The Finitary Basis of Formal Systems and a Rigorous Construction of the Propositional Calculus with Its Models

Nikolai I. Kazimirov

A note on the foundations of mathematical logic

Abstract

The propositional calculus (PC) is treated as a formal system which replaces ordinary reasoning in the functions required by mathematical proof and which is not a model of ordinary reasoning. In textbook presentations the initial stage of the construction of formal languages exhibits an apparent circle: the formation rules for formulas are written with metavariables whose range (all formulas) is defined by these same rules. Defining a formula through finite tuples of words or as the least fixed point of an operator on $\mathcal{P}(\mathfrak{S}^*)$ removes this appearance at the cost of an appeal to arithmetic and set theory, which are themselves formulated by means of the logic yet to be constructed. The note formulates a finitary starting point in the spirit of D. Hilbert, E. Post, W. Quine and A. A. Markov, sufficient for developing the syntax, substitution, the notion of theorem and the derivability relation of the propositional calculus without the notion of set and without actual infinity. It is pointed out that the notion of finite iteration, and with it contentual arithmetic, is retained at this starting point. The second part considers the functional model and Boolean algebras as models of PC and proves completeness with respect to the class of Boolean algebras and with respect to the two-element algebra. A dictionary translating the calculus into the language of algebra, computations in the Boolean ring by means of Zhegalkin polynomials, the duality principle, the uniqueness of the two-valued model and Post completeness are presented.

Part I. Syntax

1 Formal system and ordinary reasoning

This note constructs the propositional calculus (PC): Part I covers the language, the axioms, the rule of inference and derivability, and Part II covers its models. The construction is preceded by two questions: how this system is related to ordinary logic (the present section) and where the construction should begin in order to avoid a circle (§ 2). Notation is introduced as needed.

In §§ 3–5 PC is constructed as a family of predicates on words over a finite alphabet. The notions of truth, meaning and consequence play no part in the definitions; the sign $\to$ is a graphic sign, and its reading as "if …, then" does not enter the definitions. Hence the claim that PC models the logic of ordinary reasoning does not follow from the construction. It cannot be a theorem: one of the two sides being compared, ordinary reasoning, is not a formal object.

The identification of PC with ordinary logic rests on three circumstances: the reading of the signs $\neg, \wedge, \vee, \to$ as the connectives "not", "and", "or", "if …, then"; the agreement of PC with the practice of mathematical proofs; and the completeness theorem. The first is a convention about pronunciation. Comparison with ordinary speech reveals the following discrepancies. Each of the formulas below is a theorem of PC in the sense of § 4; the letters $\mathcal{A}, \mathcal{B}, \mathcal{C}$ denote arbitrary formulas.

© Nikolai Kazimirov 2026 | mathem.at

1. **Implication without a connection between premise and conclusion.** The formula $(\mathcal{A} \to \mathcal{B}) \vee (\mathcal{B} \to \mathcal{A})$ is a theorem. Under the above reading it asserts that of any two statements one implies the other, whereas the ordinary "if" presupposes a connection between premise and conclusion (the paradoxes of material implication, C. I. Lewis, 1918; the dispute over the truth criterion for conditionals goes back to Philo and Diodorus). The theorem $\mathcal{A} \to \mathcal{A}$, by contrast, agrees with ordinary logic.

2. **Order and cause in conjunction.** The formula $(\mathcal{A} \wedge \mathcal{B}) \to (\mathcal{B} \wedge \mathcal{A})$ is a theorem. The statements "they took the medicine and recovered" and "they recovered and took the medicine" convey different things (P. Strawson, 1952; H. P. Grice assigns this difference to implicatures and excludes it from the truth conditions).

3. **Bivalence.** The formula $\mathcal{A} \vee \neg\mathcal{A}$ is an axiom. Statements with a vague predicate ("a heap of a hundred grains is large"), statements about contingent future events (Aristotle, *De Interpretatione*, ch. 9) and statements with a failed presupposition ("the present king of France is bald"; B. Russell, 1905, P. Strawson, 1950) do not obviously fall under this schema. As applied to mathematical statements the law is disputed by the intuitionists (L. E. J. Brouwer).

4. **Derivation and reasoning.** Derivability is monotone (Proposition 5.4): adding a hypothesis does not cancel a derivable formula. Legal reasoning is not subject to monotonicity: a later statute partially repeals an earlier one, the earlier statute not being held false from the outset, and the premises are ordered by a scale of priorities. Nor does the rule *modus ponens* prescribe accepting the conclusion: someone who has accepted $\mathcal{A}$ and $\mathcal{A} \to \mathcal{B}$ and has discovered that $\mathcal{B}$ is false gives up one of the premises (G. Harman, 1986). Derivability is a relation between formulas; reasoning is a revision of beliefs.

Mathematical practice. In mathematical proofs discrepancies 1–2 play no essential role. Mathematical statements do not depend on time or context, and the basic form of a conditional statement is universal: "for every integer: if it is divisible by 4, then it is even". Such a statement is refuted only by a counterexample in which the premise is true and the conclusion false. The numbers 8 (both parts true), 2 (premise false, conclusion true) and 3 (both parts false) are not counterexamples. If a conditional is treated as a truth function, this requirement determines the table of the implication called *material*: it is false only when the premise is true and the conclusion false. The calculus of G. Frege (1879) was likewise created for writing down proofs in arithmetic; Frege compared the relation of his notation to ordinary language with the relation of the microscope to the eye.

Explication. The relation of PC to ordinary logic is described by the notion of *explication* (R. Carnap, 1950): an inexact concept (the explicandum) is replaced by an exact one (the explicatum), which is assessed by similarity to the explicandum, exactness, fruitfulness and simplicity. Coincidence of usage is not required: the biological concept *Piscis* replaces the everyday "fish" and does not cover the whale. W. Quine (1960, § 53) describes explication as elimination: the functions for the sake of which an unclear expression is used are singled out, a replacement is introduced for these functions, and synonymy is not claimed. The paradigm is the ordered pair in Kuratowski's form: the equality $\langle X, Y\rangle = \{\{X\}, \{X, Y\}\}$ does not express what was meant by a pair and secures the only required property: $\langle X, Y\rangle = \langle U, V\rangle$ if and only if $X = U$ and $Y = V$.

In this sense PC is a replacement for ordinary logic and is not a model of it. Material implication performs those functions of the conditional that are needed in proofs: detachment of the conclusion and the writing of universal statements. It does not perform the other functions of the word "if". The difference between a model and a replacement shows itself in case of a discrepancy: a model is corrected when it diverges from the phenomenon being modelled, whereas when PC diverges from an ordinary judgement, it is the judgement that is corrected. An implication with a false premise is accepted as true contrary to ordinary judgement, and derivability becomes the

criterion of correctness of mathematical reasoning. Thus PC serves mathematics, including computer science; for other fields of knowledge it has no such justification and should not be carried over to them. The choice of replacement is not unique: intuitionistic logic, relevance logics and conditional logics preserve other functions of the ordinary connectives and are assessed by the same criteria.

The completeness theorem. The completeness theorem (§ 9) connects two formal predicates: derivability and truth by the tables. The implication table is itself part of the replacement, so the theorem says nothing about the relation to ordinary logic. The squeezing argument of G. Kreisel (1967, for first-order logic; the scheme is the same) derives from completeness the coincidence of intuitive validity with both formal notions under two conditions: everything derivable is intuitively valid, and everything intuitively valid is true by the tables. The first condition requires accepting formulas of the form $\neg\mathcal{A} \to (\mathcal{A} \to \mathcal{B})$ as valid. It holds for the notion of correctness accepted in mathematical practice and fails for the ordinary use of the word "if". The argument concerns a notion that has already been regimented.

The replacement does not eliminate ordinary logic entirely. The definitions of PC are formulated in a contentual metalanguage which itself uses the words "if", "and", "not" (item **B4** of Convention 2.1). The relation between the object language and the metalanguage is the subject of the next section.

2 The problem of the vicious circle and the choice of a starting point

The construction of formal logic without preliminary conventions runs into the obstacle described in L. Carroll's dialogue *What the Tortoise Said to Achilles* (1895) and in W. Quine's paper *Truth by Convention* (1936). The rules for reading and transforming the symbols of the object language are formulated in some metalanguage. Full formalization of the metalanguage requires a meta-metalanguage, which yields an infinite regress; the justification of basic logic by means of that logic itself is a vicious circle (*petitio principii*).

In textbook presentations this obstacle is bypassed in two ways, each of which raises objections from the standpoint of foundations:

1) **Set-theoretic definition.** A formula is defined as an element of the intersection of all inductive subsets of the free monoid $\mathfrak{S}^*$ over the alphabet $\mathfrak{S}$ (a construction going back to Frege and Dedekind, with a quantifier over $\mathcal{P}(\mathfrak{S}^*)$) or as the last component of a finite assembly tuple $\langle \xi_1, \ldots, \xi_N \rangle \in (\mathfrak{S}^*)^N$. Such a definition presupposes axiomatic set theory (with the axioms of infinity and power set) or the arithmetic of the natural numbers $\mathbb{N}$, which are themselves formulated by means of predicate logic and PC. There is no formal contradiction here, since the system being defined and the metatheory are distinct; the objection concerns the order of justification.

2) **Conflation of definition and use.** The formulation "let $\mathcal{A}, \mathcal{B}$ be arbitrary formulas; then $(\mathcal{A} \to \mathcal{B})$ is a formula" combines the *definition* of the predicate "to be a formula" with the *use* of an already defined predicate. This creates the appearance that the metavariables $\mathcal{A}, \mathcal{B}$ range over a domain which has not yet been given at the moment the rule is written down.

The circle is broken by explicitly fixing a starting point. In accordance with the finitist programme of D. Hilbert and P. Bernays, the canonical calculi of E. Post, the theory of algorithms of A. A. Markov and the exposition of S. C. Kleene, a stock of primitive notions of the metatheory is set out, accepted without formal definition. This stock does not include the notion of set, actual infinity or a formal theory of arithmetic. The choice of a starting point is not unique; the criterion is the simplicity of the initial notions (see Remark 2.1).

Meta-convention 2.1 (Primitive finitary base of the metatheory)**.** The following notions and abilities of the contentual metalanguage (English with technical notation) are accepted without definition.

B1. Signs. Recognition of a finite list of concrete graphic signs, telling them apart, and identifying different occurrences of one sign.

B2. Words and concatenation. A word is a finite linear string of signs written from left to right. Operations: *concatenation*, that is, writing the second word immediately to the right of the first, and checking two words for sign-by-sign coincidence (notation: $\equiv$).

B3. Predicates. The predicates Atom, Form, Ax, Thm on words of the language denote certain properties of words that depend on their composition. The expression $\mathrm{Form}(\xi)$ is read: "the word ξ is a formula".

B4. Logic of the metalanguage. At the metalevel ordinary common human logic is used, in formulations as simple and unambiguous as possible.

B5. Recursive construction of classes of words. First a base is given: certain words are singled out from all words as belonging to the class. Then rules are given for generating new words from words already known to be in the class; the class must be closed under these rules. Finally it is declared that there are no other words in the class. A chain of rule applications that begins with the base and ends with a given word is called a *construction* of that word. For the class of theorems a construction is called a *derivation*.

B6. Metavariables. Letters of the metalanguage that denote words are called metavariables. If a statement is written with a metavariable $\mathcal{A}$ restricted to formulas, it is understood to be preceded by the condition "if $\mathcal{A}$ is a formula, then". The same applies to metavariables restricted to other classes.

***Remark* 2.1** (Potential infinity and arithmetic)**.** The condition "there are no other words in the class" in **B5** is the only place in the base where potential infinity is localized. It does not require the existence of a "set of all words" as a completed object, and it justifies the method of **structural induction**: a property possessed by the words of the base and preserved by the rules belongs to all words of the class. If the premises of each rule are shorter than the conclusion or belong to a previously defined class, the same recursion yields a recognition algorithm: a word is parsed by the rules down to the base. For atoms and formulas this algorithm runs in time polynomial in the length of the word. For theorems the condition fails: the premise $\mathcal{A}$ of Modus Ponens is not contained in the conclusion $\mathcal{B}$, and the decidability of the class of theorems is proved separately (Remark 4.4).

3 Alphabet, words and syntactic predicates

The object alphabet $\mathfrak{S}$ is fixed; it consists of exactly **eight** signs:

$$\mid \quad \# \quad (\quad) \quad \neg \quad \wedge \quad \vee \quad \to$$

A word over $\mathfrak{S}$ is a finite string of its signs; the class of all such words is denoted $\mathfrak{S}^*$ (the Kleene star). The Greek letters ξ, η, ζ are metavariables over words from $\mathfrak{S}^*$. Concatenation is written by juxtaposition: $(\xi \to \eta)$ is the word composed of a parenthesis, the word ξ, the sign $\to$, the word η and a parenthesis. The parentheses attached to predicates belong to the metalanguage and are shown in blue; a compound word in the argument of a predicate is enclosed in single quotes: $\mathrm{Form}('(\xi \to \eta)')$.

The propositional variables are strings composed only of strokes and enclosed on both sides by hash signs: #|#, #||#, #|||#, etc. This notation meets three requirements. The stock of variables must be potentially infinite, so a finite set of letters does not suffice. This stock must be given by the simplest rules over a finite alphabet, so letters with numerical subscripts, which require digits and the notion of a subscript, are not used. Replacing a variable must work without error, so a variable cannot be a subword of another variable: the word ||| is a subword of the word ||||, and under replacement a formula would land inside it, whereas the atom #|||# is not a subword of the atom #||||#. The letters a, b, c occur below only as metavariables over atoms and do not belong to the alphabet.

Strokes are added only within atoms, so two predicates are introduced, Atom and Form; the first enters the definition of the second only as the premise of rule **F1**.

Definition 3.1 (The predicate "to be an atom"). The predicate $\mathrm{Atom}(\xi)$ is defined recursively:

A1. *Base:* $\mathrm{Atom}(\text{'\#|\#'})$.

A2. *Rule:* if $\mathrm{Atom}(\text{'\#}\xi\text{\#'})$, then $\mathrm{Atom}(\text{'\#}\xi\text{|\#'})$.

A3. *Closure:* no other words have the property Atom.

The words #|#, #||#, #|||#, etc. have the property Atom. Instead of such elaborate notations for variables one may mentally picture p_1, p_2, p_3, etc.

Definition 3.2 (The predicate "to be a formula"). The predicate $\mathrm{Form}(\xi)$ is defined recursively:

F1. *Base:* if $\mathrm{Atom}(\xi)$, then $\mathrm{Form}(\xi)$.

F2. *Rule for negation:* if $\mathrm{Form}(\xi)$, then $\mathrm{Form}(\text{'}\neg\xi\text{'})$.

F3. *Rules for binary connectives:* if $\mathrm{Form}(\xi)$ and $\mathrm{Form}(\eta)$, then

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

F4. *Closure:* no other words have the property Form.

The implication in rule **F1** goes in one direction: every atom is a formula, while a formula containing a connective or a parenthesis is not an atom.

The capital calligraphic letters $\mathcal{A}, \mathcal{B}, \mathcal{C}$ denote formulas and the lowercase Latin letters a, b, c denote atoms; these are metavariables restricted in the sense of **B6** (Fig. 1). The notation $\mathcal{A}(a)$ denotes a formula $\mathcal{A}$ with a distinguished propositional variable a; neither the occurrence of a in $\mathcal{A}$ nor the absence of other variables from $\mathcal{A}$ is required.

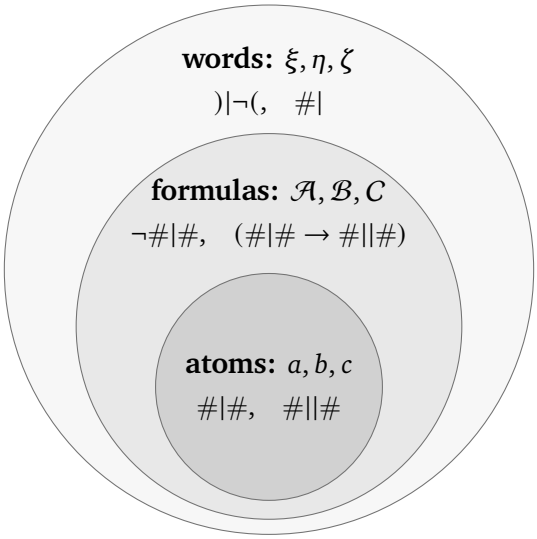


Figure 1: Ranges of the metavariables of the three kinds, with sample values.

Omission of parentheses. Parentheses are omitted only in notation in the metalanguage; the word itself contains all of them. Outer parentheses are dropped: $\mathcal{A} \to \mathcal{B}$ denotes the word $(\mathcal{A} \to \mathcal{B})$. When it occurs in a longer formula they are restored: $\neg(\mathcal{A} \to \mathcal{B})$, $C \to (\mathcal{A} \to \mathcal{B})$. Negation introduces no parentheses, so $\neg\mathcal{A} \to \mathcal{B}$ is read unambiguously as $(\neg\mathcal{A}) \to \mathcal{B}$. The further conventions in common use (the connectives $\wedge, \vee$ bind more tightly than $\to$; a chain of implications associates to the right) are not applied in the present text.

Unfolding the condition. Unfolding the condition from **B6** removes the appearance of a circle. The formulation "if $\mathcal{A}$ and $\mathcal{B}$ are formulas, then $\mathcal{A} \to \mathcal{B}$ is a formula" is written as $\mathrm{Form}('\mathcal{A} \to \mathcal{B}')$ and means

$$\text{for all words } \xi, \eta: \quad \text{if } \mathrm{Form}(\xi) \text{ and } \mathrm{Form}(\eta), \text{ then } \mathrm{Form}('(\xi \to \eta)'),$$

that is, rule **F3** for implication. In this notation ξ, η range over words given independently of the notion of formula, and the predicate Form enters only the premise and the conclusion of the rule. The totality of formulas is not presupposed. The abbreviated notation is legitimate after Definition 3.2; read as the definition itself, it is what creates the appearance of a circle.

Definition 3.3 (Substitution)**.** $\mathcal{A}[\mathcal{B}/a]$ is the word obtained from $\mathcal{A}$ by replacing all occurrences of the atom a with the formula $\mathcal{B}$. For a formula written as $\mathcal{A}(a)$ one also writes $\mathcal{A}(\mathcal{B})$.

Proposition 3.4. *$\mathcal{A}[\mathcal{B}/a]$ is a formula. Substitution commutes with the connectives: $(\neg\mathcal{A})[\mathcal{B}/a] \equiv \neg\mathcal{A}[\mathcal{B}/a]$ and $(\mathcal{A} \circ C)[\mathcal{B}/a] \equiv \mathcal{A}[\mathcal{B}/a] \circ C[\mathcal{B}/a]$, where $\circ$ is one of the signs $\wedge, \vee, \to$.*

Proof. The atoms of a formula are separated by parentheses and connectives, so an occurrence of the atom a in a formula coincides with one of its atoms and crosses neither parentheses nor connectives. Commutativity follows. Structural induction on $\mathcal{A}$: for an atom the result is $\mathcal{B}$ or the atom itself; for a compound formula one applies commutativity, the induction hypothesis and rules **F2**, **F3**. □

Analysis of words. Words and formulas are analysed by means of recursive arithmetic (T. Skolem, 1923): parentheses, the parenthesis balance and the length of a word are counted,

and induction on the length of a word is applied. This use of arithmetic is contentual, in the sense of Remark 2.1.

Lemma 3.5 (Parenthesis balance and initial segments)**.**

(a) *A formula contains equally many opening and closing parentheses; in each of its initial segments there are at least as many opening as closing parentheses; in each nonempty proper initial segment of a formula beginning with a parenthesis there are strictly more opening ones.*

(b) *A proper initial segment of a formula is not a formula.*

Proof. (a) Structural induction.

(b) Structural induction on the formula C; let $\mathcal{A}$ be a formula and a proper initial segment of C. If C is an atom, then $\mathcal{A}$ begins with a hash sign and is therefore an atom, but $\mathcal{A}$ contains fewer than two hash signs. If $C \equiv \neg C_1$, then $\mathcal{A}$ begins with the sign $\neg$; the only rule yielding such formulas is **F2**, so $\mathcal{A} \equiv \neg\mathcal{A}_1$, where the formula $\mathcal{A}_1$ is a proper initial segment of C_1, contradicting the induction hypothesis. If C begins with a parenthesis, then by (a) $\mathcal{A}$ contains strictly more opening than closing parentheses, contradicting (a) for the formula $\mathcal{A}$. □

Proposition 3.6 (Unique parsing)**.** *Every formula $\mathcal{A}$ has exactly one of the following forms: an atom; $\neg\mathcal{A}_1$; $(\mathcal{A}_1 \circ \mathcal{A}_2)$, where $\circ$ is one of the signs $\wedge, \vee, \to$. Moreover, the formulas $\mathcal{A}_1, \mathcal{A}_2$ and the sign $\circ$ are uniquely determined.*

Proof. The form of a formula is determined by its first sign: a hash sign, $\neg$ or an opening parenthesis. In the case of negation the uniqueness of $\mathcal{A}_1$ is obvious. Let $(\mathcal{A}_1 \circ \mathcal{A}_2) \equiv (\mathcal{B}_1 \bullet \mathcal{B}_2)$. If $\mathcal{A}_1 \not\equiv \mathcal{B}_1$, then one of these formulas is a proper initial segment of the other, contradicting Lemma 3.5 (b). Hence $\mathcal{A}_1 \equiv \mathcal{B}_1$, whence $\circ \equiv \bullet$ and $\mathcal{A}_2 \equiv \mathcal{B}_2$. □

4 Axioms and theorems of the calculus

The predicate "to be a theorem of PC" is defined purely syntactically: the axioms and the closure under the rule of inference are fixed. Distinct from it is the notion of *tautology*: a tautology is a formula true by the tables under all assignments of values to its atoms (Definition 7.2). The coincidence of these two predicates is the content of the soundness and completeness theorems (E. Post, 1921; a derivation from a truth table is constructed in the proof of L. Kalmár, 1935), not a definition.

The axioms are given by schemata.

Definition 4.1 (The predicate "to be an axiom")**.** The expressions in the following list contain metavariables and are not words of the language of PC. An axiom is any instance of them, that is, a word obtained by replacing the metavariables $\mathcal{A}, \mathcal{B}, C$ with concrete formulas. No other words are axioms.

- *affirmation of the consequent:* $\mathcal{A} \to (\mathcal{B} \to \mathcal{A})$;
- *Frege's axiom:* $(\mathcal{A} \to (\mathcal{B} \to C)) \to ((\mathcal{A} \to \mathcal{B}) \to (\mathcal{A} \to C))$;
- *conjunction elimination:* $(\mathcal{A} \wedge \mathcal{B}) \to \mathcal{A}$, $\quad (\mathcal{A} \wedge \mathcal{B}) \to \mathcal{B}$;
- *conjunction introduction:* $\mathcal{A} \to (\mathcal{B} \to (\mathcal{A} \wedge \mathcal{B}))$;
- *disjunction introduction:* $\mathcal{A} \to (\mathcal{A} \vee \mathcal{B})$, $\quad \mathcal{B} \to (\mathcal{A} \vee \mathcal{B})$;
- *constructive dilemma:* $(\mathcal{A} \to C) \to ((\mathcal{B} \to C) \to ((\mathcal{A} \vee \mathcal{B}) \to C))$;

- *law of Duns Scotus:* $\neg\mathcal{A} \to (\mathcal{A} \to \mathcal{B})$;
- *negation introduction:* $(\mathcal{A} \to \mathcal{B}) \to ((\mathcal{A} \to \neg\mathcal{B}) \to \neg\mathcal{A})$;
- *law of excluded middle:* $\mathcal{A} \vee \neg\mathcal{A}$.

Each instance is obtained by substituting formulas for atoms in a concrete formula and is a formula by Proposition 3.4.

Definition 4.2 (The predicate "to be a theorem of PC")**.** The predicate Thm is defined recursively:

T1. *Base:* if $\mathrm{Ax}(\mathcal{A})$, then $\mathrm{Thm}(\mathcal{A})$.

T2. *Rule Modus Ponens:* if $\mathrm{Thm}(\mathcal{A})$ and $\mathrm{Thm}('\mathcal{A} \to \mathcal{B}')$, then $\mathrm{Thm}(\mathcal{B})$.

T3. *Closure:* no other words have the property Thm.

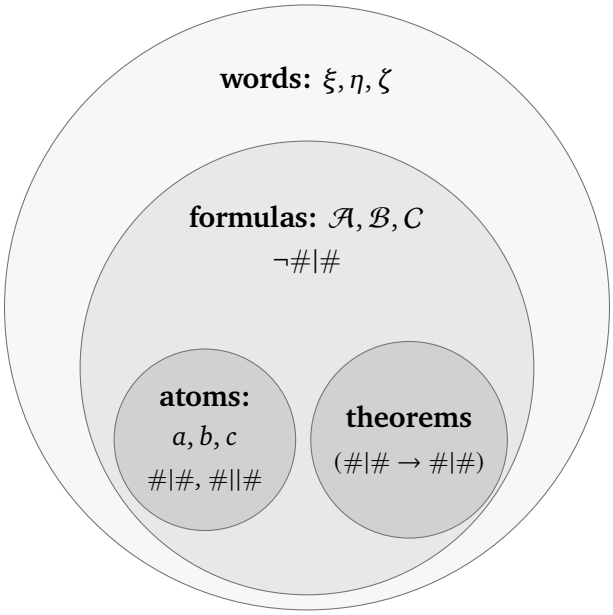


Figure 2: Theorems as a separate class of formulas. Atoms are not theorems (§ 7).

***Remark* 4.3** (A finite list of axioms)**.** The same class of theorems is obtained from 11 concrete formulas, with the atoms $\#|\#, \#||\#, \#|||\#$ in place of $\mathcal{A}, \mathcal{B}, \mathcal{C}$, by adding the rule "if $\mathcal{A}$ is a theorem, then $\mathcal{A}[\mathcal{B}/a]$ is a theorem". Substitution into an axiom of a schema yields an axiom of the same schema; conversely, an arbitrary axiom of a schema is obtained by successive substitutions through intermediate atoms longer than all atoms occurring in $\mathcal{A}, \mathcal{B}, \mathcal{C}$. Without intermediate atoms, substitution for the atom $\#||\#$ would affect its occurrences in the already substituted formula $\mathcal{A}$.

***Remark* 4.4** (Decidability)**.** The classes of atoms, formulas and theorems of PC are decidable: for any word ξ, in a finite number of steps one establishes both the holding and the failure of each of the predicates $\mathrm{Atom}(\xi)$, $\mathrm{Form}(\xi)$, $\mathrm{Thm}(\xi)$. For atoms and formulas the algorithm is parsing according to Proposition 3.6; it is polynomial in the length of the word. For theorems there is also a method of finite proof search based on the functional model of the propositional calculus.

5 Derivability in the context of a list of hypotheses

In a set-theoretic presentation the derivability relation connects an arbitrary, possibly infinite, set of formulas with a single formula. The finitary metatheory has no notion of set, and hypotheses are written as a finite list of formulas. However, forming a list of formulas requires introducing into the alphabet some fresh symbol, a separator of the formulas in the list, so that the list can be parsed unambiguously into formulas. We therefore add the comma to the alphabet as a new symbol and denote the new extended alphabet by $\mathfrak{S}'$.

Since the extended alphabet contains the original one, the predicates Atom, Form, Thm over the new alphabet define exactly the same classes of words, and the meaning of the variables $\mathcal{A}, \mathcal{B}, \ldots$ and $a, b, \ldots$ remains the same. The comma makes it possible to define a new kind of word: a list of formulas. Since the lowercase Greek letters were earlier reserved for words over the original alphabet only, for words over the new alphabet we use capital Greek letters: Γ, Δ. Their range automatically includes the words over the old alphabet as well.

Definition 5.1 (List of formulas and occurrence in a list)**.** The predicate $\mathrm{List}(\Gamma)$ and the statement $\mathcal{A} \trianglelefteq \Gamma$ ("the formula $\mathcal{A}$ occurs in the list Γ") are defined by simultaneous recursion:

L1. *Singleton list:* $\mathrm{List}(\mathcal{B})$ and $\mathcal{B} \trianglelefteq \mathcal{B}$.

L2. *Appending a formula to a list:* if $\mathrm{List}(\Gamma)$, then $\mathrm{List}('\Gamma, \mathcal{B}')$ and $\mathcal{B} \trianglelefteq \Gamma, \mathcal{B}$; if moreover $\mathcal{A} \trianglelefteq \Gamma$, then $\mathcal{A} \trianglelefteq \Gamma, \mathcal{B}$.

L3. *Closure:* no other words are lists, and no other pairs of a formula and a list stand in the relation $\trianglelefteq$.

Formulas contain no comma, so the decomposition of a list into its constituent formulas is unique.

A list of hypotheses is fixed and called a *context*. Instead of a binary relation between lists and formulas, a predicate on formulas is introduced that depends on the context as a parameter. The *empty context*, that is, the absence of hypotheses, is also allowed. The empty context is not a word: it is the case in which the parameter is not specified, so no notation for it is required. From now on the letters Γ, Δ denote contexts. For empty Γ no formula occurs in it, the expression $\Gamma, \mathcal{A}$ denotes the list consisting of the single formula $\mathcal{A}$, and the expression $\Gamma \vdash \mathcal{A}$ takes the form $\vdash \mathcal{A}$.

Definition 5.2 (Derivability in a context)**.** Let Γ be a context. The predicate $\Gamma \vdash \mathcal{A}$ ("the formula $\mathcal{A}$ is derivable in the context Γ") is defined recursively:

D1. *Axiom base:* if $\mathrm{Ax}(\mathcal{A})$, then $\Gamma \vdash \mathcal{A}$.

D2. *Hypothesis base:* if $\mathcal{A} \trianglelefteq \Gamma$, then $\Gamma \vdash \mathcal{A}$.

D3. *Rule Modus Ponens:* if $\Gamma \vdash \mathcal{A}$ and $\Gamma \vdash \mathcal{A} \to \mathcal{B}$, then $\Gamma \vdash \mathcal{B}$.

D4. *Closure:* no other formulas are derivable in the context Γ.

In the empty context rule **D2** does not apply, since no formula occurs in it: derivability $\vdash \mathcal{A}$ is given by rules **D1**, **D3**, **D4**.

Example of a derivation. A derivation in a context is a finite chain of formulas, each of which is an axiom (**D1**), a hypothesis (**D2**) or is obtained from two preceding ones by Modus Ponens (**D3**). For any atoms a, b one has $a, \neg a \vdash b$:

1.	$\neg a$	hypothesis
2.	$\neg a \to (a \to b)$	axiom (law of Duns Scotus)
3.	$a \to b$	Modus Ponens: 1, 2
4.	a	hypothesis
5.	b	Modus Ponens: 4, 3

The rows of the table contain metavariables and are not words of the language of PC, so what is given is not one derivation but a derivation schema: each instantiation of the metavariables a, b by concrete atoms gives a separate derivation, and there are infinitely many such derivations.

Proposition 5.3.

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

(b) *Replacing the premise* $\mathrm{Ax}(\mathcal{A})$ *by* $\mathrm{Thm}(\mathcal{A})$ *in rule* ***D1*** *yields the same predicate.*

Proof. (a) Rules **D1** and **D3** without context coincide with the rules for Thm. (b) By (a) a theorem has a derivation in the empty context, using only **D1** and **D3**; the same derivation works in the context Γ. □

Proposition 5.4 (Monotonicity in the context)**.** *If* $\Gamma \vdash \mathcal{A}$ *and every formula occurring in* Γ *occurs in the context* Δ*, then* $\Delta \vdash \mathcal{A}$*. In particular,* $\Gamma, \mathcal{B} \vdash \mathcal{A}$.

Proof. Induction on the derivation: rules **D1** and **D3** do not depend on the context, and a hypothesis from Γ is a hypothesis from Δ. □

Proposition 5.5 (Deduction theorem; J. Herbrand, 1930)**.** $\Gamma, \mathcal{A} \vdash \mathcal{B}$ *if and only if* $\Gamma \vdash \mathcal{A} \to \mathcal{B}$.

Proof. The converse is Modus Ponens. The direct statement is proved by induction on the derivation $\Gamma, \mathcal{A} \vdash \mathcal{B}$. For $\mathcal{B} \equiv \mathcal{A}$ one uses the theorem $\mathcal{A} \to \mathcal{A}$, derived by two applications of Modus Ponens to instances of Frege's axiom and of affirmation of the consequent. If $\mathcal{B}$ is an axiom or a hypothesis from Γ, affirmation of the consequent is applied. The Modus Ponens step is handled by Frege's axiom. □

$\Gamma[\mathcal{B}/a]$ denotes the context obtained from Γ by substitution in each of its formulas.

Proposition 5.6 (Substitution in a derivation)**.** *If* $\Gamma \vdash \mathcal{A}$*, then* $\Gamma[\mathcal{B}/a] \vdash \mathcal{A}[\mathcal{B}/a]$*. In particular, if the atom* a *does not occur in the formulas of the context* Γ*, then* $\Gamma \vdash \mathcal{A}[\mathcal{B}/a]$*, and if* $\mathrm{Thm}(\mathcal{A})$*, then* $\mathrm{Thm}(\mathcal{A}[\mathcal{B}/a])$.

Proof. Induction on the derivation $\Gamma \vdash \mathcal{A}$. Substitution into an axiom yields an axiom of the same schema. From $\mathcal{A} \trianglelefteq \Gamma$ it follows that $\mathcal{A}[\mathcal{B}/a] \trianglelefteq \Gamma[\mathcal{B}/a]$. For Modus Ponens one applies the commutativity of substitution with implication (Proposition 3.4). □

Proposition 5.6 imposes no restrictions on the variables of the formula $\mathcal{B}$: substitution is performed in a single pass, and substituted occurrences are not scanned again. A condition arises only with several successive substitutions when the result has to coincide with a given formula: the next substitution also affects the atoms introduced by the previous one.

Proposition 5.7 (Invariance under permutations of atoms)**.** *Let* π *be a permutation of finitely many atoms; the remaining atoms stay in place. Let* $\mathcal{A}^\pi$ *denote the formula obtained from* $\mathcal{A}$ *by replacing, in a single pass, each atom* a *with the atom* $\pi(a)$*, and let* Γ^π *denote the context obtained in the same way from* Γ*. Then* $\Gamma \vdash \mathcal{A}$ *if and only if* $\Gamma^\pi \vdash \mathcal{A}^\pi$*. In particular,* $\mathrm{Thm}(\mathcal{A})$ *if and only if* $\mathrm{Thm}(\mathcal{A}^\pi)$.

Proof. Induction on the derivation $\Gamma \vdash \mathcal{A}$. An instance of an axiom schema goes to an instance of the same schema, a hypothesis from Γ to a hypothesis from Γ^π, and a Modus Ponens step to a Modus Ponens step, since $(\mathcal{A} \to \mathcal{B})^\pi \equiv \mathcal{A}^\pi \to \mathcal{B}^\pi$. The converse is obtained by applying the inverse permutation. □

Bijectivity is essential: the formula $a \to a$ is a theorem and is obtained by identifying atoms in the formula $a \to b$, which is not a theorem (§ 7).

The example continued. The schema $a, \neg a \vdash b$ carries over from atoms to formulas. For formulas $\mathcal{A}, \mathcal{B}$ one chooses an atom b not occurring in $\mathcal{A}$ and an atom a different from b. Substituting $\mathcal{A}$ for a gives $\mathcal{A}, \neg\mathcal{A} \vdash b$. Since $\mathcal{A}$ does not contain the atom b, it does not occur in this context, and substituting $\mathcal{B}$ for b gives $\mathcal{A}, \neg\mathcal{A} \vdash \mathcal{B}$. By monotonicity (Proposition 5.4), any formula is derivable from every context containing a formula and its negation.

***Remark* 5.8** (Substitution in the conclusion only)**.** Substitution in the conclusion without substitution in the context does not preserve derivations: from $a \vdash a$ one would obtain $a \vdash a \wedge \neg a$, which is false for any variable a by the soundness theorem (§ 7). The consistent substitution, $a \wedge \neg a \vdash a \wedge \neg a$, is valid. Therefore substitution cannot be a rule of inference in Definition 5.2: with such a rule the deduction theorem would yield the theorem $a \to (a \wedge \neg a)$.

***Remark* 5.9** (Limits of the list-based definition)**.** Finiteness of the list suffices for the definition of derivability, since a derivation uses finitely many hypotheses. An infinite totality of hypotheses is admissible if it is given by a predicate H: rule **D2** is replaced by the rule "if $H(\mathcal{A})$, then $\mathcal{A}$ is derivable in the context H". Statements about arbitrary infinite totalities of formulas (the compactness theorem, Lindenbaum's lemma in its general form) are not formulated in the base and require a set-theoretic metatheory. The finitary base suffices for the construction of the calculus and does not cover all of its metatheory.

Part II. Models of the propositional calculus

6 The purpose of models

Part I defines theorems and derivability as properties of words. The derivability of a formula is confirmed by exhibiting a derivation, whereas its non-derivability cannot be seen directly from the definition: for that one would have to go through all possible derivations (because of Modus Ponens there arises the problem of finding the required premise $\mathcal{A}$, which is absent from the consequent). A model solves this problem. Formulas are assigned elements of some algebra (an evaluation of formulas) in such a way that all derivable formulas (theorems) receive one particular value in this algebra; formulas with other values are not derivable. The scheme of such an assignment is shown in Fig. 3: the values of atoms are given arbitrarily, the value of every other formula is determined by its structure, and all theorems receive the value 1.

The word "model" is used here in the strict mathematical sense, with a connotation entirely opposite to the one considered in § 1: there the question was whether PC models an external phenomenon, while here mathematical structures serve as models of PC. Truth tables are part of the same replacement and say nothing about ordinary logic.

A mathematical model may be used to interpret or justify a phenomenon or construction both outside mathematics and within mathematics itself. A model of the propositional calculus is precisely the second case: we construct a class of set-theoretic models of a certain formal calculus. The aim of such modelling is to transpose a purely syntactic calculus into the language of algebra (or of other branches of mathematics, for instance topology), in order, on the one hand, to make use of a more familiar set of notions and, on the other, to gain access to the tools of the developed apparatus of algebraic structures (see § 10 for an example). At the same time, the construction of a model (or, more precisely, of a class of models) demands of it a certain *exactness*: in translating from the language of the calculus into the language of algebra we must reproduce all derivable relations between formulas up to their equivalence (soundness) and take care not to introduce new ones (completeness).

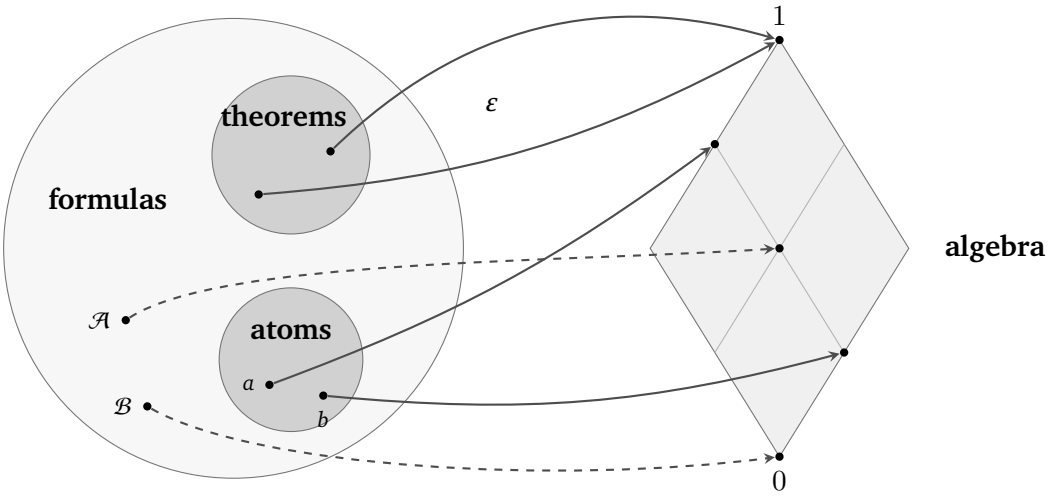


Figure 3: A model of the propositional calculus. The valuation ε assigns elements of the algebra to formulas: atoms are mapped to arbitrarily chosen elements, the values of the formulas $\mathcal{A}$ and $\mathcal{B}$ are computed from the values of their atoms, and all theorems are mapped to 1.

Moreover, this translation must be arranged so that every result about formulas can be interpreted unambiguously in the models and, conversely, every algebraic result (equation) *common* to the models of the given class can be interpreted unambiguously (modulo a strictly defined equivalence of formulas) in the calculus. This is why the two main requirements on a class of models, soundness and completeness, are decisive in choosing a class of models of a calculus. There may be several (or infinitely many) such classes; such classes may contain just a single model or many (not necessarily isomorphic) models; the way they are constructed may be anything within classical set theory; but soundness and completeness of the classes must be observed unconditionally if we want an exact algebraic picture of the calculus.

Note that it is the class of models, and not one particular model, that accomplishes sound and complete interpretation, because each particular model in the class may have some quite specific algebraic theorems that fail for the class as a whole and fail (after translation back) in the calculus. This both simplifies the situation with interpretation and complicates it. It simplifies it because the set of theorems common to the whole class of models may be considerably narrower than the theory of one particular model; it complicates it because we have to describe many models at once, keeping in mind various "edge cases".

In this part the metatheory freely uses sets, functions and relations; the finitary restrictions of Part I are lifted for convenience and for full mathematical freedom of reasoning. The notations for the metavariables $\mathcal{A}, \mathcal{B}, C$ (for formulas), a, b, c (the beginning of the Latin alphabet, for propositional variables), Γ, Δ (for contexts) and $\vdash$ (derivability) retain the same meaning as before. New: we agree to reserve part of the Latin alphabet (a few letters at the end), that is, letters such as x, y, z, for values of formulas, that is, elements of a model, the Greek ε for the valuation function, and n, m, k for natural numbers. Naturally, we shall also not hesitate to use subscripts and primes to increase the number of available variables.

7 The functional model

On the set $\mathbb{B} = \{0, 1\}$ the following operations are given: $\neg x = 1 - x$, $x \wedge y = \min(x, y)$, $x \vee y = \max(x, y)$, $x \to y = \max(1 - x, y)$. Their truth table is:

x	y	$\neg x$	$x \wedge y$	$x \vee y$	$x \to y$
0	0	1	0	0	1
0	1	1	0	1	1
1	0	0	0	1	0
1	1	0	1	1	1

Definition 7.1 (Valuation)**.** A valuation ε assigns to each atom an element 0 or 1 and is extended to all formulas by the equalities $\varepsilon(\neg\mathcal{A}) = \neg\varepsilon(\mathcal{A})$ and $\varepsilon(\mathcal{A} \circ \mathcal{B}) = \varepsilon(\mathcal{A}) \circ \varepsilon(\mathcal{B})$. The extension is well defined by unique parsing (Proposition 3.6).

The value $\varepsilon(\mathcal{A})$ depends only on the values of ε on the atoms occurring in $\mathcal{A}$. Therefore a formula all of whose atoms are among $a_1, \ldots, a_n$ defines a function $f_{\mathcal{A}} \colon \mathbb{B}^n \to \mathbb{B}$. The set B_n of all such functions with pointwise operations is called the **functional model** in n variables; here $f_{\neg\mathcal{A}} = \neg f_{\mathcal{A}}$ and $f_{\mathcal{A}\circ\mathcal{B}} = f_{\mathcal{A}} \circ f_{\mathcal{B}}$, where $\circ$ on the right-hand side is understood pointwise. The model B_n has exactly 2^{2^n} elements.

Substitution of a formula into a formula corresponds to a shift of the valuation: $\varepsilon(\mathcal{A}[\mathcal{B}/a]) = \varepsilon'(\mathcal{A})$, where the valuation ε' coincides with ε everywhere except that $\varepsilon'(a) = \varepsilon(\mathcal{B})$. This is checked by structural induction on $\mathcal{A}$.

Definition 7.2 (Tautology and semantic consequence)**.** A formula $\mathcal{A}$ is a **tautology** (notation: $\models_{\mathbb{B}} \mathcal{A}$) if $\varepsilon(\mathcal{A}) = 1$ for every valuation ε, that is, $f_{\mathcal{A}}$ is the function identically equal to 1.

A formula $\mathcal{A}$ follows logically (semantically) from a context Γ ($\Gamma \models_{\mathbb{B}} \mathcal{A}$) if for every valuation ε: if $\varepsilon(C) = 1$ for all $C \trianglelefteq \Gamma$, then $\varepsilon(\mathcal{A}) = 1$ (that is, the value 1 is transferred from the formulas of the context Γ to $\mathcal{A}$).

For empty Γ the premise holds for every valuation, so the notation $\models_{\mathbb{B}} \mathcal{A}$ agrees with the notation for a tautology.

Proposition 7.3 (Soundness)**.** *If $\Gamma \vdash \mathcal{A}$, then $\Gamma \models_{\mathbb{B}} \mathcal{A}$. In particular, every theorem is a tautology (the case of the empty context).*

Proof. Induction on the derivation. All 11 axiom schemata yield tautologies: this is a finite check by tables in three variables, and by the formula for substitution it extends to all instances. Hypotheses receive the value 1 by assumption. If $\varepsilon(\mathcal{A}) = 1$ and $\varepsilon(\mathcal{A}) \to \varepsilon(\mathcal{B}) = 1$, then $\varepsilon(\mathcal{B}) = 1$. For a theorem Proposition 5.3 (a) is applied: it is derivable in the empty context. □

Hence an atom is not a theorem (take $\varepsilon(a) = 0$), $a \nvdash a \wedge \neg a$, and the pair $\vdash \mathcal{A}$, $\vdash \neg\mathcal{A}$ is impossible (consistency of PC).

A tuple of values $\langle x_1, \ldots, x_n \rangle \in \mathbb{B}^n$ is abbreviated $\vec{x}$, so that $f(\vec{x})$ means $f(x_1, \ldots, x_n)$.

Definition 7.4 (Literal and full disjunctive normal form)**.** For an atom a and a value $s \in \mathbb{B}$, the **literal** a^s is the formula determined by the conditions $a^1 \equiv a$ and $a^0 \equiv \neg a$. For a function

$f \in B_n$ not identically zero, the **full disjunctive normal form** is the formula

$$\mathcal{D}_f \equiv \bigvee_{\substack{\vec{x}\in\mathbb{B}^n \\ f(\vec{x})=1}} \left(a_1^{x_1} \wedge \cdots \wedge a_n^{x_n}\right),$$

that is, the disjunction of conjunctions of literals over all tuples $\vec{x}$ at which f equals 1. This is notation in the metalanguage: tuples are taken in lexicographic order and parentheses in conjunctions and disjunctions are placed to the left, so that $C_1 \wedge C_2 \wedge C_3$ denotes $(C_1 \wedge C_2) \wedge C_3$.

Proposition 7.5 (Functional completeness)**.** *For $n \geqslant 1$ every function in B_n is $f_{\mathcal{A}}$ for some formula $\mathcal{A}$ with atoms among $a_1, \ldots, a_n$.*

Proof. For a function $f \in B_n$ not identically zero the equality $f_{\mathcal{D}_f} = f$ is checked. Let $\vec{y} \in \mathbb{B}^n$ and $\varepsilon(a_i) = y_i$ for all i. By the truth table, $\varepsilon(a^s) = 1$ if and only if $\varepsilon(a) = s$. Therefore the conjunction $a_1^{x_1} \wedge \cdots \wedge a_n^{x_n}$ receives the value 1 exactly when $\vec{y} = \vec{x}$, and the formula $\mathcal{D}_f$ exactly when $f(\vec{y}) = 1$. The identically zero function corresponds to the formula $a_1 \wedge \neg a_1$. □

Functional completeness means that every function is expressible by a formula and differs from logical completeness. The converse of Proposition 7.3 (everything true in the model is derivable) follows from the general construction for Boolean algebras and is proved in Theorem 9.3.

8 Boolean algebras as models

A Boolean algebra is a distributive lattice $\langle B, \wedge, \vee, 0, 1\rangle$ with complement: $x \wedge \neg x = 0$, $x \vee \neg x = 1$. In it one sets $x \to y = \neg x \vee y$, and $x \leqslant y$ means $x \wedge y = x$. Examples: $\mathbb{B}$; the functional model B_n; the set of all subsets of a set X with the set-theoretic operations. The one-element algebra, in which $0 = 1$, is not excluded.

A valuation with values in a Boolean algebra B is given and extended as in Definition 7.1. The notation $\Gamma \models_{\mathsf{BA}} \mathcal{A}$ means: in **every** Boolean algebra B, every valuation ε such that $\varepsilon(C) = 1$ for all $C \trianglelefteq \Gamma$ satisfies $\varepsilon(\mathcal{A}) = 1$. The relation $\models_{\mathbb{B}}$ is the special case with the single model $\mathbb{B}$.

Lemma 8.1.

(a) *$x \to y = 1$ if and only if $x \leqslant y$.*

(b) *A Boolean identity true in $\mathbb{B}$ is true in every Boolean algebra.*

Proof. (a) If $x \leqslant y$, then $\neg x \vee y \geqslant \neg x \vee x = 1$. If $\neg x \vee y = 1$, then $x = x \wedge (\neg x \vee y) = x \wedge y$. (b) The laws of Boolean algebra reduce each side of the identity to full disjunctive normal form, which is determined by the table of values in $\mathbb{B}$; the forms of both sides coincide. □

Proposition 8.2 (Soundness with respect to the class of Boolean algebras)**.** *If $\Gamma \vdash \mathcal{A}$, then $\Gamma \models_{\mathsf{BA}} \mathcal{A}$.*

Proof. Induction on the derivation. The axioms are tautologies, so by Lemma 8.1 (b) they receive the value 1 in every Boolean algebra. Hypotheses receive the value 1 by assumption. If $\varepsilon(\mathcal{A}) = 1$ and $\varepsilon(\mathcal{A}) \to \varepsilon(\mathcal{B}) = 1$, then by Lemma 8.1 (a) we have $1 \leqslant \varepsilon(\mathcal{B})$. □

9 Completeness

9.1 Completeness with respect to the class of Boolean algebras

A model refuting a non-derivable formula is constructed from the syntax itself. A context Γ is fixed. The notation $\mathcal{A} \preceq \mathcal{B}$ means $\Gamma \vdash \mathcal{A} \to \mathcal{B}$; formulas are equivalent, $\mathcal{A} \sim \mathcal{B}$, if $\mathcal{A} \preceq \mathcal{B}$ and $\mathcal{B} \preceq \mathcal{A}$.

Lemma 9.1. *The following properties hold; their verification uses the deduction theorem (Proposition 5.5).*

(1) $\preceq$ *is reflexive (the theorem* $\mathcal{A} \to \mathcal{A}$*) and transitive;*

(2) $\mathcal{A} \wedge \mathcal{B}$ *is the greatest lower bound of* $\mathcal{A}$ *and* $\mathcal{B}$ *with respect to* $\preceq$ *(conjunction elimination and introduction);*

(3) $\mathcal{A} \vee \mathcal{B}$ *is the least upper bound (disjunction introduction and constructive dilemma);*

(4) $\mathcal{A} \wedge (\mathcal{B} \vee C) \preceq (\mathcal{A} \wedge \mathcal{B}) \vee (\mathcal{A} \wedge C)$ *(constructive dilemma in a context with the hypothesis* $\mathcal{A}$*);*

(5) $\mathcal{A} \wedge \neg\mathcal{A} \preceq \mathcal{B}$ *(law of Duns Scotus),* $\mathcal{B} \preceq \mathcal{A} \vee \neg\mathcal{A}$ *(law of excluded middle), and* $\mathcal{A} \preceq \mathcal{B}$ *implies* $\neg\mathcal{B} \preceq \neg\mathcal{A}$ *(negation introduction);*

(6) $(\mathcal{A} \to \mathcal{B}) \sim (\neg\mathcal{A} \vee \mathcal{B})$ *(law of excluded middle for* $\mathcal{A}$*, constructive dilemma and law of Duns Scotus);*

(7) $\Gamma \vdash \mathcal{A}$ *if and only if* $\mathcal{B} \preceq \mathcal{A}$ *for every formula* $\mathcal{B}$*.*

By the lemma, $\sim$ is a congruence, and the classes $[\mathcal{A}]$ with the operations $[\mathcal{A}] \wedge [\mathcal{B}] = [\mathcal{A} \wedge \mathcal{B}]$, $[\mathcal{A}] \vee [\mathcal{B}] = [\mathcal{A} \vee \mathcal{B}]$, $\neg[\mathcal{A}] = [\neg\mathcal{A}]$ form a Boolean algebra LT_Γ, the **Lindenbaum–Tarski algebra** of the context Γ; for the empty context it is denoted LT. By item (7), its unit is the class of formulas derivable in the context. By item (6), $[\mathcal{A} \to \mathcal{B}] = [\mathcal{A}] \to [\mathcal{B}]$.

Translation dictionary. The construction LT_Γ translates the calculus into the language of algebra. The order of the Boolean algebra is the relation $\preceq$: the inequality $[\mathcal{A}] \leqslant [\mathcal{B}]$ means $\Gamma \vdash \mathcal{A} \to \mathcal{B}$, that is, by the deduction theorem, $\Gamma, \mathcal{A} \vdash \mathcal{B}$. Therefore, for a context consisting of one formula, the derivability sign translates into the inequality sign. For the algebra LT the translation is as follows:

calculus	algebra LT
formula $\mathcal{A}$	element $[\mathcal{A}]$
connectives $\neg, \wedge, \vee, \to$	operations $\neg, \wedge, \vee, \to$
equivalence $\mathcal{A} \sim \mathcal{B}$	equality $[\mathcal{A}] = [\mathcal{B}]$
$\vdash \mathcal{A}$	$[\mathcal{A}] = 1$
$\vdash \neg\mathcal{A}$	$[\mathcal{A}] = 0$
$\mathcal{A} \vdash \mathcal{B}$	$[\mathcal{A}] \leqslant [\mathcal{B}]$
$C_1, \dots, C_k \vdash \mathcal{B}$	$[C_1] \wedge \dots \wedge [C_k] \leqslant [\mathcal{B}]$
$C, \mathcal{A} \vdash \mathcal{B}$ is equivalent to $C \vdash \mathcal{A} \to \mathcal{B}$	$z \wedge x \leqslant y$ is equivalent to $z \leqslant x \to y$
rule Modus Ponens	$x \wedge (x \to y) \leqslant y$
valuation ε	homomorphism $\mathsf{LT} \to \mathbb{B}$
permutation of atoms π	automorphism $[\mathcal{A}] \mapsto [\mathcal{A}^\pi]$

Theorem 9.2 (Completeness with respect to the class of Boolean algebras)**.** $\Gamma \vdash \mathcal{A}$ *if and only if* $\Gamma \models_{\mathsf{BA}} \mathcal{A}$.

Proof. The direct statement is Proposition 8.2. Let $\Gamma \nvdash \mathcal{A}$. For the canonical valuation $\varepsilon(a) = [a]$ in LT_Γ, structural induction gives $\varepsilon(\mathcal{B}) = [\mathcal{B}]$ for all formulas $\mathcal{B}$. For $C \trianglelefteq \Gamma$ we have $\Gamma \vdash C$, so $\varepsilon(C) = 1$, whereas $\varepsilon(\mathcal{A}) = [\mathcal{A}] \neq 1$. □

9.2 Reduction to a single model

Theorem 9.3 (E. Post, 1921)**.** $\Gamma \vdash \mathcal{A}$ *if and only if* $\Gamma \models_{\mathbb{B}} \mathcal{A}$*. In particular,* $\mathrm{Thm}(\mathcal{A})$ *if and only if* $\mathcal{A}$ *is a tautology.*

Proof. The direct statement is Proposition 7.3. Let $\Gamma \nvdash \mathcal{A}$. By Lemma 9.1 (7) we have $[\mathcal{A}] \neq 1$, so $\neg[\mathcal{A}] \neq 0$, and there is an ultrafilter U of the algebra LT_Γ containing $\neg[\mathcal{A}]$.[1] The map $h\colon \mathsf{LT}_\Gamma \to \mathbb{B}$, where $h(y) = 1$ for $y \in U$ and $h(y) = 0$ otherwise, is a homomorphism. Indeed, by the definition of an ultrafilter, $\neg y \in U$ if and only if $y \notin U$, and $y \wedge z \in U$ if and only if $y \in U$ and $z \in U$; for disjunction and implication the equalities follow from these two, since $y \vee z = \neg(\neg y \wedge \neg z)$ and $y \to z = \neg y \vee z$. The valuation with $\varepsilon(a) = h([a])$ for all atoms a gives, by structural induction, $\varepsilon(\mathcal{B}) = h([\mathcal{B}])$ for all formulas $\mathcal{B}$. For $C \trianglelefteq \Gamma$ we have $[C] = 1 \in U$, whence $\varepsilon(C) = 1$, and $\neg[\mathcal{A}] \in U$ implies $\varepsilon(\mathcal{A}) = 0$. Consequently, $\Gamma \not\models_{\mathbb{B}} \mathcal{A}$. □

Corollary 9.4. *Formulas* $\mathcal{A}, \mathcal{B}$ *with atoms among* $a_1, \dots, a_n$ *are equivalent in the empty context if and only if* $f_{\mathcal{A}} = f_{\mathcal{B}}$*. The map* $[\mathcal{A}] \mapsto f_{\mathcal{A}}$ *is an isomorphism of the subalgebra of* LT *generated by the classes* $[a_1], \dots, [a_n]$ *onto the functional model* B_n*.*

Proof. $\mathcal{A} \sim \mathcal{B}$ means $\vdash \mathcal{A} \to \mathcal{B}$ and $\vdash \mathcal{B} \to \mathcal{A}$; by Theorem 9.3 this is equivalent to $f_{\mathcal{A}} = f_{\mathcal{B}}$. Hence the map is well defined and injective. Surjectivity is Proposition 7.5. □

***Remark* 9.5** (A class of models and a single model)**.** Theorem 9.2 is obtained by a direct construction: the model LT_Γ is built from derivability itself. The substantive step is Theorem 9.3: for PC one finite model suffices.

10 Computation in the model: Zhegalkin polynomials

The translation into the language of algebra yields not only a check of derivability by going through the rows of a truth table, but also computation by the rules of arithmetic. For this purpose the Boolean algebra is rebuilt into a ring.

In a Boolean algebra one sets $xy = x \wedge y$ and $x + y = (x \wedge \neg y) \vee (\neg x \wedge y)$. On $\mathbb{B}$ these are multiplication and addition modulo 2. Therefore $\mathbb{B}$ satisfies the laws of a commutative ring with unit, the identities $x^2 = x$ and $x + x = 0$, and also the equalities

$$\neg x = 1 + x, \qquad x \vee y = x + y + xy, \qquad x \to y = 1 + x + xy.$$

By Lemma 8.1 (b) all this holds in every Boolean algebra: it is a commutative ring with unit in which $x^2 = x$ (a *Boolean ring*; M. Stone, 1936), and the connectives are expressed through the ring operations.

Definition 10.1 (Zhegalkin polynomial)**.** A **monomial** in the atoms $a_1, \dots, a_n$ is a product of distinct atoms; the empty product equals 1. A **Zhegalkin polynomial** is a sum of distinct monomials; the empty sum equals 0. To a formula $\mathcal{A}$ with atoms among $a_1, \dots, a_n$ there is

[1] An *ultrafilter* of a Boolean algebra B is a set $U \subseteq B$ such that: $1 \in U$; $y, z \in U$ implies $y \wedge z \in U$; $y \in U$ and $y \leqslant z$ imply $z \in U$; for every $y \in B$ exactly one of the elements y and $\neg y$ belongs to U. In a countable Boolean algebra every nonzero element z_0 belongs to some ultrafilter $U(z_0)$, and an enumeration $y_1, y_2, \dots$ of the elements of the algebra suffices for its construction (the algebraic form of Lindenbaum's lemma). One sets $z_k = z_{k-1} \wedge y_k$ if this element is nonzero, and $z_k = z_{k-1} \wedge \neg y_k$ otherwise; in both cases $z_k \neq 0$, since $z_{k-1} = (z_{k-1} \wedge y_k) \vee (z_{k-1} \wedge \neg y_k)$. The required ultrafilter is $U(z_0) = \{y : z_k \leqslant y \text{ for some } k\}$. The condition on $\wedge$ follows from the chain z_k being decreasing; for every k the ultrafilter $U(z_0)$ contains y_k or $\neg y_k$, and it cannot contain both, since otherwise some $z_m \leqslant y_k \wedge \neg y_k = 0$. The algebra LT_Γ is countable, and its enumeration is given by an enumeration of formulas as words over a finite alphabet. The ultrafilters $U(z)$, $z \neq 0$, separate the elements of a countable algebra B: for $y \neq w$ and $z = (y \wedge \neg w) \vee (\neg y \wedge w)$ exactly one of the elements y, w belongs to $U(z)$. Therefore the map $y \mapsto \{z \in B \setminus \{0\} : y \in U(z)\}$ is an embedding of B into $\mathcal{P}(B \setminus \{0\})$; this is a special case of the representation theorem for Boolean algebras (M. Stone, 1936).

assigned the polynomial $p_{\mathcal{A}}$: $p_{a_i} = a_i$ and

$$p_{\neg\mathcal{A}} = 1 + p_{\mathcal{A}}, \qquad p_{\mathcal{A}\wedge\mathcal{B}} = p_{\mathcal{A}}p_{\mathcal{B}},$$
$$p_{\mathcal{A}\vee\mathcal{B}} = p_{\mathcal{A}} + p_{\mathcal{B}} + p_{\mathcal{A}}p_{\mathcal{B}}, \qquad p_{\mathcal{A}\to\mathcal{B}} = 1 + p_{\mathcal{A}} + p_{\mathcal{A}}p_{\mathcal{B}},$$

where parentheses are expanded and like terms are collected by the rules $x^2 = x$, $x + x = 0$.

Theorem 10.2 (I. I. Zhegalkin, 1927)**.** *For $n \geqslant 1$ every function in B_n is written by a unique Zhegalkin polynomial in n variables, and the function $f_{\mathcal{A}}$ is written by the polynomial $p_{\mathcal{A}}$.*

Proof. The second statement is checked by structural induction using the above equalities. By Proposition 7.5 every function in B_n is $f_{\mathcal{A}}$ and is therefore written by a polynomial. There are 2^n monomials in n atoms and 2^{2^n} polynomials, that is, as many as there are functions in B_n. Therefore the notation is unique. □

Corollary 10.3. *For formulas $\mathcal{A}, \mathcal{B}$ with atoms among $a_1, \ldots, a_n$:*

(a) $\vdash \mathcal{A}$ *if and only if* $p_{\mathcal{A}} = 1$*;*

(b) $\mathcal{A}$ *and* $\mathcal{B}$ *are equivalent in the empty context if and only if* $p_{\mathcal{A}} = p_{\mathcal{B}}$*;*

(c) $\mathcal{A} \vdash \mathcal{B}$ *if and only if* $p_{\mathcal{A}}p_{\mathcal{B}} = p_{\mathcal{A}}$.

Proof. By Theorem 9.3 and Corollary 9.4, derivability and equivalence reduce to equalities of functions, and by Theorem 10.2 functions are uniquely written by polynomials. Item (c) is item (a) for the formula $\mathcal{A} \to \mathcal{B}$. □

Example. Peirce's law $((a \to b) \to a) \to a$ is checked in two lines. We have $p_{a\to b} = 1 + a + ab$, whence

$$p_{(a\to b)\to a} = 1 + (1 + a + ab) + (1 + a + ab)a = a + ab + ab = a,$$

that is, the formula $(a \to b) \to a$ is equivalent to the atom a. Therefore $p_{((a\to b)\to a)\to a} = 1+a+a\cdot a = 1$, and the formula is derivable; the derivation from the axioms itself is not required for this.

General statements. A truth table checks each formula separately, whereas computation in the ring yields statements about infinite families of formulas. The expression $\mathcal{A} \leftrightarrow \mathcal{B}$ abbreviates $(\mathcal{A} \to \mathcal{B}) \wedge (\mathcal{B} \to \mathcal{A})$. Its polynomial is linear: $p_{\mathcal{A}\leftrightarrow\mathcal{B}} = 1 + p_{\mathcal{A}} + p_{\mathcal{B}}$.

Proposition 10.4 (S. Leśniewski, 1929)**.** *A formula built from atoms using the connective $\leftrightarrow$ alone is a theorem if and only if each atom occurs in it an even number of times.*

Proof. By induction on the construction: the polynomial of a formula with m occurrences of atoms equals the sum of these occurrences and $m - 1$ ones. An atom with an even number of occurrences cancels in the sum. If all atoms are such, then m is even and the polynomial equals 1. Otherwise an atom remains in the polynomial, and it differs from 1. It remains to apply Corollary 10.3 (a). □

Context and ideal. In the ring a context translates into an ideal: $C_1, \ldots, C_k \vdash \mathcal{B}$ if and only if $1 + p_{\mathcal{B}}$ lies in the ideal generated by the polynomials $1 + p_{C_i}$. If $1 + p_{\mathcal{B}}$ lies in this ideal, then wherever all p_{C_i} equal 1, the polynomial $p_{\mathcal{B}}$ also equals 1, and Theorem 9.3 applies. Conversely, for $q = p_{C_1} \cdots p_{C_k}$, derivability means $q(1 + p_{\mathcal{B}}) = 0$, whence $1 + p_{\mathcal{B}} = (1 + q)(1 + p_{\mathcal{B}})$, and $1 + q$ lies in the ideal. The homomorphism h from the proof of Theorem 9.3 is an epimorphism onto $\mathbb{F}_2$, and the complement of the ultrafilter U is its kernel, a maximal ideal of the ring LT_Γ.

11 Duality, uniqueness of the two-valued model and Post completeness

The truth table of § 7 is not selected from among others: the axioms and Modus Ponens determine it uniquely up to the permutation of the values 0 and 1, and this permutation corresponds to passing to dual formulas. Nor can the calculus be extended while preserving consistency and closure under substitution.

Definition 11.1 (Dual formula)**.** The formula $\mathcal{A}^*$ is defined by recursion: $a^* \equiv a$ for atoms, $(\neg\mathcal{A})^* \equiv \neg\mathcal{A}^*$, $(\mathcal{A}\wedge\mathcal{B})^* \equiv \mathcal{A}^*\vee\mathcal{B}^*$, $(\mathcal{A}\vee\mathcal{B})^* \equiv \mathcal{A}^*\wedge\mathcal{B}^*$, $(\mathcal{A}\to\mathcal{B})^* \equiv \neg\mathcal{A}^*\wedge\mathcal{B}^*$.

Theorem 11.2 (Duality principle; C. Peirce, E. Schröder)**.** *If the atoms of a formula $\mathcal{A}$ are among $a_1,\dots,a_n$, then $f_{\mathcal{A}^*}(x_1,\dots,x_n) = \neg f_{\mathcal{A}}(\neg x_1,\dots,\neg x_n)$. Consequently:*

(a) *$\mathcal{A}\vdash\mathcal{B}$ if and only if $\mathcal{B}^*\vdash\mathcal{A}^*$; in particular, formulas $\mathcal{A},\mathcal{B}$ are equivalent in the empty context if and only if $\mathcal{A}^*,\mathcal{B}^*$ are equivalent;*

(b) *$\vdash\mathcal{A}$ if and only if $\vdash\neg\mathcal{A}^*$.*

Proof. The equality is checked by structural induction. For an atom it is $x = \neg\neg x$, for negation it follows from the induction hypothesis, and for the remaining connectives it is a de Morgan law in $\mathbb{B}$. For example, for implication, with $\vec{y} = \langle\neg x_1,\dots,\neg x_n\rangle$, we have

$$f_{\neg\mathcal{A}^*\wedge\mathcal{B}^*}(\vec{x}) = f_{\mathcal{A}}(\vec{y}) \wedge \neg f_{\mathcal{B}}(\vec{y}) = \neg\bigl(\neg f_{\mathcal{A}}(\vec{y}) \vee f_{\mathcal{B}}(\vec{y})\bigr) = \neg f_{\mathcal{A}\to\mathcal{B}}(\vec{y}).$$

By Theorem 9.3 the derivability $\mathcal{A}\vdash\mathcal{B}$ means $f_{\mathcal{A}} \leqslant f_{\mathcal{B}}$ pointwise, and this inequality is equivalent to $f_{\mathcal{B}^*} \leqslant f_{\mathcal{A}^*}$. The derivability $\vdash\mathcal{A}$ means that $f_{\mathcal{A}}$ is identically equal to 1, which is equivalent to $f_{\mathcal{A}^*}$ being identically equal to 0. □

In terms of the dictionary, duality reverses the order: the map $y \mapsto \neg y$ is an isomorphism of a Boolean algebra onto the algebra in which $\wedge$ and $\vee$, 0 and 1 exchange roles. Therefore together with every identity its dual holds; thus distributivity $x\wedge(y\vee z) = (x\wedge y)\vee(x\wedge z)$ implies $x\vee(y\wedge z) = (x\vee y)\wedge(x\vee z)$, and to the theorem $a\vee\neg a$ there corresponds the theorem $\neg(a\wedge\neg a)$.

Proposition 11.3 (Uniqueness of the two-valued model)**.** *Let operations $\neg,\wedge,\vee,\to$ be given on a two-element set, along which valuations are extended to formulas, and let one element be distinguished. It is denoted 1, and the other element 0. If $x = 1$ and $x\to y = 1$ imply $y = 1$, and all axioms receive the value 1 under every valuation, then these are the operations of § 7.*

Proof. The condition on Modus Ponens gives $1\to 0 = 0$. Affirmation of the consequent for $x = 1$ gives $1\to(y\to 1) = 1$, whence $y\to 1 = 1$; for $x = y = 0$ it gives $0\to(0\to 0) = 1$, whence $0\to 0 = 1$. Thus implication coincides with the one in the table, and $x\to y = 1$ is equivalent to $x\leqslant y$. Conjunction elimination gives $x\wedge y \leqslant \min(x,y)$, and conjunction introduction for $x = y = 1$ gives $1\wedge 1 = 1$. Disjunction introduction gives $x\vee y \geqslant \max(x,y)$, and constructive dilemma for $x = y = z = 0$ gives $(0\vee 0)\to 0 = 1$, that is, $0\vee 0 = 0$. The law of Duns Scotus for $x = 1$, $y = 0$ gives $\neg 1\to 0 = 1$, that is, $\neg 1 = 0$. Negation introduction for $x = y = 0$ gives $1\to(1\to\neg 0) = 1$, that is, $\neg 0 = 1$. □

The number of tuples of operations on $\{0,1\}$ equals $4\cdot 16^3 = 16384$, and for each of the two distinguished values exactly one tuple is a model of PC. For the distinguished value 1 this is $\mathbb{B}$. For the distinguished value 0 this is the *flipped* model: $\neg x = 1-x$, $x\wedge y = \max(x,y)$, $x\vee y = \min(x,y)$, $x\to y = \min(1-x,y)$. The permutation of values is an isomorphism between these two models. The value of a formula $\mathcal{A}$ in the flipped model equals the value of $\mathcal{A}^*$ in $\mathbb{B}$ under the same valuation,

so computation in the flipped model is computation of the dual formula: a formula $\mathcal{A}$ is a theorem if and only if $f_{\mathcal{A}^*}$ is identically equal to 0.

By Theorem 9.3 the theorems are exactly the tautologies, so every calculus in the same language that is sound with respect to $\mathbb{B}$ is contained in PC. With closure under substitution preserved, PC can be extended only to an inconsistent calculus.

Proposition 11.4 (Post completeness)**.** *If $\mathcal{A}$ is not a theorem, then on adding to the axioms all formulas obtained from $\mathcal{A}$ by substitutions for atoms, every formula becomes derivable.*

Proof. By Theorem 9.3 there is a valuation ε with $\varepsilon(\mathcal{A}) = 0$. Let c be an atom not occurring in $\mathcal{A}$. The formula $c \to c$ has the value 1 under every valuation, and the formula $\neg(c \to c)$ has the value 0. Substituting for each atom a of the formula $\mathcal{A}$ the first of them if $\varepsilon(a) = 1$ and the second if $\varepsilon(a) = 0$ gives a formula $\mathcal{A}'$ whose value under every valuation equals $\varepsilon(\mathcal{A}) = 0$. Therefore $\neg\mathcal{A}'$ is a tautology and, by Theorem 9.3, a theorem. In the extended calculus both $\mathcal{A}'$ and $\neg\mathcal{A}'$ are derivable, whence by the law of Duns Scotus every formula is derivable. □

This result was obtained by D. Hilbert and P. Bernays (1918; see Zach, 1999) and by E. Post (1921). Consequently, every consistent calculus in the same language that is closed under substitution and Modus Ponens and differs from PC fails to contain some of its theorems. Such is the position of intuitionistic logic, relevance logics and many-valued logics lying between intuitionistic and classical logic: their theorems form a part of the theorems of PC.

Summary

The construction of Part I consists of the following stages.

1. PC is introduced as an explicatum: a formal system replacing ordinary reasoning in the functions required by mathematical proof. Its correspondence with ordinary logic is not a theorem and is not established by the completeness theorem; its domain of application is mathematics, including computer science.

2. The finitary base of the metalanguage is fixed: a finite list of signs, words and concatenation, graphic identity ≡, predicates, ordinary logic in simple formulations, recursive construction of classes of words, and metavariables.

3. The alphabet $\mathfrak{S}$ and the words $\mathfrak{S}^*$ with metavariables ξ, η, ζ are given; the predicates Atom and Form with metavariables a, b, c and $\mathcal{A}, \mathcal{B}, \mathcal{C}$ are constructed; the substitution $\mathcal{A}[\mathcal{B}/a]$, which preserves formulas, is introduced; unique parsing is proved.

4. The axioms are given by schemata, and the theorem predicate Thm by closure under Modus Ponens.

5. Finite lists of formulas are defined as words over the extended alphabet $\mathfrak{S}'$; derivability $\Gamma \vdash \mathcal{A}$ is given as a predicate on formulas in a context Γ, which may be empty; it is preserved under substitution carried out consistently in the context and in the conclusion.

6. The notion of finite iteration and contentual arithmetic are retained (Remark 2.1); statements about arbitrary infinite totalities of hypotheses are not formulated in the base.

Part II translates the calculus into the language of algebra: formulas are assigned values, and the main requirements on a class of models are soundness and completeness. The functional model B_n reduces a formula with n atoms to a Boolean function, and every such function is given by a formula. Boolean algebras form a class of models with respect to which PC is complete by virtue of the construction LT_Γ; this construction yields a translation dictionary in which derivability corresponds to the order of the algebra. Post's theorem reduces the class to the single model $\mathbb{B}$,

and Corollary 9.4 identifies B_n with a fragment of the Lindenbaum–Tarski algebra. In the Boolean ring formulas are written by Zhegalkin polynomials, and logical computations reduce to arithmetic modulo 2. The two-valued model is unique up to the permutation of values, and this permutation corresponds to passing to dual formulas. PC cannot be extended while preserving consistency and closure under substitution.

References

Carnap R. Logical Foundations of Probability. Chicago: University of Chicago Press, 1950.

Carroll L. What the Tortoise said to Achilles // Mind. 1895. Vol. 4.

Hilbert D., Bernays P. Grundlagen der Mathematik. Vol. 1. Berlin: Springer, 1934.

Kazimirov N. I. The Architecture of Mathematical Models: course notes, subsection “The Lindenbaum–Tarski algebra”.

Kazimirov N. I. Mathematics as a Foreign Language. mathem.at/Materials/maglish_index_en.html

Kleene S. C. Introduction to Metamathematics. Amsterdam: North-Holland, 1952.

Kreisel G. Informal rigour and completeness proofs // Problems in the Philosophy of Mathematics / ed. I. Lakatos. Amsterdam: North-Holland, 1967.

Markov A. A. Theory of algorithms // Trudy Mat. Inst. Steklov. 1954. Vol. 42 (in Russian).

Post E. Introduction to a general theory of elementary propositions // American Journal of Mathematics. 1921. Vol. 43.

Quine W. V. O. Word and Object. Cambridge, MA: MIT Press, 1960.

Rasiowa H., Sikorski R. The Mathematics of Metamathematics. Warsaw: PWN, 1963.

Zach R. Completeness before Post: Bernays, Hilbert, and the development of propositional logic // Bulletin of Symbolic Logic. 1999. Vol. 5.

Zhegalkin I. I. On the technique of calculating propositions in symbolic logic // Matematicheskii Sbornik. 1927. Vol. 34, no. 1 (in Russian).