Gamma \vdash \chi_1 \to \forall x\, \chi_2[x/c]$ from the premise $\Gamma \vdash \chi_1 \to \chi_2$, where c does not occur in Γ or χ_1, and x does not occur in χ_2 (the second rule **D4** is symmetric).

  - If $c \equiv a$, then a occurs in none of Γ, χ_1, and $\forall x\, \chi_2[x/a]$. Consequently, $\Gamma[b/a] \equiv \Gamma$ and $(\chi_1 \to \forall x\, \chi_2[x/a])[b/a] \equiv \chi_1 \to \forall x\, \chi_2[x/a]$, so the original derivation $\mathcal{D}$ itself supplies the required derivation.
  - If $c \not\equiv a$ and $c \not\equiv b$, then by the induction hypothesis applied to the subderivation of the premise, $\Gamma[b/a] \vdash \chi_1[b/a] \to \chi_2[b/a]$. Since $c \not\equiv b$ and c does not occur in Γ or χ_1, the variable c does not occur in $\Gamma[b/a]$ or $\chi_1[b/a]$; the variable $x \in \mathrm{BVar}$ does not occur in $\chi_2[b/a]$. Rule **D4** gives $\Gamma[b/a] \vdash \chi_1[b/a] \to \forall x\, (\chi_2[b/a])[x/c]$. Because $c \not\equiv a, b$ and $x \not\equiv a, b$, the replacements $[b/a]$ and $[x/c]$ commute: $(\chi_2[b/a])[x/c] \equiv (\chi_2[x/c])[b/a]$, which yields $\Gamma[b/a] \vdash (\chi_1 \to \forall x\, \chi_2[x/c])[b/a]$.
  - If $c \equiv b$, then since b occurs in no formula of $\mathcal{D}$, the variable b does not occur in χ_2, whence $\chi_2[x/b] \equiv \chi_2$ and $(\chi_1 \to \forall x\, \chi_2[x/b])[b/a] \equiv \chi_1[b/a] \to \forall x\, \chi_2[b/a]$. By the induction hypothesis, $\Gamma[b/a] \vdash \chi_1[b/a] \to \chi_2[b/a]$. Choosing a fresh variable $e \in \mathrm{FVar}$ occurring in none of $\Gamma[b/a]$, $\chi_1[b/a]$, and $\chi_2[b/a]$, and applying rule **D4** (noting $(\chi_2[b/a])[x/e] \equiv \chi_2[b/a]$), we obtain $\Gamma[b/a] \vdash \chi_1[b/a] \to \forall x\, \chi_2[b/a]$. □

Corollary 2.8 (α-renaming of bound variables)**.** *For any formula $\varphi \in \mathrm{Form}_\Sigma$, free variable $a \in \mathrm{FVar}$, and bound variables $x, y \in \mathrm{BVar}$ not occurring in φ, the following equivalences are derivable in the calculus:*

$$\vdash \forall x\, \varphi[x/a] \leftrightarrow \forall y\, \varphi[y/a], \qquad \vdash \exists x\, \varphi[x/a] \leftrightarrow \exists y\, \varphi[y/a].$$

In particular, for an arbitrary quantified formula $\mathsf{Q}x\, \chi$ and a bound variable $y \in \mathrm{BVar}$ not occurring in χ, one has $\vdash \mathsf{Q}x\, \chi \leftrightarrow \mathsf{Q}y\, \chi[y/x]$.

Proof. By the quantifier axiom **Ax2** (with $T \equiv a$), $\vdash \forall x\, \varphi[x/a] \to \varphi$. Since the free variable a does not occur in $\forall x\, \varphi[x/a]$ and y does not occur in φ, Bernays' rule **T3** yields $\vdash \forall x\, \varphi[x/a] \to \forall y\, \varphi[y/a]$. Symmetrically, the converse implication $\vdash \forall y\, \varphi[y/a] \to \forall x\, \varphi[x/a]$ is derivable. The case of $\exists$ is dual (from $\vdash \varphi \to \exists y\, \varphi[y/a]$ by the second rule **T3**). The second statement follows by choosing a free variable b not occurring in χ: by Proposition 1.8 (b) and Lemma 1.9, the formula $\varphi \equiv \chi[b/x]$ contains neither x nor y, and satisfies $\varphi[x/b] \equiv \chi$ and $\varphi[y/b] \equiv \chi[y/x]$. □

Definition 2.9 (α-equivalence of formulas)**.** The relation of *α-equivalence* $\varphi \approx \psi$ on Form_Σ is defined as the least equivalence relation containing all pairs of the form $\mathsf{Q}x\, \chi \approx \mathsf{Q}y\, \chi[y/x]$ (where $\mathsf{Q}x\, \chi \in \mathrm{Form}_\Sigma$ and $y \in \mathrm{BVar}$ does not occur in χ) and closed under connectives ($\varphi \approx \psi \implies \neg\varphi \approx \neg\psi$, $(\varphi \circ \chi) \approx (\psi \circ \chi)$, $(\chi \circ \varphi) \approx (\chi \circ \psi)$) and the quantifier step (if $\varphi \approx \psi$ and $x \in \mathrm{BVar}$ occurs in neither φ nor ψ, then $\mathsf{Q}x\, \varphi[x/a] \approx \mathsf{Q}x\, \psi[x/a]$).

Lemma 2.10 (Provable equivalence of α-equivalent formulas)**.** *If $\varphi \approx \psi$, then $\vdash \varphi \leftrightarrow \psi$.*

Proof. Induction on the generation of $\approx$: reflexivity, symmetry, and transitivity of $\leftrightarrow$ follow from propositional tautologies, the basic pairs $\mathsf{Q}x\, \chi \approx \mathsf{Q}y\, \chi[y/x]$ are derivable by Corollary 2.8, and preservation of provable equivalence under connectives and quantifiers is given by Proposition 2.13 (a). □

Proposition 2.11 (Relation to the empty context)**.**

(a) $\mathrm{Thm}(\varphi)$ *if and only if* $\vdash \varphi$.

(b) *If* $\mathrm{Thm}(\varphi)$*, then* $\Gamma \vdash \varphi$ *for every context* Γ.

Proof. Part (a) is immediate from the coincidence of rules **T1**–**T3** with rules **D1**, **D3**, **D4** when Γ is empty (in which no free variable occurs). Part (b) is a special case of monotonicity (Proposition 2.12 (a) with empty initial context, whose proof relies only on Lemma 2.7). □

Proposition 2.12 (Monotonicity and the Deduction Theorem)**.**

(a) (Monotonicity in the context.) *If* $\Gamma \vdash \varphi$ *and every formula occurring in* Γ *occurs in context* Δ*, then* $\Delta \vdash \varphi$.

(b) (Deduction Theorem.) $\Gamma, \varphi \vdash \psi$ *if and only if* $\Gamma \vdash \varphi \to \psi$.

Proof. (a) Proceed by induction on the length of the derivation $\mathcal{D}$ of $\Gamma \vdash \varphi$. Steps **D1**–**D3** transfer directly to context Δ (in **D2**, $\varphi \trianglelefteq \Gamma$ entails $\varphi \trianglelefteq \Delta$ by hypothesis). Suppose the last step is rule **D4**: $\Gamma \vdash \chi_1 \to \forall x\, \chi_2[x/a]$ from a subderivation $\mathcal{D}'$ of $\Gamma \vdash \chi_1 \to \chi_2$, where a does not occur in Γ or χ_1 and x does not occur in χ_2 (the second rule **D4** is symmetric). Choose a free variable $b \in \mathrm{FVar}$ occurring in no formula of the subderivation $\mathcal{D}'$ or of the list Δ. By Lemma 2.7 applied to $\mathcal{D}'$ with replacement $[b/a]$, since a does not occur in Γ or χ_1, we obtain a derivation of $\Gamma \vdash \chi_1 \to \chi_2[b/a]$ of length not exceeding that of $\mathcal{D}'$. By the induction hypothesis, $\Delta \vdash \chi_1 \to \chi_2[b/a]$. Since b does not occur in Δ or χ_1 and x does not occur in $\chi_2[b/a]$, rule **D4** in context Δ yields $\Delta \vdash \chi_1 \to \forall x\, (\chi_2[b/a])[x/b]$. Because b does not occur in χ_2, we have $(\chi_2[b/a])[x/b] \equiv \chi_2[x/a]$, i.e., $\Delta \vdash \chi_1 \to \forall x\, \chi_2[x/a]$.

(b) The forward direction is proved by induction on the derivation of $\Gamma, \varphi \vdash \psi$:

- if ψ is an axiom or occurs in Γ, then $\Gamma \vdash \psi$, and hence $\Gamma \vdash \varphi \to \psi$ (using $\psi \to (\varphi \to \psi)$ and **D3**);
- if $\psi \equiv \varphi$, the propositional theorem $\varphi \to \varphi$ is used;
- a step **D3** is handled by Frege's axiom;
- if $\psi \equiv \chi_1 \to \forall x\, \chi_2[x/a]$ is obtained by the first rule **D4** from $\Gamma, \varphi \vdash \chi_1 \to \chi_2$ under the conditions that a occurs in none of Γ, φ, and χ_1, and x does not occur in χ_2, then by the induction hypothesis $\Gamma \vdash \varphi \to (\chi_1 \to \chi_2)$. Using the propositional tautology $(\varphi \to (\chi_1 \to \chi_2)) \to ((\varphi \wedge \chi_1) \to \chi_2)$, we obtain $\Gamma \vdash (\varphi \wedge \chi_1) \to \chi_2$. Since a occurs neither in Γ nor in $\varphi \wedge \chi_1$ and x does not occur in χ_2, rule **D4** in context Γ yields $\Gamma \vdash (\varphi \wedge \chi_1) \to \forall x\, \chi_2[x/a]$, whence the propositional tautology $((\varphi \wedge \chi_1) \to \forall x\, \chi_2[x/a]) \to (\varphi \to (\chi_1 \to \forall x\, \chi_2[x/a]))$ derives $\Gamma \vdash \varphi \to (\chi_1 \to \forall x\, \chi_2[x/a])$;
- if $\psi \equiv \exists x\, \chi_1[x/a] \to \chi_2$ is obtained by the second rule **D4** from $\Gamma, \varphi \vdash \chi_1 \to \chi_2$ under the conditions that a occurs in none of Γ, φ, and χ_2, and x does not occur in χ_1, then by the induction hypothesis $\Gamma \vdash \varphi \to (\chi_1 \to \chi_2)$, whence a propositional tautology gives $\Gamma \vdash \chi_1 \to (\varphi \to \chi_2)$. Since a occurs neither in Γ nor in $\varphi \to \chi_2$ and x does not occur in χ_1, rule **D4** in context Γ yields $\Gamma \vdash \exists x\, \chi_1[x/a] \to (\varphi \to \chi_2)$, and interchanging the antecedents completes the step.

The converse follows from part (a): if $\Gamma \vdash \varphi \to \psi$, then by part (a) $\Gamma, \varphi \vdash \varphi \to \psi$, while by **D2** $\Gamma, \varphi \vdash \varphi$, whence **D3** gives $\Gamma, \varphi \vdash \psi$. □

Proposition 2.13 (Replacement of equivalent formulas and properties of equality)**.**

(a) *If* $\vdash \varphi \leftrightarrow \psi$*, then:*

$$\vdash \neg\varphi \leftrightarrow \neg\psi, \qquad \vdash (\varphi \circ \chi) \leftrightarrow (\psi \circ \chi), \qquad \vdash (\chi \circ \varphi) \leftrightarrow (\chi \circ \psi),$$

and for any free variable a *and any bound variable* x *occurring in neither* φ *nor* ψ*:*

$$\vdash \forall x\, \varphi[x/a] \leftrightarrow \forall x\, \psi[x/a], \qquad \vdash \exists x\, \varphi[x/a] \leftrightarrow \exists x\, \psi[x/a].$$

(b) (Symmetry and transitivity of equality.) *For any terms* $T, R, T_1, T_2, T_3 \in \mathrm{Term}_\Sigma$*, the formulas*

$$(T = R) \to (R = T), \qquad (T_1 = T_2) \to ((T_2 = T_3) \to (T_1 = T_3))$$

are derivable in the calculus.

Proof. (a) For the propositional connectives, the claim follows by direct substitution of FOL formulas into the corresponding propositional theorems of PC ($(p \leftrightarrow q) \to (\neg p \leftrightarrow \neg q)$, $(p \leftrightarrow q) \to ((p \circ r) \leftrightarrow (q \circ r))$, and $(p \leftrightarrow q) \to ((r \circ p) \leftrightarrow (r \circ q))$), which become theorems of FOL by group **Ax1** and rule **T2**. For $\forall$: from $\vdash \varphi \to \psi$ and the quantifier axiom $\forall x\, \varphi[x/a] \to \varphi$ (with $T \equiv a$) one obtains $\vdash \forall x\, \varphi[x/a] \to \psi$, whence by the first rule **T3** (since a does not occur in $\forall x\, \varphi[x/a]$ and x does not occur in ψ) it follows that $\vdash \forall x\, \varphi[x/a] \to \forall x\, \psi[x/a]$; the converse implication is derived symmetrically. For $\exists$, the axiom $\psi \to \exists x\, \psi[x/a]$ and the second rule **T3** are used.

(b) For symmetry, choose a free variable a not occurring in T and apply the congruence schema for formulas to $\varphi \equiv (a = T)$: this gives $(T = R) \to ((T = T) \to (R = T))$, which together with the axiom $T = T$ yields $(T = R) \to (R = T)$. For transitivity, choose a not occurring in T_3 and apply the same schema to $\varphi \equiv (a = T_3)$ at the pair (T_2, T_1): this gives $(T_2 = T_1) \to ((T_2 = T_3) \to (T_1 = T_3))$, which combined with symmetry $(T_1 = T_2) \to (T_2 = T_1)$ yields $(T_1 = T_2) \to ((T_2 = T_3) \to (T_1 = T_3))$. □

2.4 First-Order Theories

Retaining the metavariables Γ, Δ for finite lists of formulas (§ 2.3), we reserve upright uppercase letters A, B for arbitrary (not necessarily finite) sets of formulas of signature Σ. For a set of formulas $\mathrm{A} \subseteq \mathrm{Form}_\Sigma$, one writes $\mathrm{A} \vdash \varphi$ (or $\mathrm{A} \vdash_\Sigma \varphi$ when the signature needs to be indicated explicitly) if there exists a finite list Γ of formulas from A (possibly empty) such that $\Gamma \vdash \varphi$ in the sense of Definition 2.6. Because every derivation is finite, this relation is finitary, and by Proposition 2.12 it is monotonic in A and satisfies the Deduction Theorem: $\mathrm{A} \cup \{\varphi\} \vdash \psi$ if and only if $\mathrm{A} \vdash \varphi \to \psi$.

Definition 2.14 (Deductive closure and consistency)**.** Let Σ be a fixed signature, and let $\mathrm{Sent}_\Sigma \subseteq \mathrm{Form}_\Sigma$ be the set of all sentences (closed formulas) of signature Σ.

(a) The *deductive closure* of a set of formulas $\mathrm{A} \subseteq \mathrm{Form}_\Sigma$ (in particular, of a set of sentences $\mathrm{A} \subseteq \mathrm{Sent}_\Sigma$) in signature Σ is the set of all sentences derivable from A:

$$[\mathrm{A}]_\Sigma \stackrel{\text{def}}{=} \{\varphi \in \mathrm{Sent}_\Sigma \mid \mathrm{A} \vdash \varphi\}.$$

(b) A set of formulas $\mathrm{A} \subseteq \mathrm{Form}_\Sigma$ is called *consistent* if there exists no formula $\varphi \in \mathrm{Form}_\Sigma$ such that simultaneously $\mathrm{A} \vdash \varphi$ and $\mathrm{A} \vdash \neg\varphi$ (when $\mathrm{Sent}_\Sigma \neq \emptyset$, by the propositional tautology $p \to (\neg p \to q)$ this is equivalent to $[\mathrm{A}]_\Sigma \neq \mathrm{Sent}_\Sigma$; moreover, $\mathrm{Sent}_\Sigma \neq \emptyset$ both in FOL with equality, which contains $\forall x\,(x = x)$, and in FOL without equality when $\mathrm{Pred}_\Sigma \neq \emptyset$, which contains $\forall x\,P(x,\ldots,x)$), and *inconsistent* otherwise.

Because $[\mathrm{A}]_\Sigma \subseteq \mathrm{Sent}_\Sigma$ by definition, if $\mathrm{A} \subseteq \mathrm{Form}_\Sigma$ contains formulas with free variables, the inclusion $\mathrm{A} \subseteq [\mathrm{A}]_\Sigma$ fails (only $\mathrm{A} \cap \mathrm{Sent}_\Sigma \subseteq [\mathrm{A}]_\Sigma$ holds). In model theory and axiomatic theories, one generally restricts attention to sets of sentences $\mathrm{A} \subseteq \mathrm{Sent}_\Sigma$: for these, $\mathrm{A} \subseteq [\mathrm{A}]_\Sigma$ and $[\mathrm{A}]_\Sigma = [[\mathrm{A}]_\Sigma]_\Sigma$ always hold (the closure operation is idempotent).

Definition 2.15 (Theory and axiomatization)**.**

(a) A (first-order) *theory* in signature Σ is a consistent, deductively closed set of sentences $\mathcal{T} \subseteq \mathrm{Sent}_\Sigma$ (that is, $[\mathcal{T}]_\Sigma = \mathcal{T}$ and $\mathcal{T} \neq \mathrm{Sent}_\Sigma$). The elements of $\mathcal{T}$ are called *theorems* of the theory $\mathcal{T}$.

(b) For a theory $\mathcal{T}$, any subset $\mathrm{A} \subseteq \mathcal{T}$ satisfying $\mathcal{T} = [\mathrm{A}]_\Sigma$ is called an *axiomatization* (or *axiom system*) of $\mathcal{T}$, and the elements of A are called the *non-logical axioms* of that axiomatization. A theory $\mathcal{T}$ is called *finitely axiomatizable* if it admits a finite axiomatization.

For an arbitrary set of sentences $\mathrm{A} \subseteq \mathrm{Sent}_\Sigma$, its deductive closure $[\mathrm{A}]_\Sigma$ is a theory if and only if A is consistent. Every theory $\mathcal{T}$ is automatically non-empty, because deductive closure entails $[\emptyset]_\Sigma \subseteq \mathcal{T}$: every logical theorem that is a sentence belongs to $\mathcal{T}$. The consistency of $[\emptyset]_\Sigma$ (which thus serves as the smallest theory by inclusion in signature Σ) is established purely finitarily: define on Form_Σ a valuation $v(\varphi) \in \{0,1\}$ by recursion on the number of connectives and quantifiers (justified by Proposition 1.8 and Lemma 1.9), erasing terms and quantifiers: set $v(\varphi) = 1$ for every atomic formula, compute the connectives $\neg, \wedge, \vee, \to$ by the truth tables on $\{0,1\}$, and for quantified formulas set $v(\mathsf{Q}x\,\chi) = v(\chi[b/x])$, where $b \in \mathrm{FVar}$ is the shortest free variable not occurring in χ. By simultaneous induction on the number of connectives and quantifiers in φ, two properties are verified: (1) for any quantified formula $\mathsf{Q}x\,\chi$, the value $v(\chi[c/x])$ does not depend on the choice of $c \in \mathrm{FVar}$ not occurring in χ (since $\chi[c/x] \equiv (\chi[b/x])[c/b]$, the second property applies to the formula $\chi[b/x]$ of smaller complexity), and (2) $v(\varphi[T/a]) = v(\varphi)$ for all $T \in \mathrm{Term}_\Sigma$ and $a \in \mathrm{FVar}$ (for an atomic formula $\varphi[T/a]$ remains atomic, connectives are preserved by the induction hypothesis, and for $\mathsf{Q}y\,\chi$ choosing a fresh $c \notin \chi, T, a$ gives $(\chi[T/a])[c/y] \equiv (\chi[c/y])[T/a]$, whence by the induction hypothesis and property (1), $v((\mathsf{Q}y\,\chi)[T/a]) = v((\chi[c/y])[T/a]) = v(\chi[c/y]) = v(\mathsf{Q}y\,\chi)$). In particular, for the quantifier axioms **Ax2**, $v(\forall x\,\psi[x/a]) = v(\psi) = v(\psi[T/a])$ (and similarly for $\exists$), so every axiom of groups **Ax1**–**Ax3** receives the value 1, and rules **T2**–**T3** preserve the value 1. Consequently, $v(\theta) = 1$ for all θ with $\mathrm{Thm}(\theta)$, whereas $v(\neg\theta) = 0$ (in particular, for any formula $\varphi \in \mathrm{Form}_\Sigma$, $v(\varphi \to \varphi) = 1$ and $v(\neg(\varphi \to \varphi)) = 0$, so $\neg(\varphi \to \varphi)$ is not derivable).

Separating the non-logical axioms from the theory itself as a generating set highlights that one and the same theory admits different axiomatizations; independence of axioms (the requirement that no axiom $\alpha \in \mathrm{A}$ be derivable from $\mathrm{A}\setminus\{\alpha\}$) is not required in the definition of an axiomatization, which simplifies comparing different axiom systems and theories (in particular, $\mathcal{T}$ itself serves as its own axiomatization). If $\mathrm{A} \subseteq \mathrm{Sent}_\Sigma$ and a formula $\varphi \in \mathrm{Form}_\Sigma$ contains free variables $a_1, \ldots, a_k$, then by the generalization rule (Proposition 2.4)

and the $\forall$-elimination axiom, derivability $\mathrm{A} \vdash \varphi$ is equivalent to derivability of its universal closure $\mathrm{A} \vdash \forall x_1 \ldots \forall x_k \, \varphi[x_1/a_1] \ldots [x_k/a_k]$. For this reason, an open formula in non-logical axioms is frequently written as standard shorthand for its universal closure.

Definition 2.16 (Deductively complete theory)**.** A theory $\mathcal{T}$ in signature Σ is called *deductively complete* (or simply *complete*) if for every sentence $\varphi \in \mathrm{Sent}_\Sigma$, either $\varphi \in \mathcal{T}$ or $\neg\varphi \in \mathcal{T}$.

Example 2.17 (Examples of first-order theories)**.**

(1) *Pure equality and cardinality axioms.* In the empty signature $\Sigma = \emptyset$, the sole atomic predicate is equality $=$. For each $n \geq 2$, the sentence

$$\alpha_{\geq n} \overset{\text{def}}{\rightleftharpoons} \exists x_1 \ldots \exists x_n \bigwedge_{1 \leq i < j \leq n} \neg(x_i = x_j)$$

expresses the existence of at least n distinct elements, while $\beta_{=n} \overset{\text{def}}{\rightleftharpoons} \alpha_{\geq n} \wedge \neg\alpha_{\geq n+1}$ (for $n = 1$: $\beta_{=1} \overset{\text{def}}{\rightleftharpoons} \neg\alpha_{\geq 2}$) expresses the existence of exactly n elements. Each singleton set $\{\beta_{=n}\}$ ($n \geq 1$), as well as the set $\{\alpha_{\geq n} \mid n \geq 2\}$, is consistent, and the theories they generate—$[\{\beta_{=n}\}]_\emptyset$ and $\mathrm{Th}_\infty \overset{\text{def}}{\rightleftharpoons} [\{\alpha_{\geq n} \mid n \geq 2\}]_\emptyset$ (the theory of infinite sets)—are deductively complete (by quantifier elimination, every sentence in $\Sigma = \emptyset$ reduces to a Boolean combination of the sentences $\alpha_{\geq k}$; see, e.g., Chang and Keisler, 1990; Shoenfield, 1967), whereas pure logic of equality $[\emptyset]_\emptyset$ is incomplete (neither $\alpha_{\geq 2}$ nor $\neg\alpha_{\geq 2}$ is derivable in it). Moreover, Th_∞ is not finitely axiomatizable (by the Compactness Theorem, obtained below in Section 3, Remark 3.6).

(2) *Dense linear order without endpoints* (DLO). The signature $\Sigma = \{<\}$ consists of a single binary predicate symbol (written infix). The axioms are six sentences: irreflexivity $\forall x \, \neg(x < x)$, transitivity $\forall x \forall y \forall z \, ((x < y \wedge y < z) \to x < z)$, linearity $\forall x \forall y \, (x < y \vee x = y \vee y < x)$, density $\forall x \forall y \, (x < y \to \exists z \, (x < z \wedge z < y))$, and the absence of least and greatest elements $\forall x \, \exists y \, (y < x)$, $\forall x \, \exists y \, (x < y)$. This axiom system is consistent, and the theory DLO it generates is finitely axiomatizable and deductively complete (Chang and Keisler, 1990).

(3) *Torsion-free abelian groups.* In the signature $\Sigma = \{+, 0\}$ (a binary function symbol and a constant), to the abelian group axioms

$$\forall x \forall y \forall z \, ((x + y) + z = x + (y + z)), \qquad \forall x \, (x + 0 = x),$$
$$\forall x \, \exists y \, (x + y = 0), \qquad \forall x \forall y \, (x + y = y + x)$$

is added the infinite family of torsion-free axioms $\forall x \, (n \cdot x = 0 \to x = 0)$ for all $n \geq 2$, where $n \cdot T$ denotes the n-fold sum $T + \cdots + T$. This theory is incomplete (the sentence $\exists x \, \neg(x = 0)$ is independent of it); however, adjoining the nontriviality axiom $\exists x \, \neg(x = 0)$ and the divisibility schema $\forall x \, \exists y \, (n \cdot y = x)$ ($n \geq 2$) yields the complete theory of nontrivial divisible torsion-free abelian groups (Chang and Keisler, 1990; Shoenfield, 1967).

(4) *Axiomatic set theory* (ZFC). The signature $\Sigma = \{\in\}$ contains a single binary membership predicate symbol and neither constants nor function symbols. The system of non-logical axioms of ZFC comprises the axioms of extensionality, pairing, union, power

set, infinity, regularity (foundation), the axiom of choice, and the axiom schemas of separation and replacement; assuming consistency ($\mathrm{Con}(\mathsf{ZFC})$), the closure of this axiom system is a theory in the sense of Definition 2.15. The only terms of the language $\mathrm{Term}_{\{\in\}}$ are free variables, so the standard notation $\emptyset$, $\{a, b\}$, $\mathcal{P}(a)$, $a \subseteq b$ does not formally belong to the base signature and is introduced via definitional extensions.

2.5 Definitional Extensions of the Language and Theories

Developing substantive theories in a minimal signature (such as $\{\in\}$ in set theory or $\{0, S, +, \cdot\}$ in arithmetic) requires introducing new symbols—predicates, functions, and constants—as abbreviations for expressions of the original language. While a new predicate symbol is always specified by a formula, introducing a function symbol or constant requires that the defining condition uniquely determine an element in the theory.

To abbreviate notation, the unique existence quantifier is used: for a formula $\varphi \in \mathrm{Form}_\Sigma$, a free variable a, and distinct bound variables $x, y \in \mathrm{BVar}$ not occurring in φ, one sets

$$\exists! x\, \varphi[x/a] \overset{\text{def}}{\rightleftharpoons} \exists x \left(\varphi[x/a] \wedge \forall y \left(\varphi[y/a] \to y = x\right)\right).$$

Definition 2.18 (Definitional extension)**.** Let $\mathcal{T}$ be a theory in signature Σ, and let $\Sigma' = \Sigma \cup \{S\}$ be the extension of the signature by a fresh non-logical symbol S (a constant or template) whose head name does not occur in Σ (so Σ' preserves the antichain property, Definition 1.3). A *definitional extension* of $\mathcal{T}$ by the symbol S is the theory $\mathcal{T}' = [\mathcal{T} \cup \{\alpha_S\}]_{\Sigma'}$ in signature Σ', where the defining axiom $\alpha_S \in \mathrm{Sent}_{\Sigma'}$ has one of the following three forms (depending on the kind of symbol S, with the bound variables $x_1, \ldots, x_n, y \in \mathrm{BVar}$ pairwise distinct and not occurring in the defining formula $\chi \in \mathrm{Form}_\Sigma$):

(1) *Introducing a predicate symbol* $Q(a_1, \ldots, a_n)$: for an arbitrary formula $\chi \in \mathrm{Form}_\Sigma$ in which no free variables other than $a_1, \ldots, a_n$ occur, one sets

$$\alpha_Q \equiv \forall x_1 \ldots \forall x_n \left(Q(x_1, \ldots, x_n) \leftrightarrow \chi[x_1/a_1] \ldots [x_n/a_n]\right).$$

(2) *Introducing a function symbol* $F(a_1, \ldots, a_n)$ $(n \geq 1)$: let $\chi \in \mathrm{Form}_\Sigma$ be a formula all of whose free variables are among $a_1, \ldots, a_n, b$, such that the existence and uniqueness condition is derivable in $\mathcal{T}$:

$$\forall x_1 \ldots \forall x_n\, \exists! y\, \chi[x_1/a_1] \ldots [x_n/a_n][y/b] \in \mathcal{T}.$$

Then one sets

$$\alpha_F \equiv \forall x_1 \ldots \forall x_n \forall y \left(F(x_1, \ldots, x_n) = y \leftrightarrow \chi[x_1/a_1] \ldots [x_n/a_n][y/b]\right).$$

Because $\forall x_1 \ldots \forall x_n\, \exists! y\, \chi[x_1/a_1] \ldots [x_n/a_n][y/b] \in \mathcal{T}$, this axiom is deductively equivalent to the sentence $\beta_F \equiv \forall x_1 \ldots \forall x_n\, \chi[x_1/a_1] \ldots [x_n/a_n][F(x_1, \ldots, x_n)/b]$.

(3) *Introducing a constant* C: the special case of (2) for $n = 0$. If all free variables of $\chi \in \mathrm{Form}_\Sigma$ are among b (i.e., either only b or none) and $\exists! y\, \chi[y/b] \in \mathcal{T}$, then

$$\alpha_C \equiv \forall y \left(C = y \leftrightarrow \chi[y/b]\right),$$

which over $\mathcal{T}$ is equivalent to the sentence $\beta_C \equiv \chi[C/b]$.

An *iterated definitional extension* is the union of a finite or countable ascending chain of theories $\mathcal{T}_0 \subseteq \mathcal{T}_1 \subseteq \mathcal{T}_2 \subseteq \dots$ in which each step $\mathcal{T}_{k+1}$ is a definitional extension of $\mathcal{T}_k$.

Remark 2.19 (Explicit term abbreviations and propagation of congruence)**.** A frequent source of new function symbols and constants is an *explicit term abbreviation*: let $T \in \mathrm{Term}_\Sigma$ contain only free variables from $a_1, \dots, a_n$ ($n \geq 0$), and suppose a symbol $F(a_1, \dots, a_n)$ (or a constant C for $n = 0$) is introduced as an abbreviation for T. This case falls under clause (2) of Definition 2.18 with $\chi \equiv (b = T)$ (where b does not occur in T): the condition $\forall x_1 \dots \forall x_n \, \exists! y \, (y = T[x_1/a_1] \dots [x_n/a_n])$ is derivable purely logically (for the empty theory $[\emptyset]_\Sigma$) from reflexivity, symmetry, and transitivity of equality, and the axiom α_F is equivalent to

$$\beta_F \equiv \forall x_1 \dots \forall x_n \, \big(F(x_1, \dots, x_n) = T[x_1/a_1] \dots [x_n/a_n]\big).$$

Adding the new symbol S to the signature Σ' immediately expands the classes $\mathrm{Term}_{\Sigma'}$ and $\mathrm{Form}_{\Sigma'}$: in particular, F may appear inside compound terms at arbitrary depth and in the arguments of arbitrary predicates. For such an abbreviation to genuinely permit substituting F for T (and vice versa) in any context, congruence of equality must extend to the new symbol. In the formulation of the predicate calculus adopted in Definition 2.1, the congruence schemas **Ax3** apply to all terms and formulas of the current signature, so upon passing from Σ to Σ' the axioms

$$(R_1 = R_1' \wedge \dots \wedge R_n = R_n') \to \big(F(R_1, \dots, R_n) = F(R_1', \dots, R_n')\big)$$

immediately belong to the logical axioms of Σ' (in the atomic formulation of Remark 2.2, the same implication follows from congruence for T, and in the general case of Definition 2.18 (2) from congruence for χ and uniqueness). Together with schema **Ax3** for formulas $\varphi \in \mathrm{Form}_{\Sigma'}$, the defining axiom α_F entails

$$\mathcal{T}' \vdash_{\Sigma'} \varphi[F(R_1, \dots, R_n)/c] \leftrightarrow \varphi[T[R_1/a_1] \dots [R_n/a_n]/c]$$

for any terms $R_1, \dots, R_n \in \mathrm{Term}_{\Sigma'}$ and free variable c, so the new notation and the term it abbreviates are interchangeable in every formula of the extended language.

The next theorem shows that a definitional extension not only allows every formula of the extended language to be translated back into the base language Σ, but also generates no new theorems in the old language.

Theorem 2.20 (Eliminability and conservativity of definitional extensions)**.** *Let $\mathcal{T}' = [\mathcal{T} \cup \{\alpha_S\}]_{\Sigma'}$ be a definitional extension of a theory $\mathcal{T}$ in signature Σ (Definition 2.18), and let $\mathrm{FV}(\psi)$ denote the set of all free variables occurring in a formula ψ. There exists a constructive map $\varphi \mapsto \varphi^*$ from $\mathrm{Form}_{\Sigma'}$ to Form_Σ that introduces no new free variables ($\mathrm{FV}(\varphi^*) \subseteq \mathrm{FV}(\varphi)$) and satisfies the following properties:*

(a) $\varphi^* \equiv \varphi$ *for every formula* $\varphi \in \mathrm{Form}_\Sigma$*;*

(b) $\mathcal{T} \cup \{\alpha_S\} \vdash_{\Sigma'} \varphi \leftrightarrow \varphi^*$ *for every formula* $\varphi \in \mathrm{Form}_{\Sigma'}$*;*

(c) (syntactic conservativity) *for every sentence* $\varphi \in \mathrm{Sent}_\Sigma$*,*

$$\varphi \in \mathcal{T}' \iff \varphi \in \mathcal{T}.$$

Proof. First specify a canonical translation $\varphi \mapsto \varphi^* \in \mathrm{Form}_\Sigma$. Given a formula $\varphi \in \mathrm{Form}_{\Sigma'}$, rename the bound variables in the defining formula χ (inside α_S) to the shortest variables in BVar not occurring in φ: by Lemma 2.10, this leaves χ unchanged up to provable equivalence. Next, ψ^* for subformulas ψ of φ is defined by recursion on the construction of ψ (with the renamed formula χ fixed, and in the function case choosing the shortest fresh variable $y \in \mathrm{BVar}$ not occurring in φ, χ, or the already translated portion):

- If S is a predicate symbol Q, then on an atomic formula we set $Q(R_1, \dots, R_n)^* \equiv \chi[R_1/a_1] \dots [R_n/a_n]$ (here $R_i \in \mathrm{Term}_\Sigma$), and leave all other atomic formulas unchanged.

- If S is a function symbol F (or a constant C, treated as the case $n = 0$), then every atomic formula $\psi \in \mathrm{Form}_{\Sigma'}$ containing F has at least one innermost occurrence $F(R_1, \dots, R_n)$ (where $R_1, \dots, R_n \in \mathrm{Term}_\Sigma$ do not contain F). Choosing the first such occurrence, write $\psi \equiv \psi_0[F(R_1, \dots, R_n)/c]$ for the shortest free variable c occurring in neither ψ nor χ and the shortest bound variable $y \in \mathrm{BVar}$ occurring in none of φ, χ, and ψ_0^*, and replace ψ by the formula

  $$\exists y \, \big(\chi[R_1/a_1] \dots [R_n/a_n][y/b] \wedge \psi_0^*[y/c]\big),$$

  in which ψ_0 contains one fewer occurrence of F (so the recursion on the number of occurrences of F in an atomic formula terminates; when F does not occur in an atom ψ, set $\psi^* \equiv \psi$). In the special case of an explicit term abbreviation $F(a_1, \dots, a_n) = T$ (Remark 2.19), this step amounts to direct word-for-word replacement of $F(R_1, \dots, R_n)$ by $T[R_1/a_1] \dots [R_n/a_n]$ inside the terms of the atomic formula.

- To connectives and quantifiers, the translation extends canonically (independently of the choice of representation of the quantified formula):

  $$(\neg\psi)^* \equiv \neg(\psi^*), \qquad (\psi_1 \circ \psi_2)^* \equiv (\psi_1^* \circ \psi_2^*), \qquad (\mathsf{Q}x\, \psi)^* \equiv \mathsf{Q}x \, \big((\psi[b/x])^*\big)[x/b],$$

  where $b \in \mathrm{FVar}$ is the shortest free variable not occurring in ψ (by Proposition 1.8 (b) and Lemma 1.9, $\psi[b/x] \in \mathrm{Form}_\Sigma$ has strictly smaller complexity; since all newly introduced bound variables from χ and y are chosen outside φ, and hence distinct from x, the variable x does not occur in $(\psi[b/x])^*$).

Because the choice of fresh bound variables in χ and y depends on the entire formula φ being translated, for subformulas ψ_1, ψ_2 of a single formula φ, their standalone translations may differ from their translations inside φ only by renaming of fresh bound variables. Thus, by induction and Definition 2.9, the translation respects connectives, quantifiers, and replacement of free variables up to α-equivalence ($\approx$): $(\neg\psi)^* \approx \neg(\psi^*)$, $(\psi_1 \circ \psi_2)^* \approx (\psi_1^* \circ \psi_2^*)$, $(\mathsf{Q}x\, \psi_0[x/a])^* \approx \mathsf{Q}x \, (\psi_0^*)[x/a]$, and $(\psi[b/a])^* \approx (\psi^*)[b/a]$ (in the quantifier identity, when $b \notin \psi_0[x/a]$, we have $b \notin \psi_0^*$ since $\mathrm{FV}(\psi_0^*) \subseteq \mathrm{FV}(\psi_0)$, whence $((\psi_0^*)[b/a])[x/b] \equiv (\psi_0^*)[x/a]$). Up to $\approx$, the translation does not depend on the choice of fresh variables.

Claim (a) is immediate from the construction: formulas in Form_Σ do not contain S. The inclusion $\mathrm{FV}(\varphi^*) \subseteq \mathrm{FV}(\varphi)$ follows because χ contains only free variables from $a_1, \dots, a_n$ (together with b, which is bound by $\exists y$), and the substituted terms R_i contain only variables from φ. Equality $\mathrm{FV}(\varphi^*) = \mathrm{FV}(\varphi)$ does not hold in general: for instance, if $\chi \equiv \forall z \, (z = z)$, then $Q(a)^* \equiv \forall z \, (z = z)$ no longer contains a.

Claim (b) is checked by induction on the construction of φ (in the function case, with a nested induction on the number of occurrences of F in an atomic formula). For an atomic formula $Q(R_1, \ldots, R_n)$, the equivalence $Q(R_1, \ldots, R_n) \leftrightarrow \chi[R_1/a_1] \ldots [R_n/a_n]$ is obtained by substituting the terms $R_1, \ldots, R_n$ into the axiom α_Q (via the $\forall$-elimination schema). For an atomic formula $\psi \equiv \psi_0[F(R_1, \ldots, R_n)/c]$, the axiom α_F yields the equivalence

$$\mathcal{T} \cup \{\alpha_F\} \vdash_{\Sigma'} \forall y \left(F(R_1, \ldots, R_n) = y \leftrightarrow \chi[R_1/a_1] \ldots [R_n/a_n][y/b]\right).$$

From reflexivity $F(R_1, \ldots, R_n) = F(R_1, \ldots, R_n)$ and this equivalence, one derives $\psi \to \exists y \, (\chi[R_1/a_1] \ldots [R_n/a_n][y/b] \wedge \psi_0[y/c])$; conversely, $\chi[\bar{R}/\bar{a}][y/b] \wedge \psi_0[y/c]$ implies $F(\bar{R}) = y$, whence the congruence schema **Ax3** for formulas derives $\psi_0[F(\bar{R})/c] \equiv \psi$. Replacing ψ_0 by ψ_0^* is justified by the nested induction hypothesis, Proposition 2.13 (a), and Lemma 2.10; by the same results, the inductive step for connectives and quantifiers preserves equivalence (passing from $\neg(\psi_1^*)$, $\psi_1^* \circ \psi_2^*$, $\mathsf{Q}x \, (\psi_0^*)[x/a]$ to $(\neg\psi_1)^*$, $(\psi_1 \circ \psi_2)^*$, $(\mathsf{Q}x \, \psi_0[x/a])^*$ via Lemma 2.10).

To prove (c), it suffices to show that $\mathcal{T} \cup \{\alpha_S\} \vdash_{\Sigma'} \psi$ implies $\mathcal{T} \vdash_{\Sigma} \psi^*$ for every formula $\psi \in \mathrm{Form}_{\Sigma'}$ (then for $\varphi \in \mathrm{Sent}_{\Sigma}$, property (a) gives $\varphi^* \equiv \varphi$, whence $\mathcal{T} \vdash_{\Sigma} \varphi$, i.e., $\varphi \in \mathcal{T}$). Proceed by induction on the derivation from $\mathcal{T} \cup \{\alpha_S\}$ in Σ':

- *Hypotheses and axiom α_S.* If $\psi \in \mathcal{T}$, then $\psi^* \equiv \psi \in \mathcal{T}$. For $\psi \equiv \alpha_Q$, the formula $\alpha_Q^* \approx \forall x_1 \ldots \forall x_n \, (\chi[\bar{x}/\bar{a}] \leftrightarrow \chi[\bar{x}/\bar{a}])$ is a logical theorem (by Lemma 2.10). For $\psi \equiv \alpha_F$, the translation $(F(\bar{x}) = y)^*$ is equivalent in $\mathcal{T}$ to $\chi[\bar{x}/\bar{a}][y/b]$ (by $\forall \bar{x} \, \exists! z \, \chi[\bar{x}/\bar{a}][z/b] \in \mathcal{T}$: specifically, $\exists z \, (\chi[\bar{x}/\bar{a}][z/b] \wedge z = y) \leftrightarrow \chi[\bar{x}/\bar{a}][y/b]$), so $\mathcal{T} \vdash_{\Sigma} \alpha_F^*$.

- *Rules of inference.* If $\mathcal{T} \vdash_{\Sigma} \psi_1^*$ and $\mathcal{T} \vdash_{\Sigma} (\psi_1 \to \psi_2)^*$, then since $(\psi_1 \to \psi_2)^* \approx (\psi_1^* \to \psi_2^*)$ and by Lemma 2.10, we have $\mathcal{T} \vdash_{\Sigma} \psi_1^* \to \psi_2^*$, whence Modus Ponens yields $\mathcal{T} \vdash_{\Sigma} \psi_2^*$. Bernays' rules (respecting the side condition on free variables) are also preserved: from $\mathrm{FV}(\psi^*) \subseteq \mathrm{FV}(\psi)$ it follows that non-occurrence of the free variable a is preserved by translation, and the relations $(\chi_1 \to \chi_2)^* \approx (\chi_1^* \to \chi_2^*)$ and $(\chi_1 \to \forall x \, \chi_2[x/a])^* \approx \chi_1^* \to \forall x \, (\chi_2^*)[x/a]$ (and similarly for $\exists$; renaming x in χ_1^*, χ_2^* via Lemma 2.10 if needed should it occur among the fresh variables of the translation) allow applying Bernays' rule and returning to the translation of the conclusion via Lemma 2.10.

- *Logical axioms.* For a propositional axiom $\mathcal{A}[\psi_1/p][\psi_2/q][\psi_3/r]$, its translation is α-equivalent to the axiom $\mathcal{A}[\psi_1^*/p][\psi_2^*/q][\psi_3^*/r]$ and is therefore derivable in $\mathcal{T}$ by Lemma 2.10. In the predicate case $S = Q$, we have $\mathrm{Term}_{\Sigma'} = \mathrm{Term}_{\Sigma}$, and induction on the complexity of $\psi \in \mathrm{Form}_{\Sigma'}$ verifies $(\psi[T/a])^* \approx \psi^*[T/a]$ for all $T \in \mathrm{Term}_{\Sigma}$ and $a \in \mathrm{FVar}$, whence translations of axioms **Ax2** and **Ax3** are α-equivalent to axioms of the same schemas in Σ and derivable by Lemma 2.10. In the function case $S = F$ (or constant C for $n = 0$), choose a fresh free variable $d \in \mathrm{FVar}$ occurring in none of T, R, T_0, ψ, χ and distinct from $a_1, \ldots, a_n, b$, together with fresh bound variables $z_1, \ldots, z_n, y \in \mathrm{BVar}$, and assign to each term $T \in \mathrm{Term}_{\Sigma'}$ a formula $\varphi_T(d) \in \mathrm{Form}_{\Sigma}$ expressing the equality $d = T$: for $T \in \mathrm{Term}_{\Sigma}$ set $\varphi_T(d) \equiv (d = T)$, and for a compound term $F(R_1, \ldots, R_n)$ set

  $$\varphi_{F(R_1,\ldots,R_n)}(d) \equiv \exists z_1 \ldots \exists z_n \left(\varphi_{R_1}[z_1/d] \wedge \cdots \wedge \varphi_{R_n}[z_n/d] \wedge \chi[z_1/a_1] \ldots [z_n/a_n][d/b]\right)$$

  (for $n = 0$ and a constant C, set $\varphi_C(d) \equiv \chi[d/b]$). The verification then splits into three steps:

1) By structural induction on $T \in \mathrm{Term}_{\Sigma'}$, from $\forall \bar{x}\, \exists! y\, \chi[\bar{x}/\bar{a}][y/b] \in \mathcal{T}$ and congruence of equality in Σ one derives $\mathcal{T} \vdash_\Sigma \exists! y\, \varphi_T[y/d]$.

2) Because $\mathcal{T} \vdash_\Sigma \exists! y\, \varphi_T[y/d]$, the formula $\exists y\, (\varphi_T[y/d] \wedge \theta[y/a])$ is equivalent in $\mathcal{T}$ to $\forall y\, (\varphi_T[y/d] \to \theta[y/a])$ and hence commutes in $\mathcal{T}$ with all connectives $\neg, \wedge, \vee, \to$ and quantifiers $\forall, \exists$. Combined with the nested induction on the number of occurrences of F for atomic formulas (where uniqueness $\exists! z\, \varphi_R[z/d]$ allows witnesses for different occurrences of a to be identified), induction on the complexity of $\psi \in \mathrm{Form}_{\Sigma'}$ gives the equivalence

$$\mathcal{T} \vdash_\Sigma (\psi[T/a])^* \leftrightarrow \exists y \left(\varphi_T[y/d] \wedge \psi^*[y/a]\right)$$

(for $y \in \mathrm{BVar}$ occurring in neither ψ^* nor φ_T).

3) From (1) and (2) follow the translations of all schemas **Ax2** and **Ax3**: $(T = T)^*$ is equivalent in $\mathcal{T}$ to $\exists y_1 \exists y_2\, (\varphi_T[y_1/d] \wedge \varphi_T[y_2/d] \wedge y_1 = y_2)$, which holds under $\exists! y\, \varphi_T[y/d]$; the translation $(T = R)^*$ is equivalent in $\mathcal{T}$ to $\exists y\, (\varphi_T[y/d] \wedge \varphi_R[y/d])$, which by uniqueness for φ_T and φ_R entails $\forall y\, (\varphi_T[y/d] \leftrightarrow \varphi_R[y/d])$, and together with (2) yields the congruence schema for formulas (and as its special case for $\psi \equiv (a_0 = T_0[R/a])$ with replacement $[T_0[T/a]/a_0]$, the congruence schema for terms); finally, for **Ax2**, from $(\forall x\, \psi[x/a])^* \approx \forall x\, (\psi^*)[x/a]$, the quantifier axiom in Σ, and $\exists y\, \varphi_T[y/d]$ follows $\exists y\, (\varphi_T[y/d] \wedge \psi^*[y/a])$, i.e., $(\psi[T/a])^*$ by (2) (the $\exists$ case is dual).

The converse implication in (c) is immediate from the inclusion $\mathcal{T} \subseteq \mathcal{T}'$. □

By Theorem 2.20 and a straightforward induction on the number of steps, any iterated definitional extension is likewise conservative over the base theory $\mathcal{T}$ (in particular, consistency of $\mathcal{T}$ implies consistency of $\mathcal{T} \cup \{\alpha_S\}$, so $\mathcal{T}'$ is indeed a theory in signature Σ').

Example 2.21 (Definitional extensions in set theory)**.** In the signature $\Sigma = \{\in\}$ of ZFC (Example 2.17 (4)), standard set-theoretic notation is introduced by a chain of definitional extensions:

(1) The inclusion predicate $a \subseteq b$ is defined unconditionally by the formula $\chi_\subseteq \equiv \forall z\, (z \in a \to z \in b)$ with the axiom $\forall x \forall y\, (x \subseteq y \leftrightarrow \forall z\, (z \in x \to z \in y))$.

(2) The empty set constant $\emptyset$ is defined by $\chi_\emptyset \equiv \forall z\, \neg(z \in b)$: existence follows from the axioms of separation and infinity (or empty set existence), and uniqueness follows from the axiom of extensionality.

(3) The binary function symbol for the unordered pair $\{a_1, a_2\}$ is defined by $\chi_{\{\cdot,\cdot\}} \equiv \forall z\, (z \in b \leftrightarrow (z = a_1 \vee z = a_2))$, whose existence and uniqueness are ensured by the pairing axiom and the axiom of extensionality.

(4) The unary function symbol for the power set $\mathcal{P}(a)$ is defined in the already extended language $\{\in, \subseteq\}$ by $\chi_\mathcal{P} \equiv \forall z\, (z \in b \leftrightarrow z \subseteq a)$ (on the basis of the power set axiom and extensionality).

(5) A constant or function as an explicit term abbreviation: the singleton $\{a\}$ is introduced as an abbreviation for $\{a, a\}$, and the Kuratowski ordered pair $\langle a_1, a_2 \rangle$ as an abbreviation for $\{\{a_1\}, \{a_1, a_2\}\}$ following Remark 2.19.

Remark 2.22 (Hexadecimal coding of syntax and the path to Gödel's incompleteness theorems)**.** Fixing in § 1.1 a single 16-symbol object alphabet $\mathbb{A}$ for all first-order signatures yields a direct arithmetization of syntax without appealing to prime factorization. Because external lists of hypotheses and finite derivations are written with commas inside the same alphabet $\mathbb{A}$ (§ 2.3), assigning the hexadecimal digits $0, 1, \ldots,$ F to the symbols of $\mathbb{A}$ encodes every word ξ over $\mathbb{A}$ (including terms, formulas, hypothesis lists, and derivations) bijectively as a natural number $\ulcorner \xi \urcorner$ whose hexadecimal expansion is obtained by prepending a leading 1 to the digit string of ξ (the leading 1 prevents the loss of leading zeros). Under this coding, the basic operations of the special metalanguage of Part I of the PC paper—concatenation, subword occurrence testing, and word-for-word replacement $\xi[\zeta/\eta]$—are expressed by primitive recursive (indeed, Kalmar elementary) operations on base-16 codes; for a decidable signature Σ, the syntactic predicates Term_Σ, Form_Σ, Ax become primitive recursive relations on $\mathbb{N}$, and for a decidable (respectively recursively enumerable) axiomatization A, the relation "d is the code of a derivation from A of the formula with code n" is primitive recursive (respectively Σ_1), so the provability predicate $\mathrm{Prov}_\mathrm{A}(n)$ is expressed by an arithmetic Σ_1-formula.

Both of Gödel's incompleteness theorems follow from this arithmetization for any consistent theory $\mathcal{T} = [\mathrm{A}]_\Sigma$ with a recursively enumerable axiomatization A (after a conservative definitional extension by Theorem 2.20, if necessary):

- if $\mathcal{T}$ interprets induction-free Robinson arithmetic Q, then all computable relations are representable in $\mathcal{T}$ and the Diagonal Lemma applies, whence $\mathcal{T}$ is deductively incomplete (Gödel's First Incompleteness Theorem);
- if in addition the three Hilbert–Bernays–Löb derivability conditions for Prov_A are derivable in $\mathcal{T}$ (for which interpretability of elementary arithmetic $\mathsf{EA} = I\Delta_0 + \exp$, and a fortiori $I\Sigma_1$, suffices), then the sentence $\mathrm{Con}_\mathrm{A} \equiv \neg\mathrm{Prov}_\mathrm{A}(\ulcorner \bot \urcorner)$ (where $\bot \equiv \neg(\theta_0 \to \theta_0)$ for some $\theta_0 \in \mathrm{Sent}_\Sigma$), expressing the consistency of $\mathcal{T}$, does not belong to $\mathcal{T}$ (Gödel's Second Incompleteness Theorem).

The exact strength of the arithmetic and string systems sufficient for the special metalanguage of Part I of the PC paper itself is examined in the Conclusion.

3 Semantics and Models

3.1 Structures and Tarski Truth

In Section 3, the exposition passes completely to set-theoretic semantics: the metatheory freely uses sets, functions, and relations, and derivability $\mathrm{A} \vdash \varphi$ from an arbitrary set of formulas $\mathrm{A} \subseteq \mathrm{Form}_\Sigma$ is understood in the sense of § 2.4. Throughout the remainder, Σ denotes a (finite or countable) signature.

Definition 3.1 (Σ-structure and variable assignment)**.** An Σ-*structure* (or *model of signature* Σ) is a tuple $\mathfrak{M} = \langle M, \{P^\mathfrak{M}\}_{P \in \mathrm{Pred}_\Sigma}, \{F^\mathfrak{M}\}_{F \in \mathrm{Func}_\Sigma}, \{C^\mathfrak{M}\}_{C \in \mathrm{Const}_\Sigma} \rangle$, where:

- $M \neq \emptyset$ is a non-empty set called the *domain* (or *universe*) of $\mathfrak{M}$;
- for each n-ary $P \in \mathrm{Pred}_\Sigma$, an n-ary relation $P^\mathfrak{M} \subseteq M^n$ is specified;
- for each n-ary $F \in \mathrm{Func}_\Sigma$, an n-ary operation $F^\mathfrak{M} \colon M^n \to M$ is specified;

- for each constant $C \in \mathrm{Const}_\Sigma$, a distinguished element $C^{\mathfrak{M}} \in M$ is specified.

A *variable assignment* in $\mathfrak{M}$ is an arbitrary map $s\colon \mathrm{FVar} \to M$. For an assignment s, a free variable a, and an element $m \in M$, $s(a \mapsto m)$ denotes the assignment that agrees with s on all free variables except a and takes the value m at a.

The *value* of a term T in $\mathfrak{M}$ under the assignment s (denoted $T^{\mathfrak{M}}[s] \in M$) is defined by structural recursion:

$$a^{\mathfrak{M}}[s] = s(a), \qquad C^{\mathfrak{M}}[s] = C^{\mathfrak{M}}, \qquad F(T_1, \dots, T_n)^{\mathfrak{M}}[s] = F^{\mathfrak{M}}(T_1^{\mathfrak{M}}[s], \dots, T_n^{\mathfrak{M}}[s]).$$

Definition 3.2 (Truth of a formula in a structure)**.** The relation of *Tarski truth* (or *satisfaction*) $\mathfrak{M} \vDash \varphi[s]$ of a formula $\varphi \in \mathrm{Form}_\Sigma$ in an Σ-structure $\mathfrak{M}$ under a variable assignment $s\colon \mathrm{FVar} \to M$ is defined by recursion on the number of connectives and quantifiers in φ:

- $\mathfrak{M} \vDash P(T_1, \dots, T_n)[s] \iff (T_1^{\mathfrak{M}}[s], \dots, T_n^{\mathfrak{M}}[s]) \in P^{\mathfrak{M}}$;
- $\mathfrak{M} \vDash (T = R)[s] \iff T^{\mathfrak{M}}[s] = R^{\mathfrak{M}}[s]$;
- $\mathfrak{M} \vDash (\neg\varphi)[s] \iff$ it is not the case that $\mathfrak{M} \vDash \varphi[s]$;
- $\mathfrak{M} \vDash (\varphi \wedge \psi)[s] \iff \mathfrak{M} \vDash \varphi[s]$ and $\mathfrak{M} \vDash \psi[s]$;
- $\mathfrak{M} \vDash (\varphi \vee \psi)[s] \iff \mathfrak{M} \vDash \varphi[s]$ or $\mathfrak{M} \vDash \psi[s]$;
- $\mathfrak{M} \vDash (\varphi \to \psi)[s] \iff$ if $\mathfrak{M} \vDash \varphi[s]$, then $\mathfrak{M} \vDash \psi[s]$;
- for quantified formulas $\forall x\, \psi$ and $\exists x\, \psi$, let b be the shortest (in number of symbols a) free variable not occurring in ψ (then $\psi[b/x]$ is a formula of smaller complexity by Proposition 1.8 and Lemma 1.9):

$$\mathfrak{M} \vDash (\forall x\, \psi)[s] \iff \text{for every } m \in M,\ \mathfrak{M} \vDash \psi[b/x][s(b \mapsto m)],$$
$$\mathfrak{M} \vDash (\exists x\, \psi)[s] \iff \text{there exists } m \in M \text{ such that } \mathfrak{M} \vDash \psi[b/x][s(b \mapsto m)].$$

A formula φ is *true in the structure* $\mathfrak{M}$ (written $\mathfrak{M} \vDash \varphi$) if $\mathfrak{M} \vDash \varphi[s]$ for every assignment $s\colon \mathrm{FVar} \to M$; a formula is *valid* (written $\vDash \varphi$) if $\mathfrak{M} \vDash \varphi$ in every Σ-structure $\mathfrak{M}$. For a set of formulas $\mathrm{A} \subseteq \mathrm{Form}_\Sigma$ (or a finite list Γ), the notation $\mathfrak{M} \vDash \mathrm{A}[s]$ (respectively $\mathfrak{M} \vDash \Gamma[s]$) means that $\mathfrak{M} \vDash \chi[s]$ for all $\chi \in \mathrm{A}$ (respectively $\chi \trianglelefteq \Gamma$), and $\mathfrak{M} \vDash \mathrm{A}$ means that $\mathfrak{M} \vDash \chi$ for all $\chi \in \mathrm{A}$; in this case $\mathfrak{M}$ is called a *model* of A (in particular, a model of a theory $\mathcal{T}$ when $\mathrm{A} = \mathcal{T}$). A formula φ is *semantically entailed* by A ($\mathrm{A} \vDash \varphi$) if for every Σ-structure $\mathfrak{M}$ and every assignment $s\colon \mathrm{FVar} \to M$, $\mathfrak{M} \vDash \mathrm{A}[s]$ implies $\mathfrak{M} \vDash \varphi[s]$.

Lemma 3.3 (Coincidence Lemma and Substitution Lemma)**.** *Let $\mathfrak{M}$ be an Σ-structure and let $s, s_1, s_2\colon \mathrm{FVar} \to M$ be variable assignments.*

(a) (Coincidence Lemma.) *If $s_1(a) = s_2(a)$ for all free variables a occurring in T, then $T^{\mathfrak{M}}[s_1] = T^{\mathfrak{M}}[s_2]$. If $s_1(a) = s_2(a)$ for all free variables a occurring in φ, then $\mathfrak{M} \vDash \varphi[s_1] \iff \mathfrak{M} \vDash \varphi[s_2]$. In particular, for a sentence $\varphi \in \mathrm{Sent}$, truth $\mathfrak{M} \vDash \varphi[s]$ does not depend on the assignment s.*

(b) (Substitution Lemma.) *For any terms $T_0, T \in \mathrm{Term}_\Sigma$, formula $\varphi \in \mathrm{Form}_\Sigma$, and free variable a, one has $(T_0[T/a])^{\mathfrak{M}}[s] = T_0^{\mathfrak{M}}[s(a \mapsto T^{\mathfrak{M}}[s])]$ and*

$$\mathfrak{M} \vDash \varphi[T/a][s] \iff \mathfrak{M} \vDash \varphi[s(a \mapsto T^{\mathfrak{M}}[s])].$$

In particular, for any quantified formula $\forall x\, \psi$ (respectively $\exists x\, \psi$) and any free variable c not occurring in ψ, truth $\mathfrak{M} \vDash (\forall x\, \psi)[s]$ is equivalent to $\mathfrak{M} \vDash \psi[c/x][s(c \mapsto m)]$ holding for all $m \in M$ (respectively for some $m \in M$).

Proof. (a) The first statement is immediate by structural induction on the term T. The second is proved by induction on the number of connectives and quantifiers in φ: the atomic case follows from the first statement, and propositional connectives preserve equivalence. For $\varphi \equiv \mathsf{Q}x\,\psi$ and the shortest free variable b not occurring in ψ, every free variable occurring in $\psi[b/x]$ either equals b or occurs in $\mathsf{Q}x\,\psi$; hence coincidence of s_1 and s_2 on the free variables occurring in $\mathsf{Q}x\,\psi$ implies coincidence of $s_1(b \mapsto m)$ and $s_2(b \mapsto m)$ on the free variables occurring in $\psi[b/x]$ for each $m \in M$, and the induction hypothesis applies to $\psi[b/x]$.

(b) The identity for terms T_0 is proved by structural induction on T_0. The equivalence for formulas is proved by induction on the number of connectives and quantifiers in φ: the atomic case and Boolean connectives follow from the claim for terms and the induction hypothesis. Let $\varphi \equiv \forall x\,\psi$. Observe first that for any free variable c not occurring in ψ and the shortest free variable b not occurring in ψ, we have $\psi[c/x] \equiv (\psi[b/x])[c/b]$; applying the induction hypothesis to the formula $\psi[b/x]$ of smaller complexity yields

$$\begin{aligned} \mathfrak{M} \vDash \psi[c/x][s(c \mapsto m)] &\iff \mathfrak{M} \vDash \psi[b/x][(s(c \mapsto m))(b \mapsto m)] \\ &\iff \mathfrak{M} \vDash \psi[b/x][s(b \mapsto m)] \end{aligned}$$

(the last equivalence follows from part (a), since when $c \not\equiv b$ the variable c does not occur in $\psi[b/x]$). Thus the truth condition for $\forall x\,\psi$ is independent of the choice of free variable c not occurring in ψ.

Now choose a free variable c occurring in neither ψ nor T and distinct from a. Then c does not occur in $\psi[T/a]$, and word-for-word replacement gives $(\psi[T/a])[c/x] \equiv (\psi[c/x])[T/a]$. For any $m \in M$, part (a) gives $T^{\mathfrak{M}}[s(c \mapsto m)] = T^{\mathfrak{M}}[s]$ (as c does not occur in T), and since $c \not\equiv a$, we have $(s(c \mapsto m))(a \mapsto T^{\mathfrak{M}}[s]) = (s(a \mapsto T^{\mathfrak{M}}[s]))(c \mapsto m)$. Applying the induction hypothesis to the formula $\psi[c/x]$ of smaller complexity and the assignment $s(c \mapsto m)$ for each $m \in M$ yields the required equivalence:

$$\mathfrak{M} \vDash (\forall x\,\psi[T/a])[s] \iff \mathfrak{M} \vDash (\forall x\,\psi)[s(a \mapsto T^{\mathfrak{M}}[s])].$$

The case $\exists x\,\psi$ is analogous. □

3.2 Soundness and Completeness

Two principal metatheoretic results link syntactic derivability $\vdash$ with semantic entailment $\vDash$.

Proposition 3.4 (Soundness of FOL)**.** *Let* $\mathrm{A} \subseteq \mathrm{Form}_\Sigma$ *and* $\varphi \in \mathrm{Form}_\Sigma$. *If* $\mathrm{A} \vdash \varphi$, *then* $\mathrm{A} \vDash \varphi$.

Proof. By definition of $\mathrm{A} \vdash \varphi$, there exists a finite list Γ of formulas from A such that $\Gamma \vdash \varphi$ in the sense of Definition 2.6. Fix an arbitrary Σ-structure $\mathfrak{M}$ with domain M. By induction on the derivation of $\Gamma \vdash \varphi$, we show that for every assignment $s\colon \mathrm{FVar} \to M$ such that $\mathfrak{M} \vDash \Gamma[s]$, one has $\mathfrak{M} \vDash \varphi[s]$.

- *Axiom basis* (**D1**): propositional axioms are true under every assignment s by the definition of truth for $\neg, \wedge, \vee, \to$ via the classical Boolean operations on $\{0,1\}$. Equality axioms are true: reflexivity $T = T$ is obvious, and for the congruence schemas $(T = R) \to (T_0[T/a] = T_0[R/a])$ and $(T = R) \to (\psi[T/a] \to \psi[R/a])$, the equality $T^{\mathfrak{M}}[s] = R^{\mathfrak{M}}[s]$ entails $s(a \mapsto T^{\mathfrak{M}}[s]) = s(a \mapsto R^{\mathfrak{M}}[s])$, whence Lemma 3.3 (b)

yields $(T_0[T/a])^{\mathfrak{M}}[s] = (T_0[R/a])^{\mathfrak{M}}[s]$ and $\mathfrak{M} \vDash \psi[T/a][s] \iff \mathfrak{M} \vDash \psi[R/a][s]$. For the quantifier axiom $\forall x\, \psi[x/a] \to \psi[T/a]$ (where x does not occur in ψ): a does not occur in $\psi[x/a]$ and $(\psi[x/a])[a/x] \equiv \psi$, so by Lemma 3.3 (b) the condition $\mathfrak{M} \vDash (\forall x\, \psi[x/a])[s]$ means $\mathfrak{M} \vDash \psi[s(a \mapsto m)]$ for all $m \in M$, in particular for $m = T^{\mathfrak{M}}[s]$, which by the same lemma is equivalent to $\mathfrak{M} \vDash \psi[T/a][s]$. The axiom $\psi[T/a] \to \exists x\, \psi[x/a]$ is verified dually.

- *Hypothesis basis* (**D2**): if $\varphi \trianglelefteq \Gamma$, then $\mathfrak{M} \vDash \varphi[s]$ by the assumption $\mathfrak{M} \vDash \Gamma[s]$.
- *Modus Ponens* (**D3**): from $\mathfrak{M} \vDash \chi[s]$ and $\mathfrak{M} \vDash (\chi \to \varphi)[s]$, Definition 3.2 immediately yields $\mathfrak{M} \vDash \varphi[s]$.
- *Bernays' rules* (**D4**): suppose $\Gamma \vdash \psi \to \chi$, a occurs in neither Γ nor ψ, and x does not occur in χ. For an arbitrary $m \in M$, the assignment $s(a \mapsto m)$ agrees with s on all free variables occurring in Γ and in ψ. By Lemma 3.3 (a), $\mathfrak{M} \vDash \Gamma[s(a \mapsto m)]$, whence by the induction hypothesis $\mathfrak{M} \vDash (\psi \to \chi)[s(a \mapsto m)]$. If $\mathfrak{M} \vDash \psi[s]$, then again by Lemma 3.3 (a) we have $\mathfrak{M} \vDash \psi[s(a \mapsto m)]$, and hence $\mathfrak{M} \vDash \chi[s(a \mapsto m)]$ for all $m \in M$. Since a does not occur in $\chi[x/a]$ and $(\chi[x/a])[a/x] \equiv \chi$, Lemma 3.3 (b) gives $\mathfrak{M} \vDash (\forall x\, \chi[x/a])[s]$. The case $\exists x\, \psi[x/a] \to \chi$ is checked analogously.

Because for all $\mathfrak{M}$ and s the condition $\mathfrak{M} \vDash \mathrm{A}[s]$ implies $\mathfrak{M} \vDash \Gamma[s]$, it follows that $\mathrm{A} \vDash \varphi$. □

Theorem 3.5 (Gödel's Completeness Theorem)**.** *Let* $\mathrm{A} \subseteq \mathrm{Form}_\Sigma$ *and* $\varphi \in \mathrm{Form}_\Sigma$. *If* $\mathrm{A} \vDash \varphi$*, then* $\mathrm{A} \vdash \varphi$.

Remark 3.6 (Relation to the detailed proof of completeness and main corollaries)**.** A detailed proof of the Completeness Theorem (via Henkin's method) is presented in the companion note "The Henkin Construction".

Combined with soundness (Proposition 3.4), the Completeness Theorem yields two fundamental corollaries about first-order models:

(1) *Compactness Theorem:* a set of formulas $\mathrm{A} \subseteq \mathrm{Form}_\Sigma$ has a model (i.e., $\mathfrak{M} \vDash \mathrm{A}[s]$ for some $\mathfrak{M}$ and s) if and only if every finite subset $\mathrm{A}_0 \subseteq \mathrm{A}$ has a model. Indeed, choose an arbitrary formula $\chi_0 \in \mathrm{Form}_\Sigma$ (e.g., $a = a$ in FOL with equality or $P(a, \dots, a)$ in FOL without equality) and set $\bot \equiv \neg(\chi_0 \to \chi_0)$. If A has no model, then $\mathrm{A} \vDash \bot$, whence by Theorem 3.5 $\mathrm{A} \vdash \bot$, i.e., $\Gamma \vdash \bot$ for some finite list Γ of formulas from A; by Proposition 3.4, the finite set $\mathrm{A}_0 \subseteq \mathrm{A}$ of formulas occurring in Γ satisfies $\mathrm{A}_0 \vDash \bot$, and therefore A_0 has no model.

(2) *Downward Löwenheim–Skolem Theorem:* if a set of formulas $\mathrm{A} \subseteq \mathrm{Form}_\Sigma$ has a model, then it has an at most countable model (since the signature Σ, and hence the set of formulas Form_Σ, is at most countable, a model is constructed on equivalence classes of terms in the Henkin construction; for uncountable signatures, not treated in the present note, the upper bound on the cardinality of the model becomes $|\Sigma| + \aleph_0$).

3.3 The Propositional Calculus as a Fragment of the Predicate Calculus

The propositional calculus (PC) admits two representations within first-order logic. In the first, the connectives of PC remain logical connectives, propositional variables become constants of the domain (elementary propositions), and the passage from the domain to truth

values is effected by a single unary predicate; PC thus appears as a quantifier-free fragment of FOL. In the second, formulas of PC become terms, connectives become function symbols, and derivability and equivalence of formulas become relations among elements of the domain; this formalizes the algebraic semantics of PC (Part III of the PC paper).

Throughout this subsection, $\vdash_{\mathrm{PC}}$ denotes derivability in PC (§ II.3 of the PC paper), and $p_1, p_2, \ldots$ denote pairwise distinct propositional variables (corresponding to the class PVar). Because any finite list of PC formulas contains only finitely many variables, the signatures introduced below may be taken to be finite when dealing with concrete Γ and $\mathcal{A}$.

Encapsulation: PC as quantifier-free FOL. Let the signature Σ_0 consist of an infinite family of constants $C_1, C_2, \cdots \in \mathrm{Const}_{\Sigma_0}$, representing elementary propositions in the domain, and a single unary predicate symbol $P \in \mathrm{Pred}_{\Sigma_0}$ ($\mathrm{Func}_{\Sigma_0} = \emptyset$), acting as a formal truth predicate ("the proposition is true"). To each formula $\mathcal{A}$ of the language of PC is assigned a formula $\mathcal{A}^*$ of the language of FOL in signature Σ_0 by replacing each variable p_i with the atomic formula $P(C_i)$:

$$p_i^* \equiv P(C_i), \qquad (\neg\mathcal{A})^* \equiv \neg\mathcal{A}^*, \qquad (\mathcal{A} \circ \mathcal{B})^* \equiv (\mathcal{A}^* \circ \mathcal{B}^*), \quad \circ \in \{\wedge, \vee, \to\}.$$

By Definition 1.5, the formulas of the form $\mathcal{A}^*$ are precisely the closed quantifier-free formulas of equality-free FOL in signature Σ_0, and the map $\mathcal{A} \mapsto \mathcal{A}^*$ is bijective. For a list $\Gamma = \mathcal{C}_1, \ldots, \mathcal{C}_k$ in PC, one sets $\Gamma^* = \mathcal{C}_1^*, \ldots, \mathcal{C}_k^*$.

No individual variables occur in $\mathcal{A}^*$, so the truth of $\mathcal{A}^*$ in an Σ_0-structure $\mathfrak{M} = \langle M, P^{\mathfrak{M}}, \{C_i^{\mathfrak{M}}\}_{i\geq 1}\rangle$ does not depend on the assignment s (Lemma 3.3 (a)) and is determined solely by the conditions $C_i^{\mathfrak{M}} \in P^{\mathfrak{M}}$ (Definition 3.1). In other words, an Σ_0-structure $\mathfrak{M}$ induces a two-valued valuation $\varepsilon_{\mathfrak{M}}$ in the sense of the PC paper via

$$\varepsilon_{\mathfrak{M}}(p_i) = \begin{cases} 1, & \text{if } C_i^{\mathfrak{M}} \in P^{\mathfrak{M}}, \\ 0, & \text{if } C_i^{\mathfrak{M}} \notin P^{\mathfrak{M}}, \end{cases}$$

and every two-valued valuation ε arises in this way from the structure $\mathfrak{M}$ with domain $M = \{0, 1\}$, predicate $P^{\mathfrak{M}} = \{1\}$, and constants $C_i^{\mathfrak{M}} = \varepsilon(p_i)$; moreover, structural induction on $\mathcal{A}$ verifies

$$\mathfrak{M} \vDash \mathcal{A}^* \iff \varepsilon_{\mathfrak{M}}(\mathcal{A}) = 1.$$

Proposition 3.7 (Encapsulation of PC in FOL)**.** *For any list Γ of PC formulas and any PC formula $\mathcal{A}$,*

$$\Gamma \vdash_{\mathrm{PC}} \mathcal{A} \iff \Gamma^* \vdash \mathcal{A}^*,$$

where the right-hand side denotes derivability in FOL of signature Σ_0 (with or without equality).

Proof. Suppose $\Gamma \vdash_{\mathrm{PC}} \mathcal{A}$. Replacing each formula $\mathcal{B}$ in a PC derivation by $\mathcal{B}^*$ carries axioms of PC to axioms of group 1 of Definition 2.1, hypotheses from Γ to hypotheses from Γ^*, and applications of Modus Ponens to applications of rule **D3**; this yields a derivation $\Gamma^* \vdash \mathcal{A}^*$.

Conversely, suppose $\Gamma^* \vdash \mathcal{A}^*$. By Proposition 3.4, $\Gamma^* \vDash \mathcal{A}^*$. Let ε be a two-valued valuation such that $\varepsilon(\mathcal{C}) = 1$ for all $\mathcal{C} \trianglelefteq \Gamma$, and let $\mathfrak{M}$ be the Σ_0-structure with $M = \{0, 1\}$, $P^{\mathfrak{M}} = \{1\}$, and $C_i^{\mathfrak{M}} = \varepsilon(p_i)$. Then $\mathfrak{M} \vDash \Gamma^*$, whence $\mathfrak{M} \vDash \mathcal{A}^*$, i.e., $\varepsilon(\mathcal{A}) = 1$. Therefore $\Gamma \vDash_{\mathbb{B}} \mathcal{A}$, and by Post's theorem (§ III.4 of the PC paper), $\Gamma \vdash_{\mathrm{PC}} \mathcal{A}$. □

Thus PC coincides with the fragment of closed quantifier-free formulas of equality-free FOL in signature Σ_0, and derivability in FOL on this fragment is conservative: a derivation of $\mathcal{A}^*$ in FOL making arbitrary use of quantifiers and equality proves nothing beyond what is derivable in PC. On the other hand, adding quantifiers in signature Σ_0 even without equality produces statements about the domain such as $\exists x\, P(x)$ or $\forall x\, P(x)$ that do not reduce to PC formulas, while in FOL with equality statements about the cardinality of the domain and distinctness of elementary propositions also appear, such as $\exists x\, \exists y\, \neg(x = y)$ or $\neg(C_1 = C_2)$.

Embedding into the domain: the theory of Boolean algebras. Let the signature Σ_{BA} consist of the function symbols $\sim$ (arity 1), $\sqcap$, $\sqcup$, $\Rightarrow$ (arity 2) and the constants $\mathbf{0}, \mathbf{1}$; it contains no non-logical predicate symbols ($\mathrm{Pred}_{\Sigma_{\mathsf{BA}}} = \emptyset$), and the sole relation of the language is logical equality $=$. The operation symbols are distinct from the connectives $\neg, \wedge, \vee, \to$, because function symbols of a signature cannot coincide with logical symbols of the alphabet (Definition 1.3). Infix metalanguage notation $T \sqcap R$, $T \sqcup R$, $T \Rightarrow R$, $\sim T$ is used for terms, and $T \leqslant R$ abbreviates the atomic formula $(T \sqcap R) = T$. The order could also be introduced as a binary predicate symbol of the signature with the axiom $a \leqslant b \leftrightarrow (a \sqcap b) = a$; the resulting theory differs only in notation.

Let BA denote the theory of Boolean algebras: the finite list of equalities of signature Σ_{BA} expressing the laws of a distributive lattice with least element $\mathbf{0}$, greatest element $\mathbf{1}$, and complement $\sim$, together with the equality $(a \Rightarrow b) = (\sim a \sqcup b)$. The equality axioms are not included in this list: they are logical (group 3 of Definition 2.1). The models of BA are precisely Boolean algebras in the sense of § III.3 of the PC paper (including the one-element algebra).

To each variable p_i is assigned the free variable a_i, and to each formula $\mathcal{A}$ of PC is assigned a term $T_{\mathcal{A}}$ of signature Σ_{BA}:

$$T_{p_i} \equiv a_i, \qquad T_{\neg\mathcal{A}} \equiv \sim T_{\mathcal{A}},$$
$$T_{\mathcal{A}\wedge\mathcal{B}} \equiv T_{\mathcal{A}} \sqcap T_{\mathcal{B}}, \qquad T_{\mathcal{A}\vee\mathcal{B}} \equiv T_{\mathcal{A}} \sqcup T_{\mathcal{B}}, \qquad T_{\mathcal{A}\to\mathcal{B}} \equiv T_{\mathcal{A}} \Rightarrow T_{\mathcal{B}}.$$

If $\mathfrak{B}$ is a Boolean algebra viewed as an Σ_{BA}-structure, then an assignment $s\colon \mathrm{FVar} \to \mathfrak{B}$ of free FOL variables determines a valuation $\varepsilon_s(p_i) = s(a_i)$ of PC variables taking values in $\mathfrak{B}$; every PC valuation arises in this way, and structural induction on $\mathcal{A}$ verifies

$$T^{\mathfrak{B}}_{\mathcal{A}}[s] = \varepsilon_s(\mathcal{A}).$$

Under this representation, PC formulas become elements of the domain, and assertions about them become atomic equality formulas.

Proposition 3.8 (Algebraic representation of PC)**.** *For any list $\Gamma = \mathcal{C}_1, \ldots, \mathcal{C}_k$ of PC formulas and any PC formula $\mathcal{A}$,*

$$\Gamma \vdash_{\mathrm{PC}} \mathcal{A} \iff \mathsf{BA} \vdash (T_{\mathcal{C}_1} \sqcap \cdots \sqcap T_{\mathcal{C}_k}) \leqslant T_{\mathcal{A}},$$

and for empty Γ: $\vdash_{\mathrm{PC}} \mathcal{A} \iff \mathsf{BA} \vdash T_{\mathcal{A}} = \mathbf{1}$. In particular, $\vdash_{\mathrm{PC}} \mathcal{A} \leftrightarrow \mathcal{B}$ if and only if $\mathsf{BA} \vdash T_{\mathcal{A}} = T_{\mathcal{B}}$.

Proof. By the Deduction Theorem (§ II.3 of the PC paper), $\Gamma \vdash_{\mathrm{PC}} \mathcal{A}$ is equivalent to $\vdash_{\mathrm{PC}} (\mathcal{C}_1 \wedge \cdots \wedge \mathcal{C}_k) \to \mathcal{A}$, and by the completeness theorem with respect to the class of Boolean algebras (§ III.4 of the PC paper), this is equivalent to $\varepsilon(\mathcal{C}_1) \wedge \cdots \wedge \varepsilon(\mathcal{C}_k) \to \varepsilon(\mathcal{A}) = 1$ for every Boolean algebra and every valuation ε in it. By the lemma on implication in a

Boolean algebra (§ III.3 of the PC paper) and the identity $T^{\mathfrak{B}}_{\mathcal{B}}[s] = \varepsilon_s(\mathcal{B})$, this means that $(T_{\mathcal{C}_1} \sqcap \cdots \sqcap T_{\mathcal{C}_k}) \leqslant T_{\mathcal{A}}$ is true in every model of BA under every assignment, i.e., $\mathsf{BA} \vDash (T_{\mathcal{C}_1} \sqcap \cdots \sqcap T_{\mathcal{C}_k}) \leqslant T_{\mathcal{A}}$. By Theorem 3.5 and Proposition 3.4, the latter is equivalent to derivability of this formula from BA. The case of empty Γ and the equivalence claim follow in the same way. □

The theory BA formulates within FOL what in Part III of the PC paper was formulated in the set-theoretic metatheory: equivalence of formulas is expressed by equality, derivability by order, and the Lindenbaum–Tarski algebra LT is the free model of BA on the countable generating set $[p_1], [p_2], \ldots$. Adjoining quantifiers and FOL connectives over equalities allows properties of individual models to be expressed that cannot be expressed by PC formulas: the axiom $\neg(\mathbf{0} = \mathbf{1})$ excludes the one-element algebra, and the axioms $\neg(\mathbf{0} = \mathbf{1})$ and $\forall x\, (x = \mathbf{0} \vee x = \mathbf{1})$ together characterize the two-valued model $\mathbb{B}$ up to isomorphism.

Correspondence dictionary. The table below compares the notions of PC and their images under the two representations. The middle column differs from the first only in casting elementary propositions as atoms $P(C_i)$, as expressed by Proposition 3.7; the right column is the encoding of the algebraic semantics of PC in the language of FOL.

PC	**FOL in Σ_0 (encapsulation)**	**Theory BA (embedding)**
variable p_i	constant C_i and atom $P(C_i)$	free variable a_i
connectives $\neg, \wedge, \vee, \to$	logical connectives $\neg, \wedge, \vee, \to$	function symbols $\sim, \sqcap, \sqcup, \Rightarrow$
formula $\mathcal{A}$	closed quantifier-free formula $\mathcal{A}^*$	term $T_{\mathcal{A}}$
model: Boolean algebra B (two-valued $\mathbb{B}$)	Σ_0-structure $\mathfrak{M}$ (standard: $\{0,1\}$ with $P^{\mathfrak{M}} = \{1\}$)	model $\mathfrak{B} \vDash \mathsf{BA}$
valuation ε of p_i	condition $C_i^{\mathfrak{M}} \in P^{\mathfrak{M}}$	assignment s of a_i
value $\varepsilon(\mathcal{A})$	truth value of $\mathcal{A}^*$ in $\mathfrak{M}$	value of term $T^{\mathfrak{B}}_{\mathcal{A}}[s]$
truth: $\varepsilon(\mathcal{A}) = 1$	$\mathfrak{M} \vDash \mathcal{A}^*$	$\mathfrak{B} \vDash (T_{\mathcal{A}} = \mathbf{1})[s]$
theorem: $\vdash_{\mathrm{PC}} \mathcal{A}$	$\vdash \mathcal{A}^*$	$\mathsf{BA} \vdash T_{\mathcal{A}} = \mathbf{1}$
equivalence: $\vdash_{\mathrm{PC}} \mathcal{A} \leftrightarrow \mathcal{B}$	$\vdash \mathcal{A}^* \leftrightarrow \mathcal{B}^*$	$\mathsf{BA} \vdash T_{\mathcal{A}} = T_{\mathcal{B}}$
$\mathcal{A} \vdash_{\mathrm{PC}} \mathcal{B}$	$\mathcal{A}^* \vdash \mathcal{B}^*$	$\mathsf{BA} \vdash T_{\mathcal{A}} \leqslant T_{\mathcal{B}}$
$\mathcal{C}_1, \ldots, \mathcal{C}_k \vdash_{\mathrm{PC}} \mathcal{B}$	$\mathcal{C}_1^*, \ldots, \mathcal{C}_k^* \vdash \mathcal{B}^*$	$\mathsf{BA} \vdash (T_{\mathcal{C}_1} \sqcap \cdots \sqcap T_{\mathcal{C}_k}) \leqslant T_{\mathcal{B}}$

Conclusion. The Postulate of Binarity and the Spiral of Metatheories

In Part I of the PC paper, a special metalanguage of finite words was carved out of natural language, encompassing operations on words (§ I.1), generation of classes of words (§ I.2), and general statements and induction on the construction (§ I.3). For a substantive study of any formal calculus—proving unique readability, closure under substitution, the Deduction Theorem, and finitary consistency—the induction principle remains indispensable. The analysis below does not abandon induction as a working metatheory, but uncovers the internal stratification of Part I of the PC paper itself: within it emerge successively an operational core of generation and verification of concrete strings (§§ I.1–I.2, stage 0), general statements without induction (the beginning of § I.3, stage 1), and the full finitary metalanguage

with structural induction (§§ I.1–I.3 in full, stage 2), followed by set-theoretic semantics (Part III of the PC paper and Section 3 of the present paper, stage 3).

The postulate of binarity. The pre-formal foundation of the special metalanguage of Part I of the PC paper is the *postulate of binarity* in two inseparable manifestations:

(1) *Symbolic binarity:* the capacity to distinguish at least two non-coinciding symbols ($\partial \not\equiv \varrho$, $\mathtt{0} \not\equiv \mathtt{1}$). Over a one-letter alphabet $\{|\}$, concatenation is commutative ($|^m|^n \equiv |^n|^m$) and reduces to addition of lengths, yielding decidable Presburger arithmetic. Introducing a second distinguishable sign qualitatively transforms the structure: non-commutativity of adjoining ($\partial\varrho \not\equiv \varrho\partial$) and positional coding appear, whereas further expanding a finite alphabet from 2 to the 16 symbols of $\mathbb{A}$ reduces to block coding by four-bit nibbles ($16 = 2^4$) and adds no new expressive power.

(2) *Logical binarity:* bivalence of the outcome of elementary word comparison ("identical / distinct" in § I.1, "true / false" in § I.3 of the PC paper) and the rigidly fixed sense of the metalanguage connectives ("not", "and", "or", "if …, then").

Evolution of recursive rules and closure conditions. Recursive definitions of word classes of the form "basis—rules—closure" (§ I.2 of the PC paper) do not presuppose a strong metatheory precisely because the final clause "no other words belong to the class" changes its logical status as the metatheory ascends:

- at the pre-formal *stage 0* (§§ I.1–I.2 of the PC paper), the rules serve solely as templates for step-by-step verification of a finite sequence of exhibited words, and closure acts as a restrictive norm of admission (a word is admitted to the class only upon exhibiting its finite construction); as noted already in § I.2 of the PC paper, such verification relies only on inspecting words as in § I.1 and invokes neither the logic of general statements nor induction from § I.3;

- upon adding quantifiers at *stage 1* (the beginning of § I.3 of the PC paper), closure turns into an explicit Σ_1-definition of membership via existence of a construction code: $\xi \in K \iff \exists d\, \mathrm{Build}_K(d, \xi)$;

- upon adding induction on the construction at *stage 2* (§§ I.1–I.3 of the PC paper in full), closure unfolds as a schema of structural (or level) induction, allowing one to prove general statements $\forall \xi \in K\, S(\xi)$;

- upon adding the axiom of infinity at *stage 3*, closure takes the classical set-theoretic form of the least fixed point $K = \bigcap\{X \mid \mathrm{Base} \subseteq X \wedge \mathrm{Step}(X) \subseteq X\}$.

The spiral of metatheories and retrospective verification. The axiomatization of a stronger theory can already be written down as a finite text at stage 0, without yet possessing its deductive power. A qualitative increment occurs at the next step, when the constructed system is adopted as the new active metatheory and turned downward to verify and prove the consistency of the preceding stage:

Stage 0. *Generation (quantifier-free operational core, §§ I.1–I.2 of the PC paper).* By means of concatenation, word-for-word replacement, and verification of exhibited strings, the syntax and inference rules of PC and FOL (terms, formulas,

axioms, derivations) are specified and the axioms of formal theories (TC, PA, HF, ZFC) are written down, still without appealing to general statements or induction from § I.3 and without proving general metatheorems.

Stage 1. *Quantification without induction (*$\mathsf{TC} \equiv \mathsf{Q}$*).* Adjoining to the operations of § I.1 the universal and existential statements from the first half of § I.3 of the PC paper ("for all words", "there exists a word") prior to introducing the induction principle—formally, adopting FOL quantifiers over the axioms of concatenation (associativity, the editor property $\xi\eta \equiv \zeta\theta \implies \xi \equiv \zeta \vee \exists\omega\,(\xi\omega \equiv \zeta \vee \zeta\omega \equiv \xi)$, and indivisibility of symbols of a finite alphabet)—yields A. Grzegorczyk's *theory of concatenation* TC (Grzegorczyk, 2005), mutually interpretable (via sequence coding following W. Quine, 1946, or the hexadecimal codes of Remark 2.22) with induction-free *Robinson arithmetic* Q. Turning back to stage 0, the level TC/Q provides Σ_1-representability of the classes Term_Σ, Form_Σ, Ax, Thm, the Diagonal Lemma, undecidability of FOL, and Gödel's First Incompleteness Theorem. In the absence of induction, however, neither totality and uniqueness of word-for-word replacement $\xi[\zeta/\eta]$, nor unique readability (Proposition 1.8), nor the Deduction Theorem, nor finitary consistency of $[\emptyset]_\Sigma$ is provable here.

Stage 2. *Induction over the finite (*$\mathsf{TC} + \mathrm{Ind} \equiv \mathsf{PA} \equiv \mathsf{HF}$*; finitary minimum* $I\Sigma_1 \equiv_{\Pi_2} \mathsf{PRA}$ *and* EA*).* Including induction on the construction (§ I.3) closes the full special metalanguage of Part I of the PC paper (§§ I.1–I.3), and extending the induction schema to all first-order formulas yields the full theory of finite objects in three pairwise bi-interpretable forms (Ackermann, 1937; Grzegorczyk, 2005): the theory of concatenation with the induction schema $\mathsf{TC} + \mathrm{Ind}$, Peano arithmetic PA, and the theory of hereditarily finite sets HF (ZF with the negation of the axiom of infinity and $\in$-induction).

The proofs of Sections 1–2 use the lower, finitary fragment of this stage, marked by three exact features: (a) because of rules **F4** and **D4**, induction goes beyond proper subwords to a numerical measure—quantifier-connective depth of formulas ($\psi[c/b]$ in Lemma 1.9, $\chi[b/x]$ in the valuation v of § 2.4 and in the translation $(\mathsf{Q}x\,\psi)^*$ of Theorem 2.20) and derivation length (in Proposition 2.12 (a) after Lemma 2.7); (b) inductive hypotheses carry free parameters (T, a, Δ, χ) with parameter swapping and shifting inside the inductive step (in Lemma 1.7 (b) and under the double substitution $[c/b]$, $[T/a]$ in Lemma 1.9); (c) when classes are literally specified by an unbounded Σ_1 existential quantifier over constructions, this basis coincides with the arithmetic $I\Sigma_1$, which by the Parsons–Mints–Takeuti theorem (Hájek, Pudlák, 1993) is Π_2-conservative over *primitive recursive arithmetic* PRA (the boundary of Hilbert's finitism according to Tait, 1981), and upon writing explicit exponential code bounds narrows to *elementary arithmetic* $\mathsf{EA} = I\Delta_0 + \exp$ (pure $I\Delta_0$ without exp is insufficient because of 2^n truth tables in PC and iterated substitutions).

Turning downward, stage 2 completely verifies the syntax of stage 0 (parsing, substitution, deduction, eliminability of definitional extensions, finitary consistency of $[\emptyset]_\Sigma$) and proves the consistency of the induction-free theories of stage 1 ($\mathrm{Con}(\mathsf{TC})$, $\mathrm{Con}(\mathsf{Q})$ are already derivable in $\mathsf{PRA}/I\Sigma_1$), yet by Gödel's Second Incompleteness Theorem it cannot prove its own consistency

Con(PA).

Stage 3. *Actual infinity (ZFC; thresholds of countable semantics WKL_0 and ACA_0).* Since the theory of finite sets HF is already built at stage 2, passing to the set-theoretic metatheory ZFC reduces to adopting the axiom of infinity Inf (together with the power set and choice axioms over infinite sets). In terms of reverse mathematics (Simpson, 2009), two thresholds of second-order arithmetic appear within Section 3: constructing a Henkin model and the Completeness and Compactness Theorems for countable signatures are equivalent over RCA_0 to the subsystem WKL_0 (Weak Kőnig's Lemma over Σ_1-induction, which is Π^0_2-conservative over PRA by H. Friedman's theorem), whereas defining the full Tarski truth predicate $\mathfrak{M} \vDash \varphi[s]$ over an arbitrary infinite structure and carrying out the general soundness induction correspond to the system ACA_0 (whose first-order part coincides with PA).

At the level of ZFC, the totality of finite objects of stage 2 is gathered into a single actually infinite set ($V_\omega \vDash \mathsf{HF}$, $\omega \vDash \mathsf{PA}$), yielding proofs of Con(PA)/Con(HF), general first-order model theory, and comparison of arbitrary theories, whereas the question of Con(ZFC) passes to the next turn of the spiral (to large cardinal axioms or the hierarchy of `Type` universes of the Lean kernel from § 1.3). Meanwhile, ZFC itself is written as a recursive axiom schema in the 16-symbol syntax of stage 0, closing the loop of the spiral (Fig. 6).

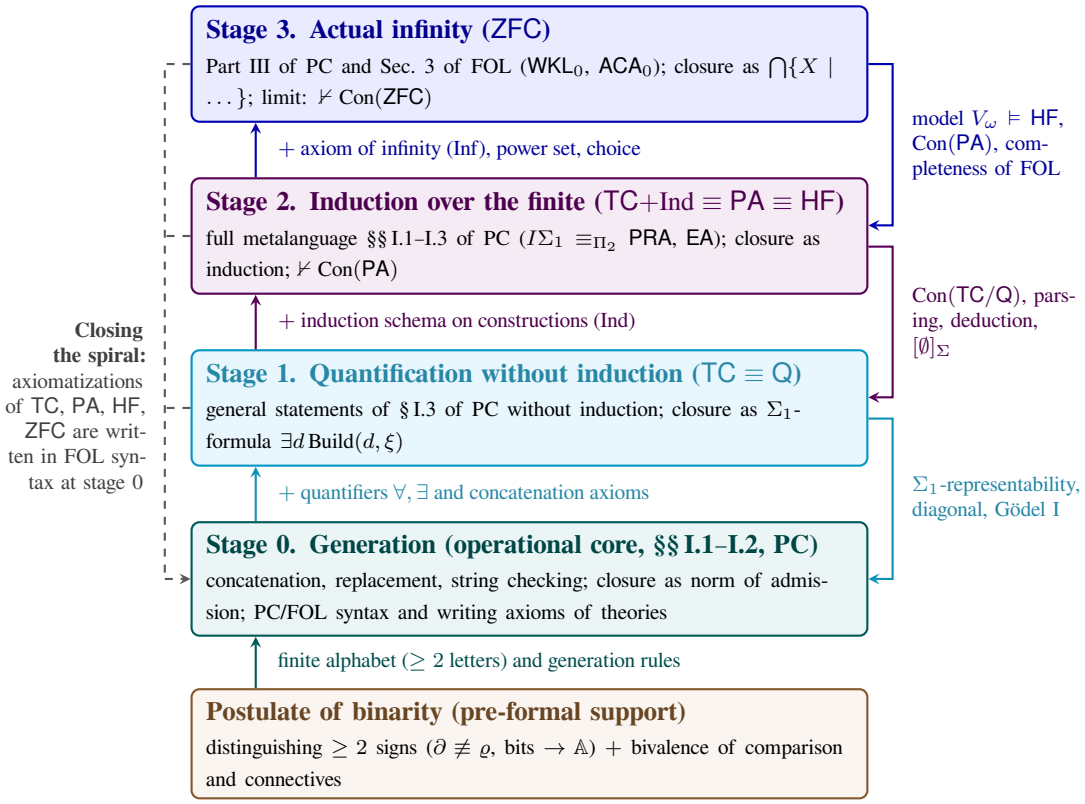


Figure 6: Spiral of metatheoretic ascent: left—return of formal axiomatizations to the syntax of stage 0; center—stepwise strengthening of the metatheory from the postulate of binarity; right—retrospective verification and consistency proof for the preceding stage.

Thus the postulate of binarity serves not as a single axiom from which all of mathematics is deductively derivable—Gödel's theorems rule out the existence of such a closed basis, and each transition to the next stage of the spiral (quantification over a potentially unbounded domain of words, closing finite iteration into an induction principle, reifying an infinite process into an actually infinite set) is an independent act of idealization logically underivable from the preceding one—but rather as the minimal a priori threshold of discursive thought. Precisely herein lies the exact separation between the two levels of the basis: the postulate of binarity together with the operational core of §§ I.1–I.2 of the PC paper provides the minimal *symbolic support* sufficient to write down and execute formal rules, whereas the full special metalanguage of §§ I.1–I.3 with structural induction forms the minimal *deductive metatheory* required to prove the soundness of those rules. Even when constructing many-valued, intuitionistic, or modal calculi, the metalanguage of their syntax and rule verification remains strictly two-valued. Below the threshold of distinction in sign and in judgment, neither formal language, nor proof, nor counting is possible; above it—already without expanding the original two-bit vocabulary—unfolds the entire inexhaustible spiral of mathematical theories and metatheories.

References

[1] Hilbert D., Bernays P. *Grundlagen der Mathematik*, Vol. 1. Berlin: Springer, 1934. (Russian transl.: Moscow, Nauka, 1979.)

[2] Kazimirov N. I. *The Finitary Basis of Formal Systems and a Rigorous Construction of the Propositional Calculus with Its Models*. https://mathem.at/Materials/propositional_calculus_en.html

[3] Kazimirov N. I. *The Henkin Construction*. https://mathem.at/Materials/henkin_construction_en.html

[4] Chang C. C., Keisler H. J. *Model Theory*. 3rd ed. Amsterdam: North-Holland, 1990.

[5] Kleene S. C. *Introduction to Metamathematics*. New York: Van Nostrand, 1952.

[6] Mendelson E. *Introduction to Mathematical Logic*. New York: Van Nostrand, 1964.

[7] Shoenfield J. R. *Mathematical Logic*. Reading, MA: Addison-Wesley, 1967.

[8] Ackermann W. Die Widerspruchsfreiheit der allgemeinen Mengenlehre. *Mathematische Annalen*, 1937, Vol. 114, pp. 305–315.

[9] Grzegorczyk A. Undecidability without arithmetization. *Studia Logica*, 2005, Vol. 79, No. 2, pp. 163–233.

[10] Hájek P., Pudlák P. *Metamathematics of First-Order Arithmetic*. Berlin: Springer, 1993.

[11] Quine W. V. Concatenation as a basis for arithmetic. *Journal of Symbolic Logic*, 1946, Vol. 11, No. 4, pp. 105–114.

[12] Simpson S. G. *Subsystems of Second Order Arithmetic*. 2nd ed. Cambridge: Cambridge University Press, 2009.

[13] Tait W. W. Finitism. *Journal of Philosophy*, 1981, Vol. 78, No. 9, pp. 524–546.