Open Journal of Mathematics and Physics | Volume 3, Article 252, ISSN: 2674-5747
https://doi.org/10.31219/osf.io/8wck9 | published: 24 Oct 2021 | https://ojmp.org

[white paper]
Diamond Open Access
[awaiting peer review]

Personal Handbook of Logic

Open Mathematics Collaboration*†

February 9, 2023

Abstract

This is a personal collection of definitions and results from first-order logic.

keywords: personal handbook, first-order logic

The most updated version of this white paper is available at
https://osf.io/8wck9/download
https://zenodo.org/record/5594984

Preamble

1. *Mathematics is the Queen of the Sciences* (Gauss).

2. *Logic is the Queen of Mathematics.*

*All *authors* with their *affiliations* appear at the end of this white paper.
†Corresponding author: mplobo@uft.edu.br | **Open Mathematics Collaboration**

Contents

Satisfaction for all variable assignment functions

Model `(formula)`

True Sentences

Satisfaction of formulas with the connective "and"

Satisfaction with the existential quantifier

Substitution into a Term

Substitution into a Formula

A term substitutable for a variable in a formula

Logical Implication `(sets of formulas)`

Valid Formulas

Universal Closure of a Formula

On the validity of a conditional statement of formulas

"Bottom-up" Deduction

"Top-down" Deduction

Decidable Set of Axioms

(Non)Logical Axioms

Equality Axioms

Quantifier Axioms

Rules of Inference

Propositional Consequence: Definition

Propositional Consequence: Tautology

Propositional Consequence: Extension to First-order Logic

Rule of Inference of type (PC)

Rules of Inference of type (QR)

On the validity and tautology of formulas

List of requirements for axioms and rules of inference

Logical Axioms: Valid

Rule of Inference: Theorem

Soundness Theorem

When a variable is not free in a formula

Variable Assignment Functions and Substitutions

Term substitution in the x-modification of the assignment function

Equality: Equivalence Relation

A set of formulas proves a formula if and only if it proves the formula for all variables

Adding/deleting a universal quantifier

The Deduction Theorem

Proofs by Contradiction

Unary Relation Symbol

Binary Relation Symbol

Two unary relation symbols

Complete Deductive System

(In)Consistent

Completeness Theorem

Soundness + Completeness

Compactness Theorem

(Finitely) Satisfiable

Finite subset of a set of formulas

First-order Sentences: Natural Numbers

Theory of a Structure

Substructure

Elementary Substructure/Extension

Truth in elementary substructure/extension

Condition for an elementary substructure

Hilbert Axiomatic System

Axioms of the Hilbert-style Calculus

Inference Rule of the Hilbert-style Calculus

Sequent Systems: Classical Logic

Proofs and Provability (in LK)

Rules for single formulas (in LK)

Multisets of Formulas

Logical constant 0

Orthologic

Intuitionistic Reasoning

Intuitionistic Propositional Logic: Syntax

Axiom Schema for Intuitionistic Propositional Logic

Modal operators

Decision problem

Undecidable

Word problem

Natural Deduction in Heyting Semantics

Lambda Calculus: Types

Lambda Calculus: Terms

Lambda Calculus: Denotational significance

System of equations in lambda calculus:
Consistent and decidable

Conversion

The Curry-Howard Isomorphism

The Normalization Theorem

The lambda-calculus: Introduction

The lambda-calculus: Formal system

The lambda-calculus: Informal interpretation

Lambda-terms: Length, occurrence, scope, free and bound variables, substitution

Lambda-terms: Change of bound variables, congruence

Lambda-terms: Simultaneous substitution

Lambda-terms: β-reduction

Lambda-terms: β-normal form

Lambda-terms: β-equality

Simple typing, Church-style

Typed λ-calculus

The Sequent Calculus LJ

The Sequent Calculus LJT

Translation of proofs from LJ to LJT

Proofs in Natural Deduction

Proof of sequents in LK

Proof of sequents in LJ and in LJT

Quantum Logics: Introduction

Introduction

3. This handbook is mostly intended for consultation.

4. Each section can be read independently.

5. Due to (3) and (4), there are redundancies in many of the definitions.

6. At the beginning of each section, we present the references used.

Metalinguistic Symbols

7. [1,2]

8. A `metalinguistic` symbol is not part of the language.

9. The symbol := means that *what is on the left side is defined by the right side of it.*

10. The symbol :≡ means that *the strings of symbols (within a language) on each side of it are identical.*

11. $\vdash$ means *deduction, logically implies*

12. $\models$ means *satisfy, truth (if there is a structure on the left), logical implication (if there is a set of sentences on the left), model*

13. $\perp$ (read *false* or *eet*) := contradictory sentence

14. The symbol $\sim\!\succ$ appears for pedagogical purpose for the sake of abbreviating an explanation; it can be read as *from, of, with, leads to, in which, etc.*

Symbols and Syntax

15. [1,2]

16. `syntax` := *symbols* (of a *language*)

17. `string` := *string of symbols* := a sequence of symbols

List of Symbols

18. [1, 2]

19. $\in$:= membership relation

20. $\notin$:= negation of the membership relation

21. $\veebar$:= exclusive or

22. $\subseteq$:= subset, substructure

23. $\prec$:= elementary substructure/extension

24. $\cup$:= union of sets

25. $\cap$:= intersection of sets

26. $\emptyset$:= empty set

27. $f\restriction_A$:= restriction of the function f to the domain A

First-order Language

28. [1, 2]

29. `first-order language` := *infinite collection of distinct symbols* (no one of which is properly contained in another) separated into the following:

    (a) *Parentheses:* (,)

    (b) *Connectives:* $\vee, \neg$

(c) *Quantifier:* $\forall$

(d) *Variables* (one for each positive integer n): $v_1, v_2, ..., v_n, ...$
Vars $= \{v_1, v_2, v_3, ...\} :=$ set of variable symbols

(e) *Equality:* $=$

(f) *Constant:* Some set of zero or more symbols

(g) *Function:* For each positive integer n, some set of zero or more n-ary function symbols

(h) *Relation:* For each positive integer n, some set of zero or more n-ary relation symbols

Terms

30. [1, 2]

31. `term` of $\mathcal{L}$:= *nonempty finite string* t of *symbols* from $\mathcal{L}$ such that either:

    (a) $t :=$ *constant symbol* (c), or

    (b) $t :=$ *variable* (v), or

    (c) $t :\equiv ft_1t_2...t_n$, where
    $f := n$-ary function symbol of $\mathcal{L}$ and
    $t_i :=$ term of $\mathcal{L}$.

32.

$$t := c \veebar v \veebar f$$

33. $\mathcal{L} :=$ first-order language

34. $\mathcal{L}$-`symbols` := *symbols of a language* $\mathcal{L}$

35. Note that (31.c) is a <u>definition by *recursion*</u>, since t is a <u>*term*</u> *if it contains substrings that are terms.*

36. `substring` := *subset* of a *string*

Formulas

37. [1–3]

38. `formulas` := assertions about the objects of the structure (model)

39. `formula` of $\mathcal{L}$:≡ *nonempty finite string* ϕ of *symbols* from $\mathcal{L}$ such that either:

    (a) $\phi :\equiv\ = t_1 t_2$, or

    (b) $\phi :\equiv R t_1 t_2 ... t_n$, or

    (c) $\phi :\equiv (\neg\alpha)$, or

    (d) $\phi :\equiv (\alpha \vee \beta)$, or

    (e) $\phi :\equiv (\forall v)(\alpha)$.

40. $\mathcal{L}$:= first-order language

41. $t_1, t_2, ..., t_n$:= *terms* of $\mathcal{L}$

42. R := n-ary relation symbol of $\mathcal{L}$

43. α, β := *formulas* of $\mathcal{L}$

44. v := *variable*

45. Note that (39.c, d, e) are <u>definitions by *recursion*</u>, since ϕ is a <u>*formula*</u> *if it contains other formulas.*

46. In (39.e), we say that the *scope* of the quantifier $\forall$ is α.

47. $p \wedge \neg p$ has two *formula occurrences* of p

Atomic Formulas

48. [1, 2]

49. `atomic formula` of $\mathcal{L}$:= *nonempty finite string* ϕ of *symbols* from $\mathcal{L}$ such that either:

   (a) $\phi :\equiv \; = t_1 t_2$, or

   (b) $\phi :\equiv Rt_1 t_2 ... t_n$.

50. $\mathcal{L}$:= first-order language

51. $t_1, t_2, ..., t_n$:= *terms* of $\mathcal{L}$

52. R := n-ary relation symbol of $\mathcal{L}$

53. *Atomic formulas* are the *primitives* (not defined under recursion).

54. `atom` := atomic formula

55. `literal` := *atom* or *its negation*

Complexity

56. [1, 2]

57. `simpler formula` := *fewer number of connectives/quantifiers*

58. `simpler formula` := *subformula of a more complex formula*

Mathematical Induction

59. [1, 2]

60. `proof by induction` on the *structure (complexity)* of the *formula*

Free Variables

61. [1, 2]

62. $v :=$ `free` in ϕ *if*

   (*a*) ϕ is *atomic* and v occurs in (is a symbol in) ϕ, or
   (*b*) $\phi :\equiv (\neg\alpha)$ and v is free in α, or
   (*c*) $\phi :\equiv (\alpha \vee \beta)$ and v is free in at least α or β, or
   (*d*) $\phi :\equiv (\forall u)(\alpha)$ and v is not u and v is free in α.

63. $u, v :=$ variables

64. $\phi, \alpha, \beta :=$ formulas

Sentences

65. [1, 2]

66. `sentences` := *formulas* that can be either **true** or **false** (with **no** *free variables*)

67. There are no free variables in the definition of a sentence so that it can be either true or false.

68. $\mathcal{L} :=$ first-order language

Structures

69. [1, 2]

70. $\mathfrak{A} :=$ *set* A together with

   (*a*) an element $c^{\mathfrak{A}}$ of A, for each *constant* symbol c of $\mathcal{L}$,
   (*b*) a *function* $f^{\mathfrak{A}} : A^n \to A$, for each n-ary function f of $\mathcal{L}$, and
   (*c*) an n-ary *relation* $R^{\mathfrak{A}}$ on A (i.e., a subset of A^n), for each n-ary relation symbol R of $\mathcal{L}$.

71.

$$\mathfrak{A} = (A, c^{\mathfrak{A}}, f^{\mathfrak{A}}, R^{\mathfrak{A}})$$

72. $c^{\mathfrak{A}} \in A, \quad f^{\mathfrak{A}} : A^n \to A, \quad R^{\mathfrak{A}} \subseteq A^n; \quad A \neq \emptyset$

73. $\mathcal{L}$:= first-order language

74. $\mathfrak{A}$:= $\mathcal{L}$-structure

75. A := the universe of $\mathfrak{A}$

76. Note that the *variables* are not part of the definition (70).

Variable Assignment Function

77. [1,2]

78. `assignment functions`

    (*i*) begin the process of *tying together* the *symbols* with the *structures*)

    (*ii*) formalize the *interpretation* of a *term/formula* in a *structure*

79. `variable assignment function` into $\mathfrak{A}$:= function s that *assigns* to each *variable* an *element* of A,

$$s : \mathit{Vars} \to A$$

80. *Vars* := set of variable symbols (domain)

81. A := universe of $\mathfrak{A}$ (codomain)

82. $s[x|a](v) = \begin{cases} s(v), & \text{if } v \text{ is a } \textit{variable} \text{ other than } x \\ a, & \text{if } v \text{ is the } \textit{variable } x \end{cases}$

83. s := variable assignment function into $\mathfrak{A}$

84. $x :=$ variable; $\quad a \in A$

85. $s[x|a](v) :=$ x-modification of the assignment function s

86. In $s[x|a](v)$, x is *assigned* to a.

87. $\mathcal{L} :=$ first-order language

88. $\mathfrak{A} :=$ $\mathcal{L}$-structure

Term Assignment Function

89. $[1, 2]$

90. $\overline{s} :=$ `term assignment function generated by` s

    (a) $(t :=$ variable$) \rightarrow (\overline{s}(t) = s(t))$

    (b) $(t :=$ constant symbol $c) \rightarrow (\overline{s}(t) = c^{\mathfrak{A}})$

    (c) $(t := ft_1t_2...t_n) \rightarrow (\overline{s}(t) = f^{\mathfrak{A}}(\overline{s}(t_1), \overline{s}(t_2), ..., \overline{s}(t_n)))$

91. $\overline{s}$ (term) is the generalization of s (variable).

92. Note that $\overline{s}$ is defined recursively.

93. set of $\mathcal{L}$-terms := domain of $\overline{s}$

94. $A :=$ codomain of $\overline{s}$

95. $s :=$ variable assignment function into $\mathfrak{A}$

96. $c^{\mathfrak{A}} \in A$

97. $\mathcal{L} :=$ first-order language

98. $\mathfrak{A} :=$ $\mathcal{L}$-structure

Satisfaction

99. [1, 2]

100. `satisfaction` := truth

101.
$$(\mathfrak{A} \models \phi[s]) := \mathfrak{A} \texttt{ satisfies } \phi \texttt{ with assignment } s \textit{ if}$$

(i) $(\phi :\equiv = t_1 t_2) \wedge (\overline{s}(t_1)$ is the same element of A as $\overline{s}(t_2))$, or

(ii) $(\phi :\equiv R t_1 t_2 ... t_n) \wedge ((\overline{s}(t_1), \overline{s}(t_2), ..., \overline{s}(t_n)) \in R^{\mathfrak{A}})$, or

(iii) $(\phi :\equiv \neg\alpha) \wedge (\mathfrak{A} \not\models \alpha[s])$, or

(iv) $(\phi :\equiv \alpha \vee \beta) \wedge ((\mathfrak{A} \models \alpha[s]) \vee (\mathfrak{B} \models \beta[s]))$, or

(v) $(\phi :\equiv \forall x \alpha) \wedge (\forall a \in A : \mathfrak{A} \models \alpha[s(x|a)])$.

102. $\mathcal{L}$:= first-order language

103. $\mathfrak{A}$:= $\mathcal{L}$-structure

104. ϕ := $\mathcal{L}$-formula

105. $s : \mathit{Vars} \to A$

106. s := variable assignment function into $\mathfrak{A}$

107. Vars := set of variable symbols

108. A := universe of $\mathfrak{A}$

109. $(\mathfrak{A} \models \Gamma[s]) \equiv (\forall \gamma \in \Gamma : \mathfrak{A} \models \gamma[s])$

110. Γ := set of $\mathcal{L}$-formulas

True Sentences

111. [1, 2]

112.

$$(\sigma \text{ is } \texttt{true} \text{ in } \mathfrak{A}) \leftrightarrow (\mathfrak{A} \models \sigma[s])$$

113. $\sigma :=$ sentence

114. $\mathfrak{A} :=$ structure

115. $s :=$ variable assignment function into $\mathfrak{A}$

116. Note that the *definition* of **satisfaction** is *relative* to an *assignment function.*

On the equality of term assignment functions

117. [1, 2]

118.

$$(\forall v \in t : s_1(v) = s_2(v)) \to (\overline{s_1}(t) = \overline{s_2}(t))$$

119. $\mathfrak{A} :=$ structure

120. $v :=$ variable

121. $s_1, s_2 :=$ variable assignment functions into $\mathfrak{A}$

122. $t :=$ term

Satisfaction of a formula with different variable assignment functions

123. [1, 2]

124.
$$(\forall v \in \phi : s_1(v) = s_2(v)) \rightarrow (\mathfrak{A} \models \phi[s_1] \leftrightarrow \mathfrak{A} \models \phi[s_2])$$

125. $\mathfrak{A} :=$ structure

126. $v :=$ *free* variable

127. $s_1, s_2 :=$ variable assignment functions into $\mathfrak{A}$

128. $\phi :=$ formula

Satisfaction for all variable assignment functions

129. [1, 2]

130.
$$(\forall s : \mathfrak{A} \models \sigma[s]) \ \veebar \ (\mathfrak{A} \models \sigma[s] \text{ for no } s)$$

131. $\mathcal{L} :=$ first-order language

132. $\mathfrak{A} := \mathcal{L}$-structure

133. $\sigma :=$ sentence in $\mathcal{L}$

134. $s :=$ variable assignment function into $\mathfrak{A}$

Model (`formula`)

135. [1, 2]

136. $(\mathfrak{A} \models \phi) := \mathfrak{A}$ is a `model of` ϕ

137.
$$\mathfrak{A} \models \phi \ \leftrightarrow \ \forall s : \mathfrak{A} \models \phi[s]$$

138.

$$\mathfrak{A} \models \Phi \ \leftrightarrow\ \forall \phi \in \Phi : \mathfrak{A} \models \phi$$

139. $\phi :=$ formula in $\mathcal{L}$

140. $s :=$ variable assignment function into $\mathfrak{A}$

141. $\Phi :=$ set of $\mathcal{L}$-formulas

142. $\mathcal{L} :=$ first-order language

143. $\mathfrak{A} := \mathcal{L}$-structure

True Sentences

144. [1, 2]

145.

$$\mathfrak{A} \models \sigma \ \leftrightarrow\ \forall s : \mathfrak{A} \models \sigma[s]$$

146. $(\mathfrak{A} \models \sigma) := \mathfrak{A}$ is a `model of` σ

147. $\sigma :=$ sentence in $\mathcal{L}$

148. σ is *true* in $\mathfrak{A}$

149. $s :=$ variable assignment function into $\mathfrak{A}$

150. $\Phi :=$ set of $\mathcal{L}$-formulas

151. $\mathcal{L} :=$ first-order language

152. $\mathfrak{A} := \mathcal{L}$-structure

Satisfaction of formulas with the connective "and"

153. [1, 2]

154.
$$\mathfrak{A} \models (\alpha \wedge \beta)[s] \leftrightarrow \mathfrak{A} \models \alpha[s] \wedge \mathfrak{A} \models \beta[s]$$

155. $(\alpha \wedge \beta) \equiv (\neg((\neg\alpha) \vee (\neg\beta)))$

156. (155) is an abbreviation.

157. $s :=$ variable assignment function into $\mathfrak{A}$

158. $\alpha[s], \beta[s] := \mathcal{L}$-formulas with assignment function s

159. $\mathcal{L} :=$ first-order language

160. $\mathfrak{A} := \mathcal{L}$-structure

Satisfaction with the existential quantifier

161. [1, 2]

162.
$$\mathfrak{A} \models (\exists x)(\alpha)[s] \leftrightarrow \exists a \in A : \mathfrak{A} \models \alpha[s[x|a]]$$

163. $\mathfrak{A} :=$ structure

164. $A :=$ universe of $\mathfrak{A}$; $\quad a \in A$

165. $x :=$ variable

166. $s[x|a](v) := x$-modification of the assignment function s

167. $\alpha :=$ formula with the x-modification of the assignment function s

Substitution into a Term

168. [1, 2]

169.

$$u_t^x \texttt{ (}u\texttt{ with }x\texttt{ replaced by }t\texttt{)} \textit{ if}$$

(i) (u is a variable not equal to x) $\to$ (u_t^x is u)

(ii) (u is x) $\to$ (u_t^x is t)

(iii) (u is a constant symbol) $\to$ (u_t^x is u)

(iv) $(u :\equiv fu_1u_2...u_n) \to (u_t^x$ is $f(u_1)_t^x(u_2)_t^x...(u_n)_t^x)$

170. $u, t, u_t^x, u_i :=$ terms

171. $x :=$ variable

172. $f := n$-ary function

173. Note that in (169.iv), the parentheses have been added for the purpose of readability; so, $(u_1)_t^x :\equiv u_1{}_t^x$.

174. `Substitution into a term` (169) is a definition by *recursion*.

Substitution into a Formula

175. [1, 2]

176.

$$\phi_t^x \texttt{ (}\phi\texttt{ with }x\texttt{ replaced by }t\texttt{)} \textit{ if}$$

(i) $(\phi :\equiv = u_1u_2) \to (\phi_t^x$ is $= (u_1)_t^x(u_2)_t^x)$

(ii) $(\phi :\equiv Ru_1u_2...u_n) \to (\phi_t^x$ is $R(u_1)_t^x(u_2)_t^x...(u_n)_t^x)$

(iii) $(\phi :\equiv \neg(\alpha)) \to (\phi_t^x$ is $\neg(\alpha_t^x))$

(iv) $(\phi :\equiv (\alpha \vee \beta)) \to (\phi_t^x$ is $(\alpha_t^x \vee \beta_t^x))$

$$(v)\ \ \phi :\equiv (\forall y)(\alpha) \ \rightarrow \ \phi^x_t = \begin{cases} \phi, & \text{if } x \text{ is } y \\ (\forall y)(\alpha^x_t), & \text{otherwise} \end{cases}$$

177. $\mathcal{L}$:= first-order language

178. ϕ, ϕ^x_t := $\mathcal{L}$-formulas

179. t := term

180. x := variable

181. R := n-ary relation

182. Note that in (176), the parentheses have been added for the purpose of readability; so, $(\phi_1)^x_t :\equiv {\phi_1}^x_t$.

183. `Substitution into a formula` (176) is a definition by *recursion.*

A term substitutable for a variable in a formula

184. [1,2]

185.

t `is substitutable for` x `in` ϕ *if*

(i) ϕ is atomic, or

(ii) $\phi :\equiv \neg(\alpha)$ and t is substitutable for x in α, or

(iii) $\phi :\equiv (\alpha \vee \beta)$ and t is substitutable for x in both α and β, or

(iv) $\phi :\equiv (\forall y)(\alpha)$ and either

(a) x is not free in ϕ, or

(b) y does not occur in t and t is substitutable for x in α.

186. $\mathcal{L}$:= first-order language

187. $\phi, \alpha, \beta :=$ $\mathcal{L}$-formulas

188. $t :=$ term

189. $x :=$ variable

190. Notice that

   (i) certain operations are allowed only if t is substitutable for x in ϕ;

   (ii) this restriction is important to preserve the truth of formulas after performing substitutions.

Logical Implication `(sets of formulas)`

191. [1, 2]

192.

$$(\forall \mathfrak{A} : \mathfrak{A} \models \Delta \rightarrow \mathfrak{A} \models \Gamma) \rightarrow (\Delta \models \Gamma)$$

193. $(\Delta \models \Gamma) := \Delta$ `logically implies` Γ

194. $\mathcal{L} :=$ first-order language

195. $\mathfrak{A} :=$ $\mathcal{L}$-structure

196. $\Delta, \Gamma :=$ sets of $\mathcal{L}$-formulas

197. (192) says that if Δ is true in $\mathfrak{A}$, then Γ is true in $\mathfrak{A}$.

198. Recall that Δ is true in $\mathfrak{A}$ if $\forall s : \mathfrak{A} \models \Delta[s]$.

199. $s :=$ variable assignment function into $\mathfrak{A}$

Valid Formulas

200. [1, 2]

201.

$$(\models \phi) \rightarrow (\phi \text{ is } \texttt{valid})$$

202. $(\emptyset \models \phi) :\equiv (\models \phi) := (\forall s : \phi \text{ is true})$

203. $\mathcal{L} :=$ first-order language

204. $\phi := \mathcal{L}$-formula

205. $s :=$ variable assignment function

206. Notice that

  (i) $\mathfrak{A} \models \sigma$ means *truth* (if there is a structure on the left), whereas

  (ii) $\Gamma \models \sigma$ means *logical implication* (if there is a set of sentences on the left).

207. $\mathfrak{A} := \mathcal{L}$-structure

208. $\Gamma :=$ set of sentences in $\mathcal{L}$

209. $\sigma :=$ sentence

Universal Closure of a Formula

210. [1, 2]

211.

$$\models \phi \;\leftrightarrow\; \models (\forall x)(\phi)$$

212.

$$(\phi \text{ has free variables } x, y, z) \;\rightarrow\; (\models \phi \;\leftrightarrow\; \models \forall x \forall y \forall z \phi)$$

213. $\forall x \forall y \forall z \phi :=$ sentence called `universal closure of` ϕ

214. $\mathcal{L}$:= first-order language

215. $\phi := \mathcal{L}$-formula

216. $x, y, z :=$ variables

On the validity of a conditional statement of formulas

217. [1,2]

218. $\models (\phi \to \psi) \ \to \ \phi \models \psi$

219. $\phi, \psi :=$ formulas

"Bottom-up" Deduction

220. [1,2]

221.

$$(D :\equiv \Sigma \vdash \phi) \ \text{ if } \ \forall i : 1 \leq i \leq n, \text{ either}$$

(i) $\phi_i \in \Lambda$, or

(ii) $\phi_i \in \Sigma$, or

(iii) $\exists(\Gamma, \phi_i) : \Gamma \subseteq \{\phi_1, \phi_2, ..., \phi_{i-1}\}$.

222. $D :\equiv (\Sigma \vdash \phi) :=$ `deduction from` Σ `of` ϕ

223. $\mathcal{L}$:= first-order language

224. $\phi, \phi_i := \mathcal{L}$-formulas

225. $\Lambda :=$ set of $\mathcal{L}$-formulas (`logical axioms`)

226. $\Sigma :=$ collection of $\mathcal{L}$-formulas (`nonlogical axioms`)

227. $(\Gamma, \phi_i) :=$ rule of inference

228. $D :=$ finite sequence $(\phi_1, \phi_2, ..., \phi_n)$ of $\mathcal{L}$-formulas

229. `bottom-up` := it defines a *deduction* in terms of its parts

"Top-down" Deduction

230. [1,2]

231.

$$\text{Thm}_\Sigma = \{\phi \mid \Sigma \vdash \phi\} \text{ is the } \texttt{smallest} \textit{ set } C \text{ such that}$$

(*i*) $\Sigma \subseteq C$

(*ii*) $\Lambda \subseteq C$

(*iii*) $((\Gamma, \theta) := \text{rule of inference } \wedge \ \Gamma \subseteq C) \rightarrow (\theta \in C)$

232. $\mathcal{L} :=$ first-order language

233. $\Sigma, \Lambda :=$ sets of $\mathcal{L}$-formulas

234. `top-down` := we can think of the collection of deductions from Σ (called Thm_Σ) as the *closure of axioms* under the application of the *rules of inference.*

Decidable Set of Axioms

235. [1,2]

236. `decidable` *set of axioms* := (we will be able to decide whether)

$$\phi \in \Lambda \ \veebar \ \phi \notin \Lambda$$

237. $\mathcal{L} :=$ first-order language

238. $\Lambda :=$ collection of *logical axioms* for $\mathcal{L}$

(Non)Logical Axioms

239. $[1, 2]$

240.
$$\Lambda \cup \Sigma := \text{expanded set of axioms}$$

241. $\mathcal{L}$:= first-order language

242. Λ := collection of `logical axioms` for $\mathcal{L}$

243. Σ := collection of `nonlogical axioms` for $\mathcal{L}$

244. Λ is fixed

245. The rules of inference are fixed.

246. Σ *must be specified for each deduction.*

247. The collection Λ of *logical axioms* is *decidable.*

248. `nonlogical axioms` := *additional axioms*, beyond the set of logical axioms

249. formula := (axiom) $\veebar$ (arise from previous formulas in the deduction via a rule of inference)

Equality Axioms

250. $[1, 2]$

251. `(E1)`
$$x = x \text{ for each variable } x$$

252. `(E2)`
$$[(x_1 = y_1) \wedge (x_2 = y_2) \wedge ... \wedge (x_n = y_n)] \rightarrow$$
$$\rightarrow (f(x_1, x_2, ..., x_n) = f(y_1, y_2, ..., y_n))$$

253. `(E3)`

$$[(x_1 = y_1) \wedge (x_2 = y_2) \wedge ... \wedge (x_n = y_n)] \rightarrow$$
$$\rightarrow (R(x_1, x_2, ..., x_n) = R(y_1, y_2, ..., y_n))$$

Quantifier Axioms

254. $[1, 2]$

255. `(Q1)`: Universal instantiation

$$(\forall x \phi) \rightarrow \phi^x_t, \quad \text{if } t \text{ is substitutable for } x \text{ in } \phi$$

256. `(Q2)`: Existential generalization

$$\phi^x_t \rightarrow (\exists x \phi), \quad \text{if } t \text{ is substitutable for } x \text{ in } \phi$$

Rules of Inference

257. $[1, 2]$

258. There are two types of `rules of inference`: *propositional consequence* and one dealing with *quantifiers*.

259. The set of *rules of inference* is *decidable*.

Propositional Consequence: Definition

260. $[1, 2]$

261. *If every truth assignment that makes each propositional formula in* Γ_P *true also makes* ϕ_P *true, then* ϕ_P *is a* `propositional consequence` of Γ_P.

262. $\Gamma_P :=$ set of propositional formulas

263. ϕ_P := propositional formula

264. Note that

$$(\phi_P := \text{tautology}) \leftrightarrow (\phi_P \text{ is a propositional consequence of } \emptyset).$$

Propositional Consequence: Tautology

265. [1, 2]

266.

$$(\phi_P \text{ is a propositional consequence of } \Gamma_P) \leftrightarrow$$
$$\leftrightarrow ([\gamma_{1P} \wedge \gamma_{2P} \wedge ... \wedge \gamma_{nP}] \rightarrow \phi_P) \text{ is a tautology}$$

267. $\Gamma_P = \{\gamma_{1P}, \gamma_{2P}, ..., \gamma_{nP}\}$:= nonempty finite set of propositional formulas

268. ϕ_P := propositional formula

Propositional Consequence: Extension to First-order Logic

269. [1, 2]

270.

$$(\phi_P \text{ is a propositional consequence of } \Gamma_P) \rightarrow$$
$$\rightarrow (\phi \text{ is a propositional consequence of } \Gamma)$$

271. $\mathcal{L}$:= first-order language

272. Γ := finite set of $\mathcal{L}$-formulas

273. ϕ := $\mathcal{L}$-formula

Rule of Inference of type (PC)

274. [1,2]

275.

$$\phi \text{ is a } \textit{propositional consequence} \text{ of } \Gamma \rightarrow$$
$$\rightarrow (\Gamma, \phi) \text{ is a } \texttt{rule of inference of type (PC)}$$

276. $\mathcal{L}$:= first-order language

277. Γ := finite set of $\mathcal{L}$-formulas

278. ϕ := $\mathcal{L}$-formula

Rules of Inference of type (QR)

279. [1,2]

280. `Rules of inference of type (QR)`

$$(i)\ \ (\{\psi \rightarrow \phi\}, (\forall x \phi))$$
$$(ii)\ \ (\{\phi \rightarrow \psi\}, (\exists x \phi) \rightarrow \psi)$$

281. x := variable (not free in ψ)

282. ψ, ϕ := formulas

283. (280) means *if x is not free in ψ:*

(i) from $\phi \rightarrow \psi$, it may be deduced $\psi \rightarrow (\forall x \phi)$;

(ii) from $\psi \rightarrow \phi$, it may be deduced $(\exists x \phi) \rightarrow \psi$.

On the validity and tautology of formulas

284. [1,2]

285. (θ is not `valid`) $\rightarrow$ (θ_P is not a `tautology`)

286. (θ_P is `tautology`) $\rightarrow$ (θ is a `valid`)

287. $\theta :=$ formula in *first-order* logic

288. $\theta_P :=$ formula in *propositional* logic

List of requirements for axioms and rules of inference

289. [1,2]

290. The following list is *required* for our **axioms** and **rules of inference**:

    (*i*) There will be an *algorithm* that will *decide*, given a *formula* θ, whether or not θ is a `logical axiom`.

    (*ii*) There will be an *algorithm* that will *decide*, given a *finite set of formulas* Γ and a *formula* θ, whether or not (Γ, θ) is a `rule of inference`.

    (*iii*) For each *rule of inference* (Γ, θ), Γ will be a *finite set of formulas.*

    (*iv*) Each *logical axiom* will be *valid.*

    (*v*) Our *rules of inference* will *preserve truth.* In other words, for each rule of inference (Γ, θ), $\Gamma \models \theta$.

291. The requirements in (290) provide the basis of the `Soundness Theorem`.

Logical Axioms: Valid

292. [1,2]

293. Theorem: `The logical axioms are valid.`

Rule of Inference: Theorem

294. [1,2]

295. Theorem:

$$(\Gamma, \theta) := \texttt{rule of inference} \;\rightarrow\; \Gamma \models \theta$$

Soundness Theorem

296. [1,2]

297.

$$\Sigma \vdash \phi \;\rightarrow\; \Sigma \models \phi$$

298. $\mathcal{L}$:= first-order language

299. Σ := set of $\mathcal{L}$-formulas

300. In words, the `Soundness Theorem` (297) tells us that *in any structure $\mathfrak{A}$ that makes all of the formulas of Σ true, ϕ is true as well.*

301. *If there is a deduction from Σ of ϕ, then Σ logically implies ϕ.*

302. The purely *syntactic notion of deduction* is *linked* to the notions of `truth` and `logical implication`.

303. The Soundness Theorem is explicitly trying to relate the *syntactical notion* of `deducibility` ($\vdash$) with the *semantical notion* of `logical implication` ($\models$).

304. *If there is a deduction of ϕ from Σ, then ϕ is true in any model of Σ.*

When a variable is not free in a formula

305. $[1, 2]$

306.

$$x \text{ is not free in } \psi \;\rightarrow\; (\phi \rightarrow \psi) \models [(\exists x \phi) \rightarrow \psi]$$

307. $x :=$ variable

308. $\psi, \phi :=$ formulas

Variable Assignment Functions and Substitutions

309. $[1, 2]$

310.

$$s' = s[x|\overline{s}(t)] \;\rightarrow\; \overline{s}(u^x_t) = \overline{s'}(u)$$

311. $u, t :=$ terms

312. $x :=$ variable

313. $s : \mathit{Vars} \rightarrow A$

314. $s :=$ variable assignment function

315. $s[x|\overline{s}(t)] := x$-modification of the assignment function s

316. $u^x_t := u$ with x replaced by t

Term substitution in the x-modification of the assignment function

317. $[1, 2]$

318.

$$\mathfrak{A} \models \phi_t^x[s] \;\leftrightarrow\; \mathfrak{A} \models \phi[s']$$

319. $\mathcal{L}$:= first-order language

320. ϕ := formula

321. x := variable

322. t := term substitutable for x in ϕ

323. $s : Vars \to A$

324. s := variable assignment function

325. $s' = s[x|\overline{s}(t)]$

326. $s[x|\overline{s}(t)]$:= x-modification of the assignment function s

Equality: Equivalence Relation

327. $[1, 2]$

328. `Equality is an equivalence relation`

$$\begin{aligned} (i) &\vdash x = x \\ (ii) &\vdash x = y \to y = x \\ (iii) &\vdash (x = y \land y = z) \to x = z \end{aligned}$$

A set of formulas proves a formula if and only if it proves the formula for all variables

329. $[1, 2]$

330.

$$\Sigma \vdash \theta \;\leftrightarrow\; \Sigma \vdash \forall x \theta$$

331. *For a formula to be true in a structure, it must be satisfied in that structure with every assignment function.*

Adding/deleting a universal quantifier

332. [1, 2]

333.

$$\Sigma \vdash \theta \;\rightarrow\; (\Sigma' \text{ is formed by taking any } \sigma \in \Sigma \text{ and } \textit{adding} \text{ or } \textit{deleting} \text{ a } \texttt{universal quantifier} \text{ whose scope is the entire formula } \;\rightarrow\; \Sigma' \vdash \theta)$$

334. *If we know $\Sigma \vdash \theta$, we can assume that every element of Σ is a sentence: By quoting (333) several times, we can replace each $\sigma \in \Sigma$ with its universal closure.*

The Deduction Theorem

335. [1, 2]

336.

$$(\Sigma \cup \theta \vdash \phi) \;\leftrightarrow\; (\Sigma \vdash (\theta \rightarrow \phi))$$

337. $\theta :=$ sentence

338. $\Sigma :=$ set of formulas

339. The `Deduction Theorem` (336) says that *there is a deduction of ϕ from the assumption θ if and only if there is a deduction of the implication $\theta \rightarrow \phi$.*

340. In (336), we omit the braces of $\Sigma \cup \{\theta\} \vdash \phi$.

341. `deduction` := *formal equivalents* of the *mathematical proofs*

Proofs by Contradiction

342. $[1, 2]$

343.

$$(\Sigma \vdash \eta) \;\leftrightarrow\; (\Sigma \cup (\neg\eta) \vdash [(\forall x)\, x = x] \wedge \neg[(\forall x)\, x = x])$$

344. $\eta :=$ sentence

Unary Relation Symbol

345. $[1, 2]$

346.

$$\vdash [(\forall x)P(x)] \rightarrow [(\exists x)P(x)]$$

347. $P :=$ unary relation symbol

Binary Relation Symbol

348. $[1, 2]$

349.

$$(\forall x)(\forall y)P(x, y) \vdash (\forall y)(\forall z)P(z, y)$$

350. $P :=$ binary relation symbol

Two unary relation symbols

351. $[1, 2]$

352.

$$\vdash [(\forall x)(P(x)) \wedge (\forall x)(Q(x))] \rightarrow (\forall x)[P(x) \wedge Q(x)]$$

353. $P, Q :=$ unary relation symbols

Complete Deductive System

354. [1, 2]

355.

$$\forall\Sigma\ \forall\phi\ (\Sigma \models \phi \to \Sigma \vdash \phi) \ \to\ (\Lambda, \Gamma_\theta) := \texttt{complete}$$

356. $\Lambda :=$ collection of logical axioms

357. $\Gamma_\theta :=$ collection of rules of inference

358. $\Sigma :=$ set of nonlogical axioms

359. $\mathcal{L} :=$ first-order language

360. $\phi := \mathcal{L}$-formula

361. If ϕ is an $\mathcal{L}$-formula that is true in *every* model of Σ, then there will be a deduction from Σ to ϕ.

362. Our ability to prove ϕ depends on ϕ being true in every model of Σ.

(In)Consistent

363. [1, 2]

364.

$$\exists(\Sigma \vdash [(\forall x)\, x = x] \land \neg[(\forall x)\, x = x]) \ \to\ \Sigma \text{ is } \texttt{inconsistent}$$

365.

$$\Sigma \text{ is } \underline{\text{not}} \text{ inconsistent} \ \to\ \Sigma \text{ is } \texttt{consistent}$$

366. $\mathcal{L} :=$ first-order language

367. $\Sigma :=$ set of $\mathcal{L}$-formulas

$$\Sigma \text{ proves a } \textit{contradiction} \ \to\ \Sigma \text{ is } \textbf{inconsistent}$$

368.

$$\Sigma \text{ is } \textbf{inconsistent} \;\rightarrow\; \exists(\Sigma \vdash \phi)$$

369. $\phi := \mathcal{L}$-formula

370. $\phi := [(\forall x)\, x = x] \wedge \neg[(\forall x)\, x = x]$

371. ϕ is a *contradictory sentence* ($\perp$).

372. $\perp$ is a sentence that is *false in every language* and *is true in no structure.*

Completeness Theorem

373. [1, 2]

374.

$$(\Sigma \models \phi) \;\rightarrow\; (\Sigma \vdash \phi)$$

375. $\mathcal{L}$:= first-order language

376. Σ := set of $\mathcal{L}$-formulas

377. $\phi := \mathcal{L}$-formula

378. *The Completeness Theorem finishes the link between* ***deducibility*** *and* ***logical implication****.*

Soundness + Completeness

379. [1, 2]

380.

$$(\Sigma \models \phi) \;\leftrightarrow\; (\Sigma \vdash \phi)$$

381. $\mathcal{L}$:= first-order language

382. $\Sigma :=$ set of $\mathcal{L}$-formulas

383. $\phi := \mathcal{L}$-formula

Compactness Theorem

384. [1, 2]

385.

$$(\exists \mathfrak{A} : \mathfrak{A} \models \Sigma) \leftrightarrow (\forall \Sigma_0 \; \exists \mathfrak{B} : \mathfrak{B} \models \Sigma_0)$$

386. $\Sigma :=$ set of axioms

387. $(\mathfrak{A} \models \Sigma) := \mathfrak{A}$ is a model of Σ

388. $\Sigma_0 \subseteq \Sigma$

389. $\Sigma_0 :=$ finite subset of Σ

390. $\mathfrak{B} :=$ model of Σ_0

391. The `Compactness Theorem`

   (*i*) is one use of the link between *deducibility* and *logical implication*;

   (*ii*) focus our attention on the *finiteness* of *deductions*;

   (*iii*) says that

$$\Sigma \text{ is satisfiable } \leftrightarrow \; \Sigma \text{ is finitely satisfiable.}$$

(Finitely) Satisfiable

392. [1, 2]

393.

$$(\exists \mathfrak{A} : \mathfrak{A} \models \Sigma) \rightarrow (\Sigma \text{ is } \texttt{satisfiable})$$

394.

$$(\forall\Sigma_0\ \exists\mathfrak{B} : \mathfrak{B} \models \Sigma_0) \to (\Sigma \text{ is } \texttt{finitely satisfiable})$$

395. $\Sigma :=$ set of axioms

396. $(\mathfrak{A} \models \Sigma) := \mathfrak{A}$ is a model of Σ

397. $\Sigma_0 \subseteq \Sigma$

398. $\Sigma_0 :=$ finite subset of Σ

399. $\mathfrak{B} :=$ model of Σ_0

Finite subset of a set of formulas

400. [1, 2]

401.

$$(\Sigma \models \theta) \ \leftrightarrow \ (\exists\Sigma_0 \subseteq \Sigma : \Sigma_0 \models \theta)$$

402. $\mathcal{L} :=$ first-order language

403. $\Sigma :=$ set of $\mathcal{L}$-formulas

404. $\theta := \mathcal{L}$-formula

405. $\Sigma_0 :=$ finite subset of Σ

First-order Sentences: Natural Numbers

406. [1, 2]

407. *No set of first-order sentences can completely characterize the structure of the natural numbers.*

Theory of a Structure

408. [1, 2]

409.

$$Th(\mathfrak{A}) = \{\phi \mid \mathfrak{A} \models \phi\}$$

410.

$$Th(\mathfrak{A}) = Th(\mathfrak{B}) \ \rightarrow \ \mathfrak{A} \equiv \mathfrak{B}$$

411.

$$(\mathfrak{A} \equiv \mathfrak{N}) \ \rightarrow \ (\mathfrak{A} \text{ is a } \texttt{model of arithmetic})$$

412. $\mathcal{L}$:= first-order language

413. $\mathfrak{A}, \mathfrak{B}$:= $\mathcal{L}$-structures

414. ϕ := $\mathcal{L}$-formula

415. $(\mathfrak{A} \equiv \mathfrak{B})$:= $\mathfrak{A}$ and $\mathfrak{B}$ are `elementarily equivalent`

416. $\mathcal{L}_{NT} = \{0, S, +, \cdot, E, <\}$

417. $\mathcal{L}_{NT}$:= language of number theory

418. $\mathfrak{N}$:= $\mathcal{L}_{NT}$-structure

Substructure

419. [1, 2]

420. $\mathfrak{A} \subseteq \mathfrak{B}$ if

(i) $A \subseteq B$

(ii) $\forall c : c^{\mathfrak{A}} = c^{\mathfrak{B}}$

(iii) $\forall R : R^{\mathfrak{A}} = R^{\mathfrak{B}} \cap A^n$

(iv) $\forall f : f^{\mathfrak{A}} = f^{\mathfrak{B}} \restriction_{A^n}$

421. (420.iv) means

$$(\forall f)\ (\forall a \in A) : f^{\mathfrak{A}}(a) = f^{\mathfrak{B}}(a).$$

422. $\mathcal{L}$:= first-order language

423. $\mathfrak{A}, \mathfrak{B}$:= $\mathcal{L}$-structures

424. $(\mathfrak{A} \subseteq \mathfrak{B})$:= $\mathfrak{A}$ is a `substructure` of $\mathfrak{B}$

425. A := universe of $\mathfrak{A}$

426. B := universe of $\mathfrak{B}$

427. R := n-ary relation symbol

428. f := n-ary function symbol

429. $f^{\mathfrak{B}} \restriction_{A^n}$:= restriction of the function $f^{\mathfrak{B}}$ to the set A^n

430. A substructure of $\mathfrak{B}$ is completely determined by its universe, and this universe can be any nonempty subset of B that contains the constants and is closed under every function f.

Elementary Substructure/Extension

431. [1, 2]

432.

$(\mathfrak{A} \prec \mathfrak{B})$:= $\mathfrak{A}$ is an `elementary substructure` of $\mathfrak{B}$
(equivalently, $\mathfrak{B}$ is an `elementary extension` of $\mathfrak{A}$) *if*

$$\forall s \forall \phi : \ \mathfrak{A} \models \phi[s] \ \leftrightarrow \ \mathfrak{B} \models \phi[s]$$

433. $\mathcal{L}$:= first-order language

434. $\mathfrak{A}, \mathfrak{B}$:= $\mathcal{L}$-structures

435. $\mathfrak{A} \subseteq \mathfrak{B}$

436. $\phi :=$ $\mathcal{L}$-formula

437. $s : \mathit{Vars} \to A$

438. $\mathit{Vars} :=$ set of variables

439. $A :=$ universe of $\mathfrak{A}$

Truth in elementary substructure/extension

440. [1, 2]

441.

$$(\mathfrak{A} \prec \mathfrak{B}) \;\to\; (\sigma \text{ is } \mathit{true} \text{ in } \mathfrak{A} \leftrightarrow \sigma \text{ is } \mathit{true} \text{ in } \mathfrak{B})$$

442. $\mathfrak{A}, \mathfrak{B} :=$ structures

443. $\sigma :=$ sentence

Condition for an elementary substructure

444. [1, 2]

445.

$$(\mathfrak{A} \subseteq \mathfrak{B}) \;\wedge\; (\forall \alpha \, \forall s : \mathfrak{B} \models \exists x \alpha[s], \; \exists a : \mathfrak{B} \models \alpha[s[x|a]]) \;\to\; (\mathfrak{A} \prec \mathfrak{B})$$

446. $\mathfrak{A}, \mathfrak{B} :=$ structures

447. $\mathfrak{A} \subseteq \mathfrak{B}$

448. $\alpha :=$ formula

449. $s : \mathit{Vars} \to A$

450. $A :=$ universe of $\mathfrak{A}$

Hilbert Axiomatic System

451. $[4, 5]$

452. *Hilbert-style calculus* is performed in the `Hilbert Axiomatic System`, composed by 9 axioms and 1 rule (*Modus Ponens*).

453. `rule` := inference rule of logic

Axioms of the Hilbert-style Calculus

454. $[4, 5]$

455. A, B, C := propositional *variables* or *formulas*

456. $\vdash A \to (B \to A)$

457. $\vdash (A \to (B \to C)) \to (A \to B) \to (A \to C)$

458. $\vdash (\neg A \to \neg B) \to B \to A$

459. $\vdash A \to (A \vee B)$

460. $\vdash A \to (B \vee A)$

461. $\vdash (A \to B) \to ((C \to B) \to (A \vee C \to B))$

462. $\vdash (A \wedge B) \to A$

463. $\vdash (A \wedge B) \to B$

464. $\vdash A \to (B \to (A \wedge B))$

Inference Rule of the Hilbert-style Calculus

465. $[4, 5]$

466. `Modus Ponens`

$$\begin{array}{l} \vdash P \\ \vdash P \to Q \\ \hline \vdash Q \end{array}$$

Sequent Systems: Classical Logic

467. [3,6]

468. `LK` := sequent system for classical logic

469. `sequents` := basic syntactic units (finite sequence of formulas)

470. α_i, β_i := formulas

471.

$$\alpha_1, ..., \alpha_m \Rightarrow \beta_1, ..., \beta_n$$

472. $m, n \geq 0$

473. (471) is a *sequent*.

474. $\Rightarrow$ is a *sequent arrow*.

475. $\alpha_1, ..., \alpha_m$:= *antecedents* (conjunctive-like "assumptions")

476. $\beta_1, ..., \beta_n$:= *succedents* (disjunctive-like "conclusions")

477. (471) means that $(\alpha_1 \wedge ... \wedge \alpha_m)$ implies $(\beta_1 \vee ... \vee \beta_n)$.

478.

$$\alpha_1, ..., \alpha_m \Rightarrow$$

means $(\alpha_1 \wedge ... \wedge \alpha_m)$ leads to a `contradiction`.

479.

$$\Rightarrow \beta_1, ..., \beta_n$$

means $(\beta_1 \vee ... \vee \beta_n)$ follows from no assumption.

480. The `provability` of a *sequent* is a *syntactical* approach.

481. The `validity` of a *sequent* is a *semantical* approach.

482. A sequent system contains *initial sequents* (axiom schemes in Hilbert-style systems) and *rules*.

483. `rule` := one/two *upper sequents* and one *lower sequent*

484. The lower sequent can be inferred from the upper sequents.

485.
$$\frac{\text{upper sequents}}{\text{lower sequent}}$$

486. $\Gamma, \Pi, \Delta, \ldots$ (capital Greek letters) := finite (possibly empty) **sequences** of formulas

487. LK has three kinds of rules:

(*i*) (left/right) rules for $\vee, \wedge, \to, \neg$,

(*ii*) cut rule,

(*iii*) (left/right) structural rules.

488. The initial sequents are of the form $\alpha \Rightarrow \alpha$.

489. `Rules for the logical connectives:`

490.

$$\frac{\alpha, \Gamma \Rightarrow \Pi \qquad \beta, \Gamma \Rightarrow \Pi}{\alpha \vee \beta, \Gamma \Rightarrow \Pi} \ (\vee\mathrm{L})$$

491.

$$\frac{\Gamma \Rightarrow \Lambda, \alpha}{\Gamma \Rightarrow \Lambda, \alpha \vee \beta} \ (\vee\mathrm{R1}) \qquad \frac{\Gamma \Rightarrow \Lambda, \beta}{\Gamma \Rightarrow \Lambda, \alpha \vee \beta} \ (\vee\mathrm{R2})$$

492.

$$\frac{\alpha, \Gamma \Rightarrow \Pi}{\alpha \wedge \beta, \Gamma \Rightarrow \Pi} \ (\wedge\mathrm{L1}) \qquad \frac{\beta, \Gamma \Rightarrow \Pi}{\alpha \wedge \beta, \Gamma \Rightarrow \Pi} \ (\wedge\mathrm{L2})$$

493.

$$\frac{\Gamma \Rightarrow \Lambda, \alpha \quad \Gamma \Rightarrow \Lambda, \beta}{\Gamma \Rightarrow \Lambda, \alpha \wedge \beta} \ (\wedge\mathrm{R})$$

494.

$$\frac{\Gamma \Rightarrow \Lambda, \alpha \quad \beta, \Delta \Rightarrow \Pi}{\alpha \to \beta, \Gamma, \Delta \Rightarrow \Lambda, \Pi} \ (\to\mathrm{L}) \qquad \frac{\alpha, \Gamma \Rightarrow \Lambda, \beta}{\Gamma \Rightarrow \Lambda, \alpha \to \beta} \ (\to\mathrm{R})$$

495.

$$\frac{\Gamma \Rightarrow \Lambda, \alpha}{\neg\alpha, \Gamma \Rightarrow \Lambda} \ (\neg\mathrm{L}) \qquad \frac{\alpha, \Gamma \Rightarrow \Lambda}{\Gamma \Rightarrow \Lambda, \neg\alpha} \ (\neg\mathrm{R})$$

496. `Cut rule:`

$$\frac{\Gamma \Rightarrow \Lambda, \alpha \quad \alpha, \Delta \Rightarrow \Pi}{\Gamma, \Delta \Rightarrow \Lambda, \Pi} \ (\mathrm{cut})$$

497. `Structural rules`:

(*i*) *exchange rules*

$$\frac{\Gamma, \alpha, \beta, \Delta \Rightarrow \Pi}{\Gamma, \beta, \alpha, \Delta \Rightarrow \Pi} \text{ (eL)} \qquad \frac{\Gamma \Rightarrow \Pi, \alpha, \beta, \Lambda}{\Gamma \Rightarrow \Pi, \beta, \alpha, \Lambda} \text{ (eR)}$$

(*ii*) *contraction rules*

$$\frac{\alpha, \alpha, \Gamma \Rightarrow \Pi}{\alpha, \Gamma \Rightarrow \Pi} \text{ (cont L)} \qquad \frac{\Gamma \Rightarrow \Pi, \alpha, \alpha}{\Gamma \Rightarrow \Pi, \alpha} \text{ (cont R)}$$

(*iii*) *weakening rules*

$$\frac{\Gamma \Rightarrow \Pi}{\alpha, \Gamma \Rightarrow \Pi} \text{ (wL)} \qquad \frac{\Gamma \Rightarrow \Pi}{\Gamma \Rightarrow \Pi, \alpha} \text{ (wR)}$$

498. The parenthesis are labels for the rules.

499. Note that

$$\frac{\Gamma \Rightarrow \Lambda, \alpha \quad \beta, \Delta \Rightarrow \Pi}{\alpha \to \beta, \Gamma, \Delta \Rightarrow \Lambda, \Pi} \text{ ($\to$L)}$$

in the special case where

$$\Gamma = \alpha, \quad \Lambda = \Delta = \emptyset, \quad \Pi = \beta,$$

the succedent is the *Modus Ponens* for the sequent arrow,

$$\frac{\alpha \Rightarrow \alpha \quad \beta \Rightarrow \beta}{\alpha \to \beta, \alpha \Rightarrow \beta} \text{ ($\to$L).}$$

500. `active formulas` := formulas in the rules

501. `cut formula` := active formula of the cut rule

502. `principal formula` := formulas in lower sequents of the rules

503. `side formulas` := other formulas

504. `left rules` := $(\# \Rightarrow)$

505. `right rules` := $(\Rightarrow \#)$

506. *When the upper sequent is provable, its lower sequent is also provable.*

507. The **structural rules** control the *order* (exchange), *duplication* (contraction), and *omission* (weakening) of formulas in the cedents of a given sequent.

508. The left contraction rule means that *each formula occurrence in the antecedents can be used more than once.*

Proofs and Provability (in LK)

509. [3]

510.

$$\begin{aligned} \texttt{P} &:= \text{proof (in } \texttt{LK}\text{) of } (\Gamma \Rightarrow \Delta), \\ &:= \text{a finite tree-like figure defined inductively as follows} \end{aligned}$$

(*i*) every sequent in `P`, except the initial sequents, is obtained by an application of any one of the rules,

(*ii*) $(\Gamma \Rightarrow \Delta)$:= end sequent of `P`.

511. `LK` := sequent system for classical logic

512. $(\Gamma \Rightarrow \Delta)$:= sequent

513. `end sequent` := single lowest sequent

514.

$$(\Gamma \Rightarrow \Delta \text{ is } \texttt{provable in LK}) \leftrightarrow (\texttt{there is a proof} \text{ of } \Gamma \Rightarrow \Delta)$$

515.

$$(\alpha \text{ is } \texttt{provable in LK}) \leftrightarrow (\Rightarrow \alpha \text{ is } \texttt{provable in LK})$$

516. $\alpha :=$ formula

517. $(\Rightarrow \alpha) :=$ sequent

Rules for single formulas (in LK)

518. [3]

519. We will rewrite the rules of `LK` considering only single formulas in the sequents, instead of sequences of formulas, assuming that some sequences are empty.

520. `LK` := sequent system for classical logic

521. `Rules for the logical connectives:`

522.
$$\frac{\alpha \Rightarrow \pi \quad \beta \Rightarrow \pi}{\alpha \vee \beta \Rightarrow \pi}\ (\vee\mathrm{L})$$

523.
$$\frac{\gamma \Rightarrow \alpha}{\gamma \Rightarrow \alpha \vee \beta}\ (\vee\mathrm{R1}) \qquad \frac{\gamma \Rightarrow \beta}{\gamma \Rightarrow \alpha \vee \beta}\ (\vee\mathrm{R2})$$

524.
$$\frac{\alpha \Rightarrow \pi}{\alpha \wedge \beta \Rightarrow \pi}\ (\wedge\mathrm{L1}) \qquad \frac{\beta \Rightarrow \pi}{\alpha \wedge \beta \Rightarrow \pi}\ (\wedge\mathrm{L2})$$

525.
$$\frac{\gamma \Rightarrow \alpha \quad \gamma \Rightarrow \beta}{\gamma \Rightarrow \alpha \wedge \beta}\ (\wedge\mathrm{R})$$

526.
$$\frac{\gamma \Rightarrow \alpha \quad \beta \Rightarrow \pi}{\alpha \rightarrow \beta, \gamma \Rightarrow \pi}\ (\rightarrow\mathrm{L}) \qquad \frac{\alpha, \gamma \Rightarrow \beta}{\gamma \Rightarrow \alpha \rightarrow \beta}\ (\rightarrow\mathrm{R})$$

527.
$$\frac{\gamma \Rightarrow \lambda, \alpha}{\neg\alpha, \gamma \Rightarrow \lambda}\ (\neg\mathrm{L}) \qquad \frac{\alpha, \gamma \Rightarrow \lambda}{\gamma \Rightarrow \lambda, \neg\alpha}\ (\neg\mathrm{R})$$

528. `Cut rule:`

$$\frac{\gamma \Rightarrow \alpha \quad \alpha \Rightarrow \pi}{\gamma \Rightarrow \pi}\ \text{(cut)}$$

529. `Structural rules:`

(i) *exchange rules*

$$\frac{\alpha, \beta \Rightarrow \pi}{\beta, \alpha \Rightarrow \pi}\ \text{(eL)} \qquad \frac{\gamma \Rightarrow \alpha, \beta}{\gamma \Rightarrow \beta, \alpha}\ \text{(eR)}$$

(ii) *contraction rules*

$$\frac{\alpha, \alpha \Rightarrow \pi}{\alpha \Rightarrow \pi}\ \text{(cont L)} \qquad \frac{\gamma \Rightarrow \alpha, \alpha}{\gamma \Rightarrow \alpha}\ \text{(cont R)}$$

(iii) *weakening rules*

$$\frac{\gamma \Rightarrow \pi}{\alpha, \gamma \Rightarrow \pi}\ (w\text{L}) \qquad \frac{\gamma \Rightarrow \pi}{\gamma \Rightarrow \pi, \alpha}\ (w\text{R})$$

Multisets of Formulas

530. [3]

531.

two `multisets` are *distinguished* from each other $\leftrightarrow$
$\leftrightarrow$ the *multiplicity* of any member of them is *different*

532.

$$(\forall \Phi_1, \Phi_2 \in S^* : \Phi_1 \simeq \Phi_2) \leftrightarrow$$
$$\leftrightarrow (\forall s \in S : \text{multiplicity of } s \text{ in } \Phi_1 = \text{multiplicity of } s \text{ in } \Phi_2)$$

533. `multiplicity` := number of occurrences of any formula

534. $\{\alpha, \beta, \alpha\} = \{\beta, \alpha, \alpha\} \neq \{\alpha, \beta\}$

535. S := set of formulas; $\quad S^*$:= set of multisets

536. $S^* :=$ all finite sequence of $s \in S$

537. $\simeq :=$ equivalence relation on S^*

538. $S^* = \{\Phi_i \mid \Phi_i := \text{multiset}\}$

539. $\Phi_1, \Phi_2 :=$ multisets

540.

$$(M = S^*/\simeq) \rightarrow (M := \text{the set of all finite multisets of } s \in S)$$

541. $S^*/\simeq :=$ quotient set

Logical constant 0

542. [3]

543. `0` := *falsum* (falsehood) := arbitrary contradiction

544. $(\neg\alpha) \equiv (\alpha \rightarrow 0)$

545. $(0 \Rightarrow) :=$ initial sequent meaning *the falsum implies anything*

Orthologic

546. [7, 8]

547. `orthologic (minimal quantum logic)` := logic associated with the **order relation** of *ortholattices*

548.

$$\mathcal{O} := \textit{ortholattice} := \text{bounded lattice with } p^\perp$$

549.

$$\forall p \in \mathcal{O} : p \vee p^\perp = \top$$

550. `bounded lattice` := lattice with *smallest* ($\perp$) and *biggest* ($\top$) elements

551. `lattice` := poset such that every two elements have an *infimum* and a *supremum*

552. `poset` := partial ordered set

553. `partial order` := reflexive, transitive, and antisymmetric relation

554. $p^\perp$:= *orthocomplement* (order-reversing involution $p \mapsto \neg p$)

555. In particular, $\forall p, q \in \mathcal{O}$:

$$\begin{array}{rcl} p \leq q & \Rightarrow & q^\perp \leq p^\perp \\ \neg p^\perp & = & p \\ \neg \perp & = & \top \\ \neg(p \vee q) & = & p^\perp \wedge q^\perp \\ \neg(p \wedge q) & = & p^\perp \vee q^\perp \\ p \wedge p^\perp & = & \perp \end{array}$$

556. The other De Morgan's laws hold.

557. $\nexists$ distributive law between $(\wedge, \vee)$

558. In the sequent calculus style the *axiomatization of orthologic* is *sound* and *complete.*

559. `Axiomatization of Orthologic:`

560.

$$\frac{}{A \vdash A}\, ax \qquad\qquad \frac{A \vdash B \quad B \vdash C}{A \vdash C}\, cut$$

561.

$$\frac{}{A \wedge B \vdash A}\, \wedge_1 L \quad \frac{}{A \wedge B \vdash B}\, \wedge_2 L \quad \frac{C \vdash A \quad C \vdash B}{C \vdash A \wedge B}\, \wedge R \quad \frac{}{C \vdash \top}\, \top R$$

562.

$$\frac{}{A \vdash A \vee B} \vee_1 R \quad \frac{}{B \vdash A \vee B} \vee_2 R \quad \frac{A \vdash C \quad B \vdash C}{A \vee B \vdash C} \vee L \quad \frac{}{\bot \vdash C} \bot L$$

563.

$$\frac{A \vdash B}{\neg B \vdash \neg A} \neg \quad \frac{}{A \vdash \neg\neg A} \neg\neg R \quad \frac{}{\neg\neg A \vdash A} \neg\neg L \quad \frac{}{\top \vdash A \vee \neg A} \mathit{tnd}$$

564. (560) $\rightsquigarrow$ (pre) order relation

565. (561) $\rightsquigarrow$ bounded *inf* semi-lattice

566. (562) $\rightsquigarrow$ bounded *sup* semi-lattice

567. (563) $\rightsquigarrow$ ingredients related to the *orthocomplement* $\neg A$

568. (565) + (566) $\rightsquigarrow$ provides the *structure* of a bounded lattice

Intuitionistic Reasoning

569. [9]

570. $\neg A$ is an abbreviation for $A \to \bot$, i.e.,

$$\neg A \equiv (A \to \bot).$$

571. Conjecture: *Nothing is a proof of* $\bot$ *(falsity).*

572. Many `laws` from `classical logic` are *no longer valid* due to the **constructive meaning** of the *intuitionistic connectives.*

573. The **validity** of $A \vee \neg A$ means *there is a method to solve all mathematical problems.*

574. There is a **translation** from `classical` formulas to `intuitionistic` ones.

575. Classical propositional logic can be defined within the intuitionistic logic.

576. $\to, \wedge, \vee$ are all **independent**.

577. In intuitionistic propositional logic, an infinite number of non-equivalent formulas can be built from only one atomic formula P [10].

578. Due to the intuitionistic refinement, equivalent formulas in classical propositional logic become no longer equivalent in intuitionistic propositional logic.

579. The *intuitionistic logic* has a **richer language** than the *classical one*.

580. Atomic formulas and connectives have a constructive interpretation.

Intuitionistic Propositional Logic: Syntax

581. [9]

582.

`alphabet` := consists of the following symbols:

(*i*) $P_1, P_2, P_3, ...$:= *atomic formulas* or *propositional variables* [interpreted as (atomic) propositions]

(*ii*) $\to, \wedge, \vee, \neg$:= *connectives*

(*iii*) (,) := *brackets*

583. `Constructive interpretation of the connectives:`

(*i*) $(A \to B)$:= one has a construction that transforms any proof of A into a proof of B,

(*ii*) $(A \wedge B)$:= one can construct a proof of A and one can construct a proof of B

(iii) $(A \vee B) :=$ one has an algorithm that yields a proof of A or a proof of B

(iv) $(\neg A) := (A \rightarrow \perp)$

(v) $\perp :=$ atomic formula (falsity)

584. A proof of $\perp$ implies a proof of any formula.

585. `Formulas`

(i) $(P := P_1 \veebar P_2 \veebar P_3 \veebar ...) \rightarrow (P :=$ atomic formula$)$

(ii) $(A, B :=$ formulas$) \rightarrow ((A \rightarrow B), (A \wedge B), (A \vee B), (\neg A) :=$ composite formulas$)$

586. $\veebar$ is the exclusive *or*.

Axiom Schema for Intuitionistic Propositional Logic

587. [9]

588. There are *ten axioms* and *one rule* in the intuitionistic propositional logic, which is obtained by replacing the axiom $\neg\neg A \rightarrow A$ of classical logic by $\neg A \rightarrow (A \rightarrow B)$.

589. `Axioms:`

590. $A \rightarrow (B \rightarrow A)$

591. $(A \rightarrow B) \rightarrow ((A \rightarrow (B \rightarrow C)) \rightarrow (A \rightarrow C))$

592. $A \rightarrow (B \rightarrow A \wedge B)$

593. $A \wedge B \rightarrow A$

594. $A \wedge B \rightarrow B$

595. $A \to A \vee B$

596. $B \to A \vee B$

597. $(A \to C) \to ((B \to C) \to (A \vee B \to C))$

598. $(A \to B) \to ((A \to \neg B) \to \neg A)$

599. $\neg A \to (A \to B)$

600. `Rule of inference:` (Modus Ponens)

$$A, A \to B \vdash B.$$

Modal operators

601. [3]

602. $\Box, \Diamond :=$ (unary) modal operators

603. $\Diamond\varphi \equiv \neg\Box\neg\varphi$

604. $\Box$ can be interpreted as *necessarily*.

605. $\Diamond$ can be interpreted as *possibly*.

606. $\varphi :=$ formula

Decision problem

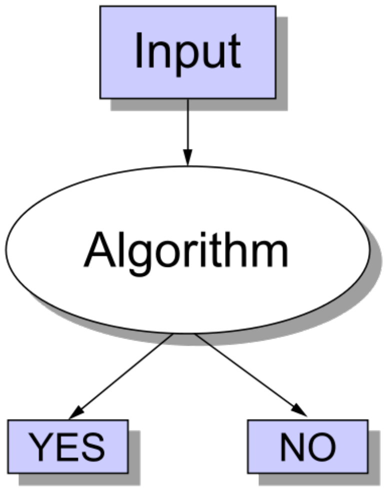
607. [16]

608. In `computability theory` and `computational complexity theory`, a **decision problem** is a *problem that can be posed as a yes–no question of the input values*.

609. An *example* of a *decision problem* is *deciding whether a given natural number is prime*.

610. A *decision problem* which can be *solved* by an *algorithm* is called `decidable`.

611.



Undecidable

612. [15]

613. In computability theory and computational complexity theory, an `undecidable problem` is a **decision problem** for which it is *proved to be impossible to construct an algorithm that always leads to a correct yes-or-no answer.*

614. The *halting problem* is an *example*: it can be proven that *there is no algorithm that correctly determines whether arbitrary programs eventually halt when run.*

Word problem

615. [17]

616. In mathematics and computer science, a `word problem` for a set S with respect to a system of finite encodings of its elements is the *algorithmic problem of deciding whether two given representatives represent the same element of the set.*

617. The problem is commonly encountered in abstract algebra, where given a presentation of an algebraic structure by generators and relators, *the problem is to determine if two expressions represent the same element; a prototypical example is the word problem for groups.*

618. Less formally, the word problem in an algebra is: given a set of identities E, and two expressions x and y, *is it possible to transform x into y using the identities in E as rewriting rules in both directions?*

619. While answering the question in (618) may not seem hard, the remarkable (and deep) result that emerges, in many important cases, is that the *problem is undecidable.*

620. Many, if not most all, `undecidable problems` in mathematics *can be posed as word problems.*

621. List of undecidable problems
`https://en.wikipedia.org/wiki/List_of_undecidable_problems`

Natural Deduction in Heyting Semantics

622. [19, 20]

623. rules of natural deduction + Heyting Semantics $\rightsquigarrow$ special way of constructing functions

624. $A, B, B_i :=$ formulas

625. formula $A :=$ set of its possible deductions; e.g., if $A = \{\alpha, \beta\}$ then both α and β prove A

626. hypotheses $B_i \in A$

627. $(B_1, ..., B_n \vdash A) \;\equiv\; t[x_1, ..., x_n] : B_1 \times ... \times B_n \to A$

628. $x_i :=$ variables

629. Two occurrences of the same formula B_i in the same *parcel of hypotheses* correspond to the same variable.

630. `The rules`

(*i*) **Hypothesis:** A

(*ii*) **Introductions:**

$$\frac{A \quad B}{A \wedge B}\wedge\mathcal{I} \qquad \frac{\begin{matrix}[A]\\ \vdots \\ B\end{matrix}}{A \to B}\to\mathcal{I}x \qquad \frac{A}{\forall x.A}\forall\mathcal{I} \qquad \frac{A[a/x]}{\exists x.A}\exists\mathcal{I}$$

$$\frac{A}{A \vee B}\vee 1\mathcal{I} \qquad \frac{B}{A \vee B}\vee 2\mathcal{I} \qquad \frac{\begin{matrix}[A]\\ \vdots \\ \bot\end{matrix}}{\neg A}\neg\mathcal{I}$$

$$\frac{\begin{matrix}[A]\\ \vdots \\ B\end{matrix} \quad \begin{matrix}[B]\\ \vdots \\ A\end{matrix}}{A \leftrightarrow B}\leftrightarrow\mathcal{I}$$

(*iii*) **Eliminations:**

$$\frac{A \wedge B}{A}\wedge 1\mathcal{E} \qquad \frac{A \wedge B}{B}\wedge 2\mathcal{E} \qquad \frac{A \to B \quad A}{B}\to\mathcal{E}$$

$$\frac{\exists x.A \quad \begin{matrix}[A]\\ \vdots \\ B\end{matrix}}{B}\exists\mathcal{E} \qquad \frac{\forall x.A}{A[a/x]}\forall\mathcal{E}$$

$$\dfrac{A \vee B \qquad \begin{matrix}[A]\\ \vdots \\ C\end{matrix} \qquad \begin{matrix}[B]\\ \vdots \\ C\end{matrix}}{C}\ \vee\mathcal{E} \qquad \dfrac{\neg A \qquad A}{\bot}\ \neg\mathcal{E}$$

$$\dfrac{A \leftrightarrow B \qquad A}{B}\ \leftrightarrow\mathcal{E}1 \qquad \dfrac{A \leftrightarrow B \qquad B}{A}\ \leftrightarrow\mathcal{E}2$$

(*iv*) **Absurdity:**

$$\begin{matrix}[\neg A]\\ \vdots \\ \dfrac{\bot}{A}\ \bot\end{matrix} \qquad \dfrac{\bot}{B}\ \bot\mathcal{E}$$

631. In $\exists\mathcal{E}$, x **cannot be free** in B and in any hypothesis that has not being canceled, except in A, in the deduction of B.

632. In $\forall\mathcal{I}$, x **cannot be free** in any hypothesis that has not being canceled in the deduction of A.

633. In (630), a is free for x in A.

634. The left deduction of (630.*iv*) is called *reductio ad absurdum.*

635. The `fingerprint` of classical logic is the *reductio ad absurdum.*

636. Interpretation of the rules

   (*i*) $\exists! B_1 : B_1 \vdash A \ \Rightarrow\ x \equiv (B_1 \vdash A) \ \Rightarrow\ x \in B_1 \in A$

   (*ii*) $(u[x_1, ..., x_n] : A) \ \wedge\ (v[x_1, ..., x_n] : B) \ \Rightarrow$
   $\Rightarrow \langle u[x_1, ..., x_n], v[x_1, ..., x_n]\rangle : A \wedge B$
   (note that u and v have been made to depend on the same variables; their choices are correlated)

$$\frac{\vdots}{u : A} \qquad \frac{\vdots}{v : B} \qquad \dfrac{\begin{matrix}\vdots & & \vdots \\ u : A & & v : B\end{matrix}}{\langle u, v\rangle : A \wedge B}$$

(iii) $t[x_1, ..., x_n] : A \wedge B \;\Rightarrow\; \pi^1 t[x_1, ..., x_n] : A$
$t :=$ proof of a conjunction
$\pi^1 t :=$ first projection
$\pi^2 t : B$
$\pi^2 t :=$ second projection

$$\frac{\vdots}{t : A \wedge B} \qquad \dfrac{\begin{matrix}\vdots \\ t : A \wedge B\end{matrix}}{\pi^1 t : A} \qquad \dfrac{\begin{matrix}\vdots \\ t : A \wedge B\end{matrix}}{\pi^2 t : B}$$

The following *equations* are the *essence* of the ***correspondence*** between *logic* and *computer science*:

$$\pi^1\langle u, v\rangle = u; \qquad \pi^2\langle u, v\rangle = v; \qquad \langle \pi^1 t, \pi^2 t\rangle = t.$$

$$\dfrac{\dfrac{\begin{matrix}\vdots & & \vdots \\ u : A & & v : B\end{matrix}}{\langle u, v\rangle : A \wedge B}}{u : A} \qquad \frac{\vdots}{u : A}$$

$$\dfrac{\dfrac{\begin{matrix}\vdots & & \vdots \\ u : A & & v : B\end{matrix}}{\langle u, v\rangle : A \wedge B}}{\pi^2\langle u, v\rangle : A} \qquad \frac{\vdots}{v : B}$$

(*iv*) $\lambda x.v$ is a function from A to B with $v[a, x_1, ..., x_n] \in V$, $a \in A$
(in $\lambda x.v[x, x_1, ..., x_n]$, x is bound)
(note that *binding* corresponds to *discharge*)

$$\dfrac{\begin{array}{c}[x : A]\\ \vdots \\ v : B\end{array}}{\lambda x.v : A \to B}$$

(*v*) $(t[x_1, ..., x_n] : A \to B) \ \wedge \ (u[x_1, ..., x_n] : A) \ \Rightarrow$
$\Rightarrow \ t[x_1, ..., x_n]u[x_1, ..., x_n] : B$
$t : A \to B$ for fixed values of $x_1, ..., x_n$
$u \in A$; $\ t(u) \in B$

$$\dfrac{t : A \to B \qquad u : A}{tu : B}$$

We have:

$$(\lambda x.v)u = v[u/x],$$
$$\lambda x.tx = t \quad \text{(when } x \text{ is not free in } t\text{)}.$$

$$\dfrac{\begin{array}{c}[x : A]\\ \vdots \\ v : B\end{array}}{\lambda x.v : A \to B} \qquad \dfrac{t : A \to B \qquad u : A}{v : B} \qquad \dfrac{\begin{array}{c}[x : A]\\ \vdots\end{array}}{v[u/x] : B}$$

637. In `natural deduction`, a **proof** is **normal** if it does <u>not</u> contain any sequence of an *introduction* and an *elimination* rule.
(*menemonic rule:* $\texttt{Nn}_{ie}$)

Lambda Calculus: Types

638. [19]

639. In *Heyting's approach*, **formulas** become **types**.

640. The only types are the following:

(i) $T_1, ..., T_n :=$ atomic types $:=$ types;

(ii) $(U, V :=$ types$) \Rightarrow (U \times V,\ U \to V :=$ types$)$.

Lambda Calculus: Terms

641. [19]

642. **Proofs** become **terms**.

643. *mnemonic rule:* $(\texttt{ft}_y\texttt{.pt}_e) \equiv$ (formulas $\sim\succ$ types, proofs $\sim\succ$ terms)

644. `term of type` $A :=$ *proof* of a *formula* A

645. $x_0^T, ..., x_n^T, ... :=$ terms of type T

646. $(u, v :=$ terms of types U and $V) \to (\langle u, v\rangle :=$ term of type $U \times V)$

647. $(t :=$ term of type $U \times V) \to (\pi^1 t, \pi^2 t :=$ terms of types U and V, respectively$)$

648. $((v :=$ term of type $V) \wedge (x_n^U :=$ variable of type $U)) \to$
$\to (\lambda x_n^U.v :=$ term of type $U \to V)$

649.

$$\frac{\begin{array}{c}[x_n^U \in U]\\ \vdots \\ v \in V\end{array}}{\lambda x_n^U.v \in U \to V}$$

650. $(t, u :=$ terms of type $U \to V$ and U, respectively$) \to$
$\to (t\,u :=$ term of type $V)$

Lambda Calculus: Denotational significance

651. [19]

652. (object of type $U \to V$) $\equiv$ (function $f : U \to V$)

653. (object of type $U \times V$) $\equiv$ (ordered pair $\langle u, v\rangle$, $u \in U$ and $v \in V$)

654. x^T := variable of type T

655. $\langle u, v\rangle$:= ordered pair

656. $\pi^1 t$:= first projection of t

657. $\pi^2 t$:= second projection of t

658. $\lambda x^U.v : U \to V$ such that $\lambda x^U.v[u] = v[u/x]$ with $x^U \equiv u$

659. u := object of type U

660. tu := function t applied to the argument u

661. The following are *primary* equations:

$$\begin{aligned} \pi^1\langle u, v\rangle &= u, \\ \pi^2\langle u, v\rangle &= v, \\ (\lambda x^U.v)u &= v[u/x]. \end{aligned}$$

662. The following are *secondary* equations:

$$\begin{aligned} \langle \pi^1 t, \pi^2 t\rangle &= t, \\ \lambda x^U.tx &= t \quad (x \text{ not free in } t). \end{aligned}$$

System of equations in lambda calculus: Consistent and decidable

663. [19]

664. **Theorem.** The system given by (661) and (662) is `consistent` and `decidable`.

665. ***Consistency*** means that $x = y$, where x and y are *distinct variables*, cannot be proved.

Conversion

666. [19]

667. $t, t' :=$ terms

668. In `natural deduction`, a **proof** is **normal** if it does not contain any sequence of an *introduction* and an *elimination* rule.
(*menemonic rule:* $\texttt{Nn}_{ie}$)

669. $(\lambda x^U.v)u \;\sim\!\succ$ introduction

670. $\{\pi^1\langle u,v\rangle, \pi^2\langle u,v\rangle\} \sim\!\succ$ elimination

671. none subterms are of the form $(\lambda x^U.v)u$ or $\pi^1\langle u,v\rangle$ or $\pi^2\langle u,v\rangle \Rightarrow$
$\Rightarrow$ `term` := **normal form**

672. t converts to t' if either:

$$
\begin{aligned}
(i)\ & t = \pi^1\langle u,v\rangle,\ \ t' = u; \quad \text{or} \\
(ii)\ & t = \pi^2\langle u,v\rangle,\ \ t' = v; \quad \text{or} \\
(iii)\ & t = (\lambda x^U.v)u,\ \ t' = v[u/x].
\end{aligned}
$$

$$
\dfrac{\begin{matrix}[x^U \in U] \\ \vdots \\ v \in V\end{matrix}}{\lambda x^U.v \in U \to V}
$$

673. $t :=$ `redex`

674. $t' :=$ `contractum`

675. t and t' are of the same type

676. $\exists$ sequence $u = t_0, t_1, ..., t_{n-1}, t_n = v$: for $i = 0, 1, ..., n-1$, t_{i+1} is obtained from t_i by *replacing* a **redex** by its **contractum** $\Rightarrow$ $\Rightarrow$ $u \rightsquigarrow v$

677. $(u \rightsquigarrow v) := u$ reduces to v

678. $\rightsquigarrow$ is *reflexive* and *transitive.*

679. $((t \rightsquigarrow u) \wedge (u := \texttt{normal})) \equiv (\exists! u : u :=$ normal form for $t)$

680. $(t := \texttt{normal}) \leftrightarrow t$ is in `head normal form` $(\lambda x_1 x_2 ... x_n . y\, u_1 u_2 ... u_m)$ (where $y = x_i \veebar y \neq x_i$, u_j are normal)

681. *A term converts in one step, reduces in many.*

682. **Conversion** can be identified as `rewriting`, the left member being rewritten to the right one.

The Curry-Howard Isomorphism

683. [18–20]

684. This is an **isomorphism** between `proofs` and `functional terms`.

685. variable $x_i^A \equiv$ deduction A (A in parcel i)

686. Recall the following rules for natural deduction

   (*i*) **Hypothesis:** $x : A$

   (*ii*) **Introductions:**

$$\frac{x : A \qquad y : B}{xy : A \wedge B}\ \wedge\mathcal{I} \qquad \frac{\begin{matrix}[x : A] \\ \vdots \\ y : B\end{matrix}}{\lambda x . xy : A \to B}\ \to \mathcal{I}x \qquad \frac{x : A}{\forall \xi . A}\ \forall\mathcal{I} \qquad \frac{A[a/\xi]}{\exists \xi . A}\ \exists\mathcal{I}$$

$$\frac{x : A}{A \vee B}\ \vee 1\mathcal{I} \qquad \frac{y : B}{A \vee B}\ \vee 2\mathcal{I} \qquad \frac{\begin{array}{c}[x : A]\\ \vdots\\ \bot\end{array}}{\neg A}\ \neg\mathcal{I}$$

$$\frac{\begin{array}{cc}[A] & [B]\\ \vdots & \vdots\\ B & A\end{array}}{A \leftrightarrow B}\ \leftrightarrow \mathcal{I}$$

(iii) **Eliminations:**

$$\frac{xy : A \wedge B}{x : A}\ \wedge 1\mathcal{E} \qquad \frac{xy : A \wedge B}{y : B}\ \wedge 2\mathcal{E} \qquad \frac{\lambda x.xy : A \to B \qquad x : A}{y : B}\ \to \mathcal{E}$$

$$\frac{\begin{array}{cc} & [A]\\ & \vdots\\ \exists x.A & B\end{array}}{B}\ \exists\mathcal{E} \qquad \frac{\forall \xi.A}{A[a/\xi]}\ \forall\mathcal{E}$$

$$\frac{\begin{array}{ccc} & [x : A] & [y : B]\\ & \vdots & \vdots\\ A \vee B & z : C & z : C\end{array}}{z : C}\ \vee\mathcal{E} \qquad \frac{\neg A \qquad A}{\bot}\ \neg\mathcal{E}$$

$$\frac{A \leftrightarrow B \qquad A}{B}\ \leftrightarrow \mathcal{E}1 \qquad \frac{A \leftrightarrow B \qquad B}{A}\ \leftrightarrow \mathcal{E}2$$

(iv) **Absurdity:**

$$\frac{\begin{array}{c}[\neg A]\\ \vdots\\ \bot\end{array}}{x : A}\ \bot$$

687.

$$\frac{\vdots}{u : A} \qquad \frac{\vdots}{v : B} \qquad \frac{\begin{array}{cc}\vdots & \vdots\\ u : A & v : B\end{array}}{\langle u, v\rangle : A \wedge B}\ \wedge\mathcal{I}$$

688.

$$\dfrac{\vdots}{t : A \wedge B} \qquad \dfrac{\begin{matrix}\vdots \\ t : A \wedge B\end{matrix}}{\pi^1 t : A}\ {\scriptstyle \wedge 1\mathcal{E}} \qquad \dfrac{\begin{matrix}\vdots \\ t : A \wedge B\end{matrix}}{\pi^2 t : B}\ {\scriptstyle \wedge 2\mathcal{E}}$$

689. if the deleted hypotheses form parcel i

$$\dfrac{\begin{matrix}[x_i : A] \\ \vdots \\ v : B\end{matrix}}{\lambda x_i^A . v : A \to B}\ {\scriptstyle \to \mathcal{I} x_i}$$

690. term tu

$$\dfrac{t : A \to B \qquad u : A}{tu : B}\ {\scriptstyle \to \mathcal{E}}$$

691. **Conversion**, **normality**, and **reduction** *correspond perfectly on both sides of the isomorphism.*
(*mnemonic*: `cnr.iso`)

The Normalization Theorem

692. [19]

693. typed λ-calculus $\rightsquigarrow$ behaves well computationally

694. `Normalization Theorem` $\rightsquigarrow$ **existence** (`normal form`)

695. `Church-Rosser property` $\rightsquigarrow$ **uniqueness** (`normal form`)

696. *mnemonic:* `NeCRu`

697. (694) $\rightsquigarrow$ two forms:

(*i*) `weak` $\rightsquigarrow$ $\exists$ terminating strategy (normalization)

(*ii*) `strong` $\rightsquigarrow$ all possible strategies (normalization) terminate

The lambda-calculus: Introduction

698. [21]

699. λ-calculus $\rightsquigarrow$ collection of several formal systems

700. Example:

701. $f(x) = x - y; \qquad g(y) = x - y$

702. $f : x \mapsto x - y; \qquad g : y \mapsto x - y$

703. $f = \lambda x.x - y; \qquad g = \lambda y.x - y$

704. $f(0) = 0 - y; \qquad f(1) = 1 - y$

705. $(\lambda x.x - y)(0) = 0 - y; \qquad (\lambda x.x - y)(1) = 1 - y$

The lambda-calculus: Formal system

706. [21]

707. λ-`term` := **atom** $\veebar$ **application** $\veebar$ **abstraction**

    (*a*) v_i, c_i := λ-terms (`atoms`)

    (*b*) $(M, N := \lambda\text{-terms}) \rightarrow ((MN) := \lambda\text{-term}$ (`application`))

    (*c*) $(M := \lambda\text{-term} \ \wedge \ x := \text{variable}) \rightarrow$
    $\rightarrow ((\lambda x.M) := \lambda\text{-term}$ (`abstraction`))

708. v_i := variables

709. c_i := atomic constants

710. $x, y, z :=$ distinct variables $\Rightarrow M = yz \Rightarrow (\lambda x.M) = (\lambda x.(yz)) :=$ `vacuous abstraction` (x does not occur in M) $:=$ constant functions

711. λ and λx are not terms.

712. $M, N, P, Q, ... :=$ λ-terms

713. $x, y, z, u, v, w, ... :=$ variables

714. $M \equiv N$ means *syntactic identity*, i.e., M is exactly the same term as N.

715. Application: $MNPQ \equiv ((((MN)P)Q)$
(association from *left to right*)

716. $\lambda x.PQ \equiv (\lambda x.(PQ))$

717. Abstraction: $\lambda x_1 x_2 ... x_n.M \equiv (\lambda x_1.(\lambda x_2.(...(\lambda x_n.M))))$
(from *right to left*)

718. *menemonic:* `app.lr`, `abs.rl`

719. $(MN \equiv PQ) \rightarrow (M \equiv P \ \wedge \ N \equiv Q)$

720. $(\lambda x.M \equiv \lambda y.P) \rightarrow (x \equiv y \ \wedge \ M \equiv P)$

721. $k = 0$ in $P \equiv MN_1...N_k$ $(k \geq 0)$ means $P \equiv M$.

722. $n = 0$ in $\lambda x_1...x_n.PQ$ means PQ.

723. $\lambda :=$ (abbreviated as) λ-calculus in general

724. `iff` $:=$ if and only if

The lambda-calculus: Informal interpretation

725. [21]

726. ($M :=$ `function/operator`) $\Rightarrow$ ($MN :=$ `application` of M to N)

727. $(\lambda x.M)(N) :=$ `operator/function` substituting N for x in M

728. $xy :=$ application

729. $\lambda x.x(xy) :=$ the operation of applying a function twice to y

730. $(\lambda x.x(xy))(N) = N(Ny)$ holds for all terms N.

731. $\lambda x.y :=$ constant function (value y for all arguments)

732. $(\lambda x.y)N = y$

Lambda-terms: Length, occurrence, scope, free and bound variables, substitution

733. [21]

734.
$$lgh(M) := \texttt{total number of occurences} \text{ of } c_i, v_i \text{ in } M$$
   (a) $lgh(a) = 1$
   (b) $lgh(MN) = lgh(M) + lgh(N)$
   (c) $lgh(\lambda x.M) = 1 + lgh(M)$

735. $lgh(M) :=$ length of M

736. $M, N, P, Q := \lambda$-terms

737. $c_i, v_i, a, x := \lambda$-terms (atoms)

738. $x, y, z, u, v, v_i :=$ variables

739. *induction* on $M \equiv$ *induction* on $lgh(M)$

740. e.g., $M \equiv xyz(\lambda xy.uv) \ \rightarrow \ lgh(M) = 7$

741.

$$P \texttt{ occurs in } Q \equiv P \texttt{ is a subterm of } Q \equiv Q \texttt{ contains } P$$
$$(\textit{relation} \text{ defined by } \textit{induction} \text{ on } Q)$$

(a) P occurs in P

(b) (P occurs in M) $\vee$ (P occurs in N) $\rightarrow$ (P occurs in MN)

(c) (P occurs in M) $\vee$ ($P \equiv x$) $\rightarrow$ (P occurs in $\lambda x.M$)

742. In $z(\lambda y.(xyz))$ there are two occurences of z and y, and one occurrence of x.

743. In $\lambda x.M$, M is the `scope` of λx.

744. (i) ($x \in M$ in $\lambda x.M$) $\rightarrow$ (x is `bound`)

(ii) the x in λx is `bound and binding`

(iii) x is `free` otherwise

745. In $x\lambda x.x$, the left x is a *free variable* and the right x is a *bound variable.*

746. $\mathrm{FV}(P)$:= set of all free variables of P

747. `closed term` := a term with no free variables

748.

$$[N/x]M := \texttt{substitution of } N, \ \forall x^f \in M$$

749. x^f := free occurrence of x

750. The definition of `substitution` is by *induction* on M:
(let $x \not\equiv y$ and $z \notin \mathrm{FV}(NP)$)

(a) $[N/x]x \equiv N$

(b) $[N/x]a \equiv a, \ \forall a \not\equiv x$

(c) $[N/x](PQ) \equiv ([N/x]P \ [N/x]Q)$

(d) $[N/x](\lambda x.P) \equiv \lambda x.P$

(e) $x \notin \mathrm{FV}(P) \ \rightarrow\ [N/x](\lambda y.P) \equiv \lambda y.P$

(f) $(x \in \mathrm{FV}(P) \wedge y \notin \mathrm{FV}(N)) \ \rightarrow\ [N/x](\lambda y.P) \equiv \lambda y.[N/x]P$

(g) $(x \in \mathrm{FV}(P) \wedge y \in \mathrm{FV}(N)) \ \rightarrow\ [N/x](\lambda y.P) \equiv \lambda z.[N/x][z/y]P$

751. (a) $[x/x]M \equiv M$

(b) $x \notin \mathrm{FV}(M) \ \rightarrow\ [N/x]M \equiv M$

(c) $x \in \mathrm{FV}(M) \ \rightarrow\ \mathrm{FV}([N/x]M) = \mathrm{FV}(N) \cup (\mathrm{FV}(M) - \{x\})$

(d) $lgh([y/x]M) = lgh(M)$

752. Let x, y, v be distinct, let no variable bound in M be free in vPQ

(a) $v \notin \mathrm{FV}(M) \ \rightarrow\ [P/v][v/x]M \equiv [P/x]M$

(b) $v \notin \mathrm{FV}(M) \ \rightarrow\ [x/v][v/x]M \equiv M$

(c) $y \notin \mathrm{FV}(P) \ \rightarrow\ [P/x][Q/y]M \equiv [([P/x]Q)/y][P/x]M$

(d) $y \notin \mathrm{FV}(P) \wedge x \notin \mathrm{FV}(Q) \ \rightarrow\ [P/x][Q/y]M \equiv [Q/y][P/x]M$

(e) $[P/x][Q/x]M \equiv [([P/x]Q)/x]M$

Lambda-terms: Change of bound variables, congruence

753. [21]

754. P contains an occurrence of $\lambda x.M$.

755. $y \notin \mathrm{FV}(M)$

756.

$$(\lambda x.M \equiv \lambda y.[y/x]M) := \texttt{change of bound variable}$$
$$(\alpha\texttt{-conversion} \text{ in } P)$$

757. $(P \equiv_\alpha Q) \leftrightarrow P$ can be converted to Q by a finite (or empty) number of changes (756)

758. $(P \equiv_\alpha Q) :=$ P is `congruent to` $Q :=$ P α-`converts to` Q

759.

$$P \equiv_\alpha Q \;\to\; \mathrm{FV}(P) = \mathrm{FV}(Q)$$

760. $\equiv_\alpha$ is an *equivalence relation.*

761. Removing the condition on bounded variables in M, (752) also holds for $\equiv_\alpha$.

762.

$$(M \equiv_\alpha M') \wedge (N \equiv_\alpha N') \;\to\; [N/x]M \equiv_\alpha [N'/x]M'$$

763. (762) shows that substitution is well-behaved regarding $\equiv_\alpha$.

764. *We can think of* $\equiv$ *and* $\equiv_\alpha$ *as being identical.*

Lambda-terms: Simultaneous substitution

765. [21]

766. See (750).

767.

$$[N_1/x_1, ..., N_n/x_n]M := \texttt{simultaneous substitution} \text{ for } n \geq 2$$

768. $[N_1/x_1, ..., N_n/x_n]M$ can be different from $[N_1/x_1]...[N_n/x_n]M$.

Lambda-terms: β-reduction

769. [21]

770.

$$(\lambda x.M)N := \beta\texttt{-redex of } [N/x]M$$

771.

$$[N/x]M := \texttt{contractum} \text{ of } (\lambda x.M)N$$

772. In this context $\supseteq$ means *contains an occurrence of* a λ-term.

773.

$$(P \supseteq (\lambda x.M)N) \ \wedge \ P' \equiv [[N/x]M]P/(\lambda x.M)N) \quad \leftrightarrow \quad P \rhd_{1\beta} P'$$

774. $(P \rhd_{1\beta} P') := P$ β-`contracts` to P' (contraction of the redex-occurrence in P)

775. $(P \rhd_{\beta} P') := P$ β-`reduces to` Q
iff P can be changed to Q by a finite number of β-contractions and changes of bound variables

776. β-reduction not necessarily simplifies a term; it terminates when there are no redexes.

Lambda-terms: β-normal form

777. [21]

778. β-`normal form` (β-nf) := a term with no β-redexes

779. β-nf (or $\lambda\beta$-nf) := `class of all` β-`normal forms`

780. $P \rhd_{1\beta} (Q \text{ in } \beta\text{-}nf) \ \rightarrow \ Q := \beta$-`normal form of` P

781. $P, Q :=$ terms

782. A term can have a *normal form* and also an *infinite reduction*.

783. $\Omega \equiv (\lambda x.xx)(\lambda x.xx)$

784. Ω is not a normal form (it always reduces to itself)

785. $\Omega :=$ `minimal` (it cannot be reduced to any different term)

786. The α-steps (756) are allowed in β-reductions in order to change bound variables at the beginning of the reduction and therefore avoid having to change variables while substituting.

787. lambda-calculus $\sim$ programming language $\rightsquigarrow$ two β-reductions reach the same normal form $\rightsquigarrow$ the end-result is independent of the path $\rightsquigarrow$ Church-Rosser theorem: the normal form of a term is unique

788. $\triangleright_\beta$, FV, and $\supseteq$: (*nothing new can be introduced during a reduction*)

$$P \triangleright_\beta Q \;\to\; \mathrm{FV}(P) \supseteq \mathrm{FV}(Q)$$

789. `Substitution` and $\triangleright_\beta$: ($\triangleright_\beta$ *is preserved by substitution*)

$$(P \triangleright_\beta P') \land (Q \triangleright_\beta Q') \;\to\; [P/x]Q \triangleright_\beta [P'/x]Q'$$

790. `Church-Rosser theorem for` $\triangleright_\beta$

$$(P \triangleright_\beta M) \land (P \triangleright_\beta N) \;\to\; \exists T : M(\triangleright_\beta T) \land (N \triangleright_\beta T)$$

$$\begin{tikzcd}
 & P \arrow[dl] \arrow[dr] & \\
M \arrow[dr, dashed] & & N \arrow[dl, dashed] \\
 & \exists\, T &
\end{tikzcd}$$

791. The property in (790) is called `confluence`.

792. The theorem (790) states that *β-reduction is confluent.*

793. *If P has a β-normal form, it is unique modulo* $\equiv_\alpha$

$$(P \triangleright_\beta M) \land (P \triangleright_\beta N) \;\to\; M \equiv_\alpha N$$

794. β-nf is the `smallest class` such that:

(*a*) $\forall a\,(a \in \beta\text{-nf})$

(b) $M_1, ..., M_n \in \beta\text{-nf} \ \rightarrow \ \forall a : aM_1...M_n \in \beta\text{-nf}$

(c) $M \in \beta\text{-nf} \ \rightarrow \ \lambda x.M \in \beta\text{-nf}$

795. $a :=$ `atoms`

796.

$$(M \equiv aM_1...M_n) \wedge (M \rhd_\beta N) \wedge (M_i \rhd_\beta N_i \text{ for } i = 1, ...n) \ \rightarrow \\ \rightarrow \ N \equiv aN_1...N_n$$

Lambda-terms: β-equality

797. [21]

798.

$$P =_\beta Q \quad \leftrightarrow \quad \exists P_0, ..., P_n \ (n \geq 0) : \\ (\forall i \leq n-1)(P_i \rhd_{1\beta} P_{i+1} \ \vee \ P_{i+1} \rhd_{1\beta} P_i \ \vee \ P_i \equiv_\alpha P_{i+1}), \\ P_0 \equiv P, \qquad P_n \equiv Q$$

799. $(P =_\beta Q) := P$ is β-`equal` (β-`convertible`)

800. $(P =_\beta Q)$ means Q *can be obtained from* P *by a finite (or empty) (reversed)* β*-contractions and changes of variables.*

801.

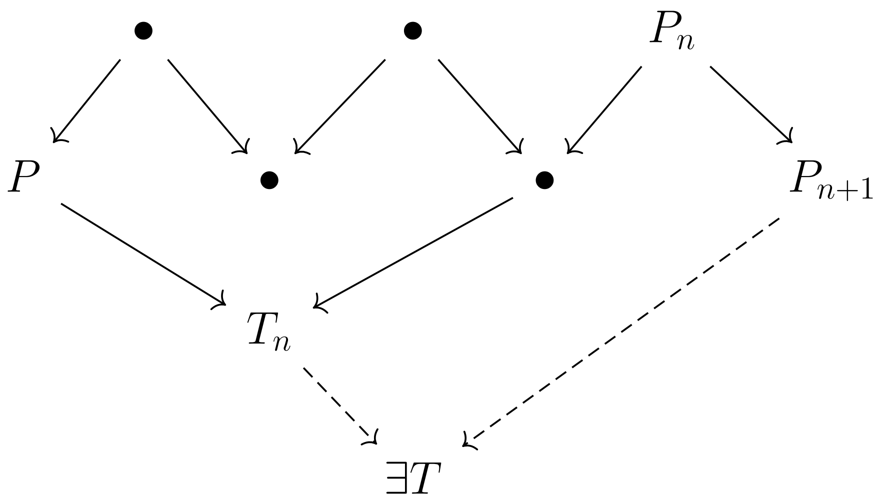
$$(P =_\beta Q) \wedge (P \equiv_\alpha P') \wedge (Q \equiv_\alpha Q') \ \rightarrow \ P' =_\beta Q'$$

802. `Substitution lemma for` β`-equality`

$$(M =_\beta M') \wedge (N =_\beta N') \ \rightarrow \ [N/x]M =_\beta [N'/x]M'$$

803. `Church-Rosser theorem for` $=_\beta$

$$P =_\beta Q \ \rightarrow \ \exists T : (M \rhd_\beta T) \wedge (N \rhd_\beta T)$$



Two β-convertible terms can both be reduced to the same term.

804. β-`convertibility` is called $=$.

805.

$$(P =_\beta Q) \wedge (Q := \beta\text{-normal form}) \;\rightarrow\; P \triangleright_\beta Q$$

806.

$$(P =_\beta Q) \;\rightarrow\; (P, Q := \text{same } \beta\text{-nf}) \veebar (P, Q := \text{no } \beta\text{-nf})$$

807.

$$(P, Q \in \beta\text{-nf}) \wedge (P =_\beta Q) \;\rightarrow\; P \equiv_\alpha Q$$

808. the relation β-nf is non-trivial $\sim\succ$ not all terms are β-convertible to each other

809. e.g., since $\lambda xy.xy \not\equiv_\alpha \lambda xy.yx$ then $\lambda xy.xy \neq_\beta \lambda xy.yx$

810. `Uniqueness of normal form`: *A term is β-equal to at most one β-normal form, modulo changes of bound variables.*

811.

$$(a, b := \text{atoms}) \wedge (aM_1...M_m =_\beta bN_1...N_n) \;\rightarrow$$
$$\rightarrow\; (a \equiv b) \wedge (m = n) \wedge (M_i =_\beta N_i, \forall i \leq m)$$

812. terms without normal forms $\sim\succ$ computed for ever (without reaching a result)

813. λI-terms

(a) $v_i, c_i :=$ λI-terms (atoms)

(b) $(M, N := \lambda\text{I-terms}) \rightarrow ((MN) := \lambda\text{-term}$ (application))

(c) $(M := \lambda\text{I-term} \;\wedge\; x := \text{free variable in } M) \rightarrow$
$\rightarrow ((\lambda x.M) := \lambda\text{I-term}$ (abstraction))

814. (λI-term := has a normal form) $\rightarrow$ (all its subterms have a normal form)

Simple typing, Church-style

815. [21]

816. mathematics $\sim\succ$ definition + function $\sim\succ$ statement of the kind (inputs + outputs)

817. λ-calculus $\sim\succ$ modify λ $\sim\succ$ *attach expressions to terms* (called types) $\sim\succ$ like labels (to denote input/output sets)

818. two approaches

(i) *Church-style* (explicit or rigid)

(ii) *Curry-style* (implicit)

819. Church-style $\sim\succ$ *term's type* is a *built-in part* of the *term*

820. atomic types := finite/infinite sequence of symbols

821. Simple types

(a) $(\forall a : a := \text{atomic type}) \;\rightarrow\; (a :=$ type)

(b) $(\sigma, \tau := \text{types}) \;\rightarrow\; ((\sigma \rightarrow \tau) :=$ function type)

822. atomic type $\sim\succ$ denotes a set

823. $\mathtt{N}$:= atomic type for the set of natural numbers

824. $(\sigma \to \tau)$:= `set of functions` from σ (*domain*) to τ (*range*)

825. $(\mathtt{N} \to (\mathtt{N} \to \mathtt{N}))$:= `set of functions` from *numbers* to *functions*

826. $(\rho \to \sigma \to \tau) \ \equiv \ (\rho \to (\sigma \to \tau))$
(association from *right to left*)

Typed λ-calculus

827. [21]

828. x := untyped variable

829. τ, σ := types

830. $\exists_\infty$:= there is an infinite number

831. `Typed variables`

$$x^\tau := \texttt{variable of type } \tau$$

(a) (*consistency condition*) $\nexists x : (\exists x^\tau \exists x^\sigma) \wedge (\tau \not\equiv \sigma)$

(b) $\forall \tau \, \exists_\infty x_i^\tau$

832. $x^\tau \in \tau$

833. $x^{\mathtt{N}}$:= arbitrary number

834. $x^{\mathtt{N} \to \mathtt{N}}$:= function

835. x^τ := `typed` *variables*

836. c^τ := `typed` *atomic constants*

837. `Simply typed` λ`-terms`

(a) x^τ, c^τ := `typed` λ`-terms`

(b) $(M^{\sigma\to\tau}, N^{\sigma}$:= typed λ-terms) $\to$ $(M^{\sigma\to\tau}N^{\sigma})^{\tau}$:= `typed` λ`-term of type` τ

(c) $(x^{\sigma}$:= typed variable) $\wedge$ $(M^{\tau}$:= typed λ-term) $\Rightarrow$
$\Rightarrow$ $(\lambda x^{\sigma}.M^{\tau})^{\sigma\to\tau}$:= `typed` λ`-term of type` $\sigma \to \tau$

838. M^{τ} := typed term

839. $M^{\tau} \in \tau$

840. $(M^{\sigma\to\tau}$:= function ϕ from σ to τ) $\wedge$ $(N^{\sigma}$:= member a of σ) $\Rightarrow$
$\Rightarrow$ $(M^{\sigma\to\tau}N^{\sigma})^{\tau} := \phi(a) \in \tau$

841. e.g., $\overline{0}^{\mathtt{N}}$ (atom) := zero; $\overline{\sigma}^{\mathtt{N}\to\mathtt{N}}$:= successor function

The Sequent Calculus LJ

842. [22]

843. LJ := intuitionistic logic

844. The following notation is an abbreviation for an inductive definition

$$A ::= X \mid A \to A.$$

845. ::= is a definition by induction.

846. Note that in (844), at the same time that the inductive definition is given, it is also said that the propositional variable X and the formula A will be used to denote elements of the set being defined.

847. (844) := `grammar` for a version of LJ (the implication is the sole connective)

848. $X \in \mathcal{V}_{\mathcal{F}}$:= infinite set of `propositional variable names`

849. A, B, C := formulas

850. `named formula` := pair (formula, name)

851. Γ := set of named formulas

852. $(\Gamma \vdash A)$:= sequent of LJ

853. $(A, \text{A's name}) \notin \Gamma \;\rightarrow\; ((\Gamma, A) \;\equiv\; (\Gamma \cup \{A\}))$

854. Irrelevant formulas in axioms are admitted.

855. `Rules of LJ:`

$$\frac{}{\Gamma, A \vdash A}\,Ax \qquad \frac{\Gamma, A, A \vdash B}{\Gamma, A \vdash B}\,Cont$$

$$\frac{\Gamma \vdash A \qquad \Gamma, B \vdash C}{\Gamma, A \rightarrow B \vdash C}\,I_L \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \rightarrow B}\,I_R$$

$$\frac{\Gamma \vdash A \qquad \Gamma, A \vdash B}{\Gamma \vdash B}\,Cut$$

The Sequent Calculus LJT

856. [22, 23]

857. $(\Gamma; \vdash A)$, $(\Gamma; A \vdash A)$:= sequents of LJT

858. Γ := set of named formulas

859. `stoup` := the special place between ; and $\vdash$

860. $\exists_{\leq 1}$ formula in the stoup.

861. `Rules of LJT:`

$$\frac{}{\Gamma; A \vdash A}\,Ax \qquad \frac{\Gamma, A; A \vdash B}{\Gamma, A; \vdash B}\,Cont$$

$$\frac{\Gamma; \vdash A \qquad \Gamma; B \vdash C}{\Gamma; A \rightarrow B \vdash C}\,I_L \qquad \frac{\Gamma, A; \vdash B}{\Gamma; \vdash A \rightarrow B}\,I_R$$

862. `Head-cut rule:` (in the stoup)

$$\frac{\Gamma; \Pi \vdash A \qquad \Gamma; A \vdash B}{\Gamma; \Pi \vdash B}\ C_H$$

863. `Mid-cut rule:` (not in the stoup)

$$\frac{\Gamma; \vdash A \qquad \Gamma, A; \Pi \vdash B}{\Gamma; \Pi \vdash B}\ C_M$$

864. $X :=$ formula

865. $(\Pi = \emptyset)\ \veebar\ (\exists! X \in \Pi)$

Translation of proofs from LJ to LJT

866. $[22, 23]$

867. Irrelevant formulas in axioms are admitted.

868. In the following, $\rightsquigarrow$ means `translation` from LJ to LJT.

869.

$$\frac{}{\Gamma, A \vdash A}\ Ax \qquad \rightsquigarrow \qquad \dfrac{\dfrac{}{\Gamma, A; A \vdash A}\ Ax}{\Gamma, A; \vdash A}\ Cont$$

870. $(A \rightarrow B) \in \Gamma$

$$\frac{\begin{matrix}\vdots \\ \Gamma \vdash A\end{matrix} \qquad \begin{matrix}\vdots \\ \Gamma, B \vdash C\end{matrix}}{\Gamma \vdash C}\ I_L \qquad \rightsquigarrow \qquad \dfrac{\dfrac{\dfrac{\begin{matrix}\vdots \\ \Gamma; \vdash A\end{matrix} \qquad \dfrac{}{\Gamma; B \vdash B}\ Ax}{\Gamma; A \rightarrow B \vdash B}\ I_L}{\Gamma; \vdash B}\ Cont \qquad \begin{matrix}\vdots \\ \Gamma, B; \vdash C\end{matrix}}{\Gamma; \vdash C}\ C_M$$

871.

$$\frac{\begin{matrix}\vdots \\ \Gamma, A \vdash B\end{matrix}}{\Gamma \vdash A \rightarrow B}\ I_R \qquad \rightsquigarrow \qquad \frac{\begin{matrix}\vdots \\ \Gamma, A; \vdash B\end{matrix}}{\Gamma; \vdash A \rightarrow B}\ I_R$$

872.

$$\dfrac{\begin{array}{cc}\vdots & \vdots \\ \Gamma \vdash A & \Gamma, A \vdash B\end{array}}{\Gamma \vdash B}\ Cut \quad \rightsquigarrow \quad \dfrac{\begin{array}{cc}\vdots & \vdots \\ \Gamma; \vdash A & \Gamma, A; \vdash B\end{array}}{\Gamma; \vdash B}\ C_M$$

Proofs in Natural Deduction

873. $[19, 20]$

874. Recall the following rules for natural deduction

(i) **Hypothesis:** A

(ii) **Introductions:**

$$\dfrac{A \qquad B}{A \wedge B}\ \wedge\mathcal{I} \qquad \dfrac{\begin{matrix}[A]\\ \vdots \\ B\end{matrix}}{A \rightarrow B}\ \rightarrow\mathcal{I}x \qquad \dfrac{A}{\forall x.A}\ \forall\mathcal{I} \qquad \dfrac{A[a/x]}{\exists x.A}\ \exists\mathcal{I}$$

$$\dfrac{A}{A \vee B}\ \vee1\mathcal{I} \qquad \dfrac{B}{A \vee B}\ \vee2\mathcal{I} \qquad \dfrac{\begin{matrix}[A]\\ \vdots \\ \bot\end{matrix}}{\neg A}\ \neg\mathcal{I}$$

$$\dfrac{\begin{matrix}[A]\\ \vdots \\ B\end{matrix} \qquad \begin{matrix}[B]\\ \vdots \\ A\end{matrix}}{A \leftrightarrow B}\ \leftrightarrow\mathcal{I}$$

(iii) **Eliminations:**

$$\dfrac{A \wedge B}{A}\ \wedge1\mathcal{E} \qquad \dfrac{A \wedge B}{B}\ \wedge2\mathcal{E} \qquad \dfrac{A \rightarrow B \qquad A}{B}\ \rightarrow\mathcal{E}$$

$$\dfrac{\exists x.A \qquad \begin{matrix}[A]\\ \vdots \\ B\end{matrix}}{B}\ \exists\mathcal{E} \qquad \dfrac{\forall x.A}{A[a/x]}\ \forall\mathcal{E}$$

$$\dfrac{A \vee B \qquad \begin{matrix}[A]\\ \vdots \\ C\end{matrix} \qquad \begin{matrix}[B]\\ \vdots \\ C\end{matrix}}{C}\ \vee\mathcal{E} \qquad \dfrac{\neg A \qquad A}{\bot}\ \neg\mathcal{E}$$

$$\frac{A \leftrightarrow B \quad A}{B} \leftrightarrow \mathcal{E}1 \qquad \frac{A \leftrightarrow B \quad B}{A} \leftrightarrow \mathcal{E}2$$

(iv) **Absurdity:**

$$\frac{\begin{array}{c}[\neg A] \\ \vdots \\ \bot\end{array}}{A} \bot$$

---

875. Show that $A \to (B \to C) \vdash B \to (A \to C)$.

876. *Proof in natural deduction*

$$\dfrac{\dfrac{\dfrac{\dfrac{A \to (B \to C) \quad [A]_x}{B \to C} \to \mathcal{E} \qquad [B]_y}{C} \to \mathcal{E}}{A \to C} \to \mathcal{I}x}{B \to (A \to C)} \to \mathcal{I}y$$

877. *Proof in simply typed λ-calculus*

$$\dfrac{\dfrac{\dfrac{\dfrac{z : \alpha \to (\beta \to \gamma), x : \alpha, y : \beta \vdash z : \alpha \to (\beta \to \gamma) \quad z : \alpha \to (\beta \to \gamma), x : \alpha, y : \beta \vdash x : \alpha}{z : \alpha \to (\beta \to \gamma), x : \alpha, y : \beta \vdash zx : \beta \to \gamma} \to \mathcal{E} \qquad z : \alpha \to (\beta \to \gamma), x : \alpha, y : \beta \vdash y : \beta}{z : \alpha \to (\beta \to \gamma), x : \alpha, y : \beta \vdash (zx)y : \gamma} \to \mathcal{E}}{z : \alpha \to (\beta \to \gamma), y : \beta \vdash \lambda x.(zx)y : \alpha \to \gamma} \to \mathcal{I}x}{z : \alpha \to (\beta \to \gamma) \vdash \lambda y.\lambda x.(zx)y : \beta \to (\alpha \to \gamma)} \to \mathcal{I}y$$

878. *Proof in the natural deduction with λ-terms*

$$\dfrac{\dfrac{\dfrac{\dfrac{z : A \to (B \to C) \quad [x : A]}{zx : B \to C} \to \mathcal{E} \qquad [y : B]}{(zx)y : C} \to \mathcal{E}}{\lambda x.(zx)y : A \to C} \to \mathcal{I}x}{\lambda y.\lambda x.(zx)y : B \to (A \to C)} \to \mathcal{I}y$$

879.

$$t = \lambda y.\lambda x.(zx)y = \lambda yx.zxy$$

---

880. Derive Pierce's law: $((A \to B) \to A) \to A$.

881. *Proof in natural deduction*

882.

$$\dfrac{\dfrac{\dfrac{[\neg A]_v \qquad \dfrac{[(A \to B) \to A]_w \qquad \dfrac{\dfrac{\dfrac{[\neg A]_v \quad [A]_u}{\bot}\ \neg E}{B}\ \bot E}{A \to B}\ {\to} Iu}{A}\ {\to} E}{\bot}\ \neg E}{A}\ \text{red. abs. } v}{((A \to B) \to A) \to A}\ {\to} Iw$$

883. Show that $\forall x(A \to B) \to (\exists x A \to \exists x B)$.

884. *Proof in natural deduction*

885.

$$
\dfrac{\dfrac{\dfrac{[\exists x A]_v \qquad \dfrac{\dfrac{\dfrac{[\forall x(A \to B)]_u}{A \to B}\,\forall E \qquad [A]}{B}\,\to E}{\exists x B}\,\exists I}{\exists x B}\,\exists E}{\exists x A \to \exists x B}\,\to Iv}{\forall x(A \to B) \to (\exists x A \to \exists x B)}\,\to Iu
$$

---

886. Show that $\vdash (A \to (B \to C)) \to (A \to B) \to (A \to C)$.

887. *Proof in natural deduction*

888.

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[A \to (B \to C)]_z \qquad [A]_x}{B \to C}\,\to E \qquad \dfrac{[A \to B]_y \qquad [A]_x}{B}\,\to E}{C}\,\to E}{A \to C}\,\to I_x}{(A \to B) \to (A \to C)}\,\to I_y}{(A \to (B \to C)) \to ((A \to B) \to (A \to C))}\,\to I_z
$$

Proof of sequents in LK

889. $[3, 28]$

890. `Rules for the logical connectives:`

891.

$$\frac{\alpha, \Gamma \Rightarrow \Pi \quad \beta, \Gamma \Rightarrow \Pi}{\alpha \vee \beta, \Gamma \Rightarrow \Pi}\ (\vee\mathrm{L})$$

892.

$$\frac{\Gamma \Rightarrow \Lambda, \alpha}{\Gamma \Rightarrow \Lambda, \alpha \vee \beta}\ (\vee\mathrm{R1}) \qquad \frac{\Gamma \Rightarrow \Lambda, \beta}{\Gamma \Rightarrow \Lambda, \alpha \vee \beta}\ (\vee\mathrm{R2})$$

893.

$$\frac{\alpha, \Gamma \Rightarrow \Pi}{\alpha \wedge \beta, \Gamma \Rightarrow \Pi}\ (\wedge\mathrm{L1}) \qquad \frac{\beta, \Gamma \Rightarrow \Pi}{\alpha \wedge \beta, \Gamma \Rightarrow \Pi}\ (\wedge\mathrm{L2})$$

894.

$$\frac{\Gamma \Rightarrow \Lambda, \alpha \quad \Gamma \Rightarrow \Lambda, \beta}{\Gamma \Rightarrow \Lambda, \alpha \wedge \beta}\ (\wedge\mathrm{R})$$

895.

$$\frac{\Gamma \Rightarrow \Lambda, \alpha \quad \beta, \Delta \Rightarrow \Pi}{\alpha \rightarrow \beta, \Gamma, \Delta \Rightarrow \Lambda, \Pi}\ (\rightarrow\mathrm{L}) \qquad \frac{\alpha, \Gamma \Rightarrow \Lambda, \beta}{\Gamma \Rightarrow \Lambda, \alpha \rightarrow \beta}\ (\rightarrow\mathrm{R})$$

896.

$$\frac{\Gamma \Rightarrow \Lambda, \alpha}{\neg\alpha, \Gamma \Rightarrow \Lambda}\ (\neg\mathrm{L}) \qquad \frac{\alpha, \Gamma \Rightarrow \Lambda}{\Gamma \Rightarrow \Lambda, \neg\alpha}\ (\neg\mathrm{R})$$

897. `Cut rule:`

$$\frac{\Gamma \Rightarrow \Lambda, \alpha \quad \alpha, \Delta \Rightarrow \Pi}{\Gamma, \Delta \Rightarrow \Lambda, \Pi}\ (\mathrm{cut})$$

898. `Structural rules:`

(*i*) *exchange rules*

$$\frac{\Gamma, \alpha, \beta, \Delta \Rightarrow \Pi}{\Gamma, \beta, \alpha, \Delta \Rightarrow \Pi}\ (\mathrm{eL}) \qquad \frac{\Gamma \Rightarrow \Pi, \alpha, \beta, \Lambda}{\Gamma \Rightarrow \Pi, \beta, \alpha, \Lambda}\ (\mathrm{eR})$$

(*ii*) *contraction rules*

$$\frac{\alpha, \alpha, \Gamma \Rightarrow \Pi}{\alpha, \Gamma \Rightarrow \Pi}\ (\mathrm{cont\ L}) \qquad \frac{\Gamma \Rightarrow \Pi, \alpha, \alpha}{\Gamma \Rightarrow \Pi, \alpha}\ (\mathrm{cont\ R})$$

(*iii*) *weakening rules*

$$\frac{\Gamma \Rightarrow \Pi}{\alpha, \Gamma \Rightarrow \Pi}\ (w\mathrm{L}) \qquad \frac{\Gamma \Rightarrow \Pi}{\Gamma \Rightarrow \Pi, \alpha}\ (w\mathrm{R})$$

---

899. `Prove the following sequent in LK`

$$\Rightarrow A \to (B \to A).$$

900. *Proof*

$$\cfrac{\cfrac{\cfrac{A \Rightarrow A}{B, A \Rightarrow A}\ (w\mathrm{L})}{A \Rightarrow B \to A}\ (\to\mathrm{R})}{\Rightarrow A \to (B \to A)}\ (\to\mathrm{R})$$

---

901. `Prove the following sequent in LK`

$$A \to (B \to C) \Rightarrow (A \to B) \to (A \to C).$$

902. *Proof*

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A \Rightarrow A \qquad \dfrac{A \Rightarrow A}{B, A \Rightarrow A}\,(w\mathrm{L})}{A \to B, A, A \Rightarrow A}\,(\to\mathrm{L})}{A \to B, A \Rightarrow A}\,(\text{cont L}) \qquad \dfrac{\dfrac{A \Rightarrow A \qquad B \Rightarrow B}{A \to B, A \Rightarrow B}\,(\to\mathrm{L}) \qquad C \Rightarrow C}{B \to C, A \to B, A \Rightarrow C}\,(\to\mathrm{L})}{A \to (B \to C), A \to B, A, A \to B, A \Rightarrow C}\,(\to\mathrm{L})}{A \to (B \to C), A \to B, A \to B, A, A \Rightarrow C}\,(\mathrm{eL})}{A \to (B \to C), A \to B, A \Rightarrow C}\,(\text{cont L})}{A \to (B \to C), A \to B \Rightarrow A \to C}\,(\to\mathrm{R})}{A \to (B \to C) \Rightarrow (A \to B) \to (A \to C)}\,(\to\mathrm{R})
$$

---

903. `Prove the following sequent in LK`

$$\Rightarrow A \vee \neg A.$$

904. *Proof*

$$
\dfrac{\dfrac{\dfrac{\dfrac{A \Rightarrow A}{A \Rightarrow A \vee \neg A}\,(\vee\mathrm{R})}{\Rightarrow A \vee \neg A, \neg A}\,(\neg\mathrm{R})}{\Rightarrow A \vee \neg A, A \vee \neg A}\,(\vee\mathrm{R})}{\Rightarrow A \vee \neg A}\,(\text{cont R})
$$

905. *Proof*

$$
\dfrac{\dfrac{\dfrac{\dfrac{A \Rightarrow A}{\Rightarrow A, \neg A}\,(\neg\mathrm{R})}{\Rightarrow A \vee \neg A, A}\,(\vee\mathrm{R})}{\Rightarrow A \vee \neg A, A \vee \neg A}\,(\vee\mathrm{R})}{\Rightarrow A \vee \neg A}\,(\text{cont R})
$$

906. `Prove the following sequent in LK`

$$\neg(A \land B) \Rightarrow \neg A \lor \neg B.$$

907. *Proof*

$$
\begin{prooftree}
\AxiomC{$A \Rightarrow A$}
\doubleLine
\RightLabel{(wL,eL)}
\UnaryInfC{$A, B \Rightarrow A$}
\AxiomC{$B \Rightarrow B$}
\RightLabel{(wL)}
\UnaryInfC{$A, B \Rightarrow B$}
\RightLabel{($\land$R)}
\BinaryInfC{$A, B \Rightarrow A \land B$}
\doubleLine
\RightLabel{($\neg$L,$\neg$R,$\neg$R,eR)}
\UnaryInfC{$\neg(A \land B) \Rightarrow \neg A, \neg B$}
\doubleLine
\RightLabel{($\lor$R,$\lor$R,cont R)}
\UnaryInfC{$\neg(A \land B) \Rightarrow \neg A \lor \neg B$}
\end{prooftree}
$$

---

908. `Prove the following sequent in LK`

$$(A \to B) \to A \Rightarrow A.$$

909. *Proof*

$$
\begin{prooftree}
\AxiomC{$A \Rightarrow A$}
\RightLabel{(wR)}
\UnaryInfC{$A \Rightarrow A, B$}
\RightLabel{($\to$R)}
\UnaryInfC{$\Rightarrow A, A \to B$}
\AxiomC{$A \Rightarrow A$}
\RightLabel{($\to$L)}
\BinaryInfC{$(A \to B) \to A \Rightarrow A, A$}
\RightLabel{(cont R)}
\UnaryInfC{$(A \to B) \to A \Rightarrow A$}
\end{prooftree}
$$

---

910. `Prove the following sequent in LK`

$$A \to (B \to C) \Rightarrow B \to (A \to C).$$

911. *Proof*

$$
\begin{prooftree}
\AxiomC{$A \Rightarrow A$}
\AxiomC{$B \Rightarrow B$}
\AxiomC{$C \Rightarrow C$}
\RightLabel{($\to$L)}
\BinaryInfC{$B \to C, B \Rightarrow C$}
\RightLabel{($\to$L)}
\BinaryInfC{$A \to (B \to C), A, B \Rightarrow C$}
\doubleLine
\RightLabel{(eL,$\to$R)}
\UnaryInfC{$A \to (B \to C), B \Rightarrow A \to C$}
\doubleLine
\RightLabel{(eL,$\to$R)}
\UnaryInfC{$A \to (B \to C) \Rightarrow B \to (A \to C)$}
\end{prooftree}
$$

912. Let $\mathcal{D}_i$ $[S]$ be a `proof tree` of the *sequent* S.

913. $\mathcal{D}_1$ $[A \Rightarrow B \to A]$

$$
\begin{prooftree}
\AxiomC{$A \Rightarrow A$}
\RightLabel{wL}
\UnaryInfC{$B, A \Rightarrow A$}
\RightLabel{$\to$R}
\UnaryInfC{$A \Rightarrow B \to A$}
\end{prooftree}
$$

914. $\mathcal{D}_2$ $[A \to B, A \Rightarrow A]$

$$
\begin{prooftree}
\AxiomC{$A \Rightarrow A$}
\AxiomC{$A \Rightarrow A$}
\RightLabel{wL}
\UnaryInfC{$B, A \Rightarrow A$}
\doubleLine
\RightLabel{$\to$L,cL}
\BinaryInfC{$A \to B, A \Rightarrow A$}
\end{prooftree}
$$

915. $\mathcal{D}_3$ $[A \to B, A \Rightarrow B]$

$$
\begin{prooftree}
\AxiomC{$A \Rightarrow A$}
\AxiomC{$B \Rightarrow B$}
\RightLabel{$\to$L}
\BinaryInfC{$A \to B, A \Rightarrow B$}
\end{prooftree}
$$

916. $\mathcal{D}_4$ $[A \to (B \to C), A \to B, A \Rightarrow C]$

$$
\begin{prooftree}
\AxiomC{$\vdots\ \mathcal{D}_2$}
\noLine
\UnaryInfC{$A \to B, A \Rightarrow A$}
\AxiomC{$\vdots\ \mathcal{D}_3$}
\noLine
\UnaryInfC{$A \to B, A \Rightarrow B$}
\AxiomC{$C \Rightarrow C$}
\RightLabel{$\to$L}
\BinaryInfC{$B \to C, A \to B, A \Rightarrow C$}
\doubleLine
\RightLabel{$\to$L,cL}
\BinaryInfC{$A \to (B \to C), A \to B, A \Rightarrow C$}
\end{prooftree}
$$

917. $\mathcal{D}_5$ $[A \to (B \to C) \Rightarrow (A \to B) \to (A \to C)]$

$$
\begin{prooftree}
\AxiomC{$\vdots\ \mathcal{D}_4$}
\noLine
\UnaryInfC{$A \to (B \to C), A \to B, A \Rightarrow C$}
\RightLabel{$\to$R}
\UnaryInfC{$A \to (B \to C) \Rightarrow (A \to B) \to (A \to C)$}
\end{prooftree}
$$

918. $\mathcal{D}_6$ $[\Rightarrow A \vee \neg A]$

$$
\begin{prooftree}
\AxiomC{$A \Rightarrow A$}
\RightLabel{$\vee$R}
\UnaryInfC{$A \Rightarrow A \vee \neg A$}
\RightLabel{$\neg$R}
\UnaryInfC{$\Rightarrow A \vee \neg A, \neg A$}
\doubleLine
\RightLabel{$\vee$R, cR}
\UnaryInfC{$\Rightarrow A \vee \neg A$}
\end{prooftree}
$$

919. $\mathcal{D}_7$ $[A, B \Rightarrow A \wedge B]$

$$
\begin{prooftree}
\AxiomC{$A \Rightarrow A$}
\doubleLine
\RightLabel{\scriptsize wL,eL}
\UnaryInfC{$A, B \Rightarrow A$}
\AxiomC{$B \Rightarrow B$}
\RightLabel{\scriptsize wL}
\UnaryInfC{$A, B \Rightarrow B$}
\RightLabel{\scriptsize $\wedge$R}
\BinaryInfC{$A, B \Rightarrow A \wedge B$}
\end{prooftree}
$$

920. $\mathcal{D}_8$ $[\neg(A \wedge B) \Rightarrow \neg A \vee \neg B]$

$$
\begin{prooftree}
\AxiomC{$\vdots\ \mathcal{D}_7$}
\noLine
\UnaryInfC{$A, B \Rightarrow A \wedge B$}
\doubleLine
\RightLabel{\scriptsize $\neg$L,$\neg$R,$\neg$R,eR}
\UnaryInfC{$\neg(A \wedge B) \Rightarrow \neg A, \neg B$}
\doubleLine
\RightLabel{\scriptsize $\vee$R,$\vee$R,cR}
\UnaryInfC{$\neg(A \wedge B) \Rightarrow \neg A \vee \neg B$}
\end{prooftree}
$$

921. $\mathcal{D}_9$ $[\Rightarrow A, A \to B]$

$$
\begin{prooftree}
\AxiomC{$A \Rightarrow A$}
\RightLabel{\scriptsize wR}
\UnaryInfC{$A \Rightarrow A, B$}
\RightLabel{\scriptsize $\to$R}
\UnaryInfC{$\Rightarrow A, A \to B$}
\end{prooftree}
$$

922. $\mathcal{D}_{10}$ $[(A \to B) \to A \Rightarrow A]$

$$
\begin{prooftree}
\AxiomC{$\vdots\ \mathcal{D}_9$}
\noLine
\UnaryInfC{$\Rightarrow A, A \to B$}
\AxiomC{$A \Rightarrow A$}
\doubleLine
\RightLabel{\scriptsize $\to$L, cR}
\BinaryInfC{$(A \to B) \to A \Rightarrow A$}
\end{prooftree}
$$

923. $\mathcal{D}_{11}$ $[A \to (B \to C), A, B \Rightarrow C]$

$$
\begin{prooftree}
\AxiomC{$A \Rightarrow A$}
\AxiomC{$B \Rightarrow B$}
\AxiomC{$C \Rightarrow C$}
\RightLabel{\scriptsize $\to$L}
\BinaryInfC{$B \to C, B \Rightarrow C$}
\RightLabel{\scriptsize $\to$L}
\BinaryInfC{$A \to (B \to C), A, B \Rightarrow C$}
\end{prooftree}
$$

924. $\mathcal{D}_{12}$ $[A \to (B \to C) \Rightarrow B \to (A \to C)]$

$$\dfrac{\dfrac{\begin{matrix}\vdots\,\mathcal{D}_{11}\\ A \to (B \to C), A, B \Rightarrow C\end{matrix}}{A \to (B \to C), B \Rightarrow A \to C}\;\text{eL,}{\to}\text{R}}{A \to (B \to C) \Rightarrow B \to (A \to C)}\;\text{eL,}{\to}\text{R}$$

925. Suppose $\Rightarrow B, A$ and $A \Rightarrow B$.

$$\dfrac{\dfrac{\Rightarrow B, A \qquad A \Rightarrow B}{\Rightarrow B, B}\;\text{cut}}{\Rightarrow B}\;\text{cR}$$

926. `A-cut` is **not** *contraction-free* in LK.

$$\dfrac{\Rightarrow B, A \qquad A \Rightarrow B}{\Rightarrow B}\;\text{A-cut}$$

927. $\mathcal{D}_{13}$ $[\Rightarrow A \vee \neg A]$:= *contraction-free proof* of LEM in LK with A-cut *if A-cut is an atomic (primitive) rule.*

$$\dfrac{\dfrac{\dfrac{A \Rightarrow A}{A \Rightarrow A \vee \neg A}\;\vee\text{R}}{\Rightarrow A \vee \neg A, \neg A}\;\neg\text{R} \qquad \dfrac{\dfrac{A \Rightarrow A}{\neg A \Rightarrow \neg A}\;\neg\text{L,eL,}\neg\text{R}}{\neg A \Rightarrow A \vee \neg A}\;\vee\text{R}}{\Rightarrow A \vee \neg A}\;\text{A-cut}$$

928. Suppose $\Rightarrow A, B$.

$$\dfrac{\dfrac{\dfrac{\Rightarrow A, B}{\Rightarrow A \vee B, A}\;\vee\text{R,eR}}{\Rightarrow A \vee B, A \vee B}\;\vee\text{R}}{\Rightarrow A \vee B}\;\text{cont R}$$

929. Suppose $\Rightarrow A \vee B$.

$$\dfrac{\Rightarrow A \vee B \qquad \dfrac{\dfrac{A \Rightarrow A}{A \Rightarrow A, B}\;w\text{R} \qquad \dfrac{B \Rightarrow B}{B \Rightarrow A, B}\;w\text{R,eR}}{A \vee B \Rightarrow A, B}\;\vee\text{L}}{\Rightarrow A, B}\;\text{A-cut}$$

Proof of sequents in LJ and in LJT

930. [22]

931. Consider LJ with the sole connective $\to$.

932. `Rules of LJ:`

$$\frac{}{\Gamma, A \vdash A}\,{\scriptstyle Ax} \qquad\qquad \frac{\Gamma, A, A \vdash B}{\Gamma, A \vdash B}\,{\scriptstyle Cont}$$

$$\frac{\Gamma \vdash A \qquad \Gamma, B \vdash C}{\Gamma, A \to B \vdash C}\,{\scriptstyle I_L} \qquad\qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\,{\scriptstyle I_R}$$

$$\frac{\Gamma \vdash A \qquad \Gamma, A \vdash B}{\Gamma \vdash B}\,{\scriptstyle Cut}$$

---

933. `Prove the following sequent in LJ`

$$\vdash (A \to (B \to C)) \to ((A \to B) \to (A \to C)).$$

934. *Proof in* `LJ`

$$\frac{\dfrac{\dfrac{\dfrac{}{A, A \to B \vdash A}\,{\scriptstyle Ax} \qquad \dfrac{\dfrac{\dfrac{}{A \vdash A}\,{\scriptstyle Ax} \qquad \dfrac{}{A, B \vdash B}\,{\scriptstyle Ax}}{A, A \to B \vdash B}\,{\scriptstyle I_L} \qquad \dfrac{}{A, A \to B, C \vdash C}\,{\scriptstyle Ax}}{A, A \to B, B \to C \vdash C}\,{\scriptstyle I_L}}{A, A \to B, A \to (B \to C) \vdash C}\,{\scriptstyle I_L}}{A \to B, A \to (B \to C) \vdash A \to C}\,{\scriptstyle I_R}}{\dfrac{A \to (B \to C) \vdash (A \to B) \to (A \to C)}{\vdash (A \to (B \to C)) \to ((A \to B) \to (A \to C))}\,{\scriptstyle I_R}}\,{\scriptstyle I_R}$$

935. `Prove the following sequent in LJT`

$$\vdash (A \to (B \to C)) \to ((A \to B) \to (A \to C)).$$

936. `Rules of LJT (without cut):`

$$\frac{}{\Gamma; A \vdash A}\ {\scriptstyle Ax} \qquad\qquad \frac{\Gamma, A; A \vdash B}{\Gamma, A; \vdash B}\ {\scriptstyle Cont}$$

$$\frac{\Gamma; \vdash A \qquad \Gamma; B \vdash C}{\Gamma; A \to B \vdash C}\ {\scriptstyle I_L} \qquad\qquad \frac{\Gamma, A; \vdash B}{\Gamma; \vdash A \to B}\ {\scriptstyle I_R}$$

937. *Proof in* `LJT`

$$\begin{prooftree}
\AxiomC{}
\RightLabel{D_1}
\UnaryInfC{$A, A \to B; \vdash A$}
\AxiomC{}
\RightLabel{D_1}
\UnaryInfC{$A; \vdash A$}
\AxiomC{}
\RightLabel{Ax}
\UnaryInfC{$A; B \vdash B$}
\RightLabel{I_L}
\BinaryInfC{$A; A \to B \vdash B$}
\RightLabel{Der}
\UnaryInfC{$A, A \to B; \vdash B$}
\AxiomC{}
\RightLabel{Ax}
\UnaryInfC{$A, A \to B; C \vdash C$}
\RightLabel{I_L}
\BinaryInfC{$A, A \to B; B \to C \vdash C$}
\RightLabel{I_L}
\BinaryInfC{$A, A \to B; A \to (B \to C) \vdash C$}
\RightLabel{Der}
\UnaryInfC{$A, A \to B, A \to (B \to C); \vdash C$}
\RightLabel{I_R}
\UnaryInfC{$A \to B, A \to (B \to C); \vdash A \to C$}
\RightLabel{I_R}
\UnaryInfC{$A \to (B \to C); \vdash (A \to B) \to (A \to C)$}
\RightLabel{I_R}
\UnaryInfC{$\vdash (A \to (B \to C)) \to ((A \to B) \to (A \to C))$}
\end{prooftree}$$

938. $D_1 :=$

$$\frac{\dfrac{}{A; A \vdash A}\ {\scriptstyle Ax}}{A; \vdash A}\ {\scriptstyle Cont}$$

939.

$$\frac{\Gamma; A \vdash B}{\Gamma, A; \vdash B}\ {\scriptstyle Der}$$

940. $Der :=$

$$\frac{\dfrac{\Gamma; A \vdash B}{\Gamma, A; A \vdash B}\ \text{adding irrelevant formula}}{\Gamma, A; \vdash B}\ {\scriptstyle Cont}$$

Open Invitation

Review, *add* content, and `co-author` this white paper [24, 25].
Join the **Open Mathematics Collaboration**.
Send your contribution to `mplobo@uft.edu.br`.

Open Science

The **latex file** for this *white paper* together with other *supplementary files* are available in [26, 27].

How to cite this paper?

https://doi.org/10.31219/osf.io/8wck9
https://zenodo.org/record/5594984

Acknowledgements

+ **Center for Open Science**
  https://cos.io
+ **Open Science Framework**
  https://osf.io
+ **Zenodo**
  https://zenodo.org

Agreement

The author agrees with [25].

License

References

[1] Leary, Christopher C., and Lars Kristiansen. *A friendly introduction to mathematical logic*, 2nd edition, 2015. `https://knightscholar.geneseo.edu/geneseo-authors/6/`

[2] CC. Creative Commons. *CC-By Attribution 4.0 International.* `https://creativecommons.org/licenses/by/4.0`

[3] Ono, Hiroakira. *Proof Theory and Algebra in Logic*. Singapore: Springer, 2019.

[4] Gabbay, Dov M., and Franz Guenthner, eds. *Handbook of Philosophical Logic*. Vol. 1. Dordrecht: Kluwer Academic Publishers, 2001.

[5] Lobo, Matheus P. "Hilbert-style Proof Calculus for Propositional Logic in ABC Notation." *OSF Preprints*, 25 Nov. 2019. `https://doi.org/10.31219/osf.io/jd3gp`

[6] Buss, Samuel R. "An introduction to proof theory." *Handbook of Proof Theory* 137 (1998): 1-78. `https://bit.ly/3BnmMrk`

[7] Laurent, Olivier. "Focusing in orthologic." *Logical Methods in Computer Science* 13 (2017). `https://lmcs.episciences.org/3808/pdf`

[8] Warner, Steve. *Real Analysis for Beginners*. GET 800, 2020.

[9] De Swart, Harrie. *Philosophical and Mathematical Logic*. Springer International Publishing, 2018.

[10] Nishimura, Iwao. "On formulas of one variable in intuitionistic propositional calculus." *The Journal of Symbolic Logic* 25.4 (1960): 327-331. https://doi.org/10.2307/2963526

[11] Dalla Chiara, Maria Luisa, and Roberto Giuntini. "Quantum logics." *Handbook of Philosophical Logic*. Springer, Dordrecht, 2002. 129-228. https://arxiv.org/pdf/quant-ph/0101028.pdf

[12] WolframAlpha. https://www.wolframalpha.com

[13] Birkhoff, Garrett, and John Von Neumann. "The logic of quantum mechanics." *Annals of Mathematics* (1936): 823-843.

[14] Rasga, João, and Cristina Sernadas. *Decidability of Logical Theories and Their Combination*. Springer International Publishing, 2020.

[15] Wikipedia. "Undecidable problem." https://en.wikipedia.org/wiki/Undecidable_problem

[16] Wikipedia. "Decision problem." https://en.wikipedia.org/wiki/Decision_problem

[17] Wikipedia. "Word problem." https://en.wikipedia.org/wiki/Word_problem_(mathematics)

[18] Sørensen, Morten Heine, and Pawel Urzyczyn. *Lectures on the Curry-Howard Isomorphism*. Elsevier, 2006.

[19] Girard, Jean-Yves, Paul Taylor, and Yves Lafont. *Proofs and types*. Cambridge: Cambridge university press, 1989.

[20] Wikipedia. Dedução Natural. https://pt.wikipedia.org/wiki/Deducao_natural

[21] Hindley, J. Roger, and Jonathan P. Seldin. *Lambda-calculus and Combinators, an Introduction*. Cambridge: Cambridge University Press, 2008.

[22] Herbelin, Hugo. "A λ-calculus structure isomorphic to Gentzen-style sequent calculus structure." *International Workshop on Computer Science Logic*. Springer, Berlin, Heidelberg, 1994.

[23] Girard, Jean-Yves. "A new constructive logic: classic logic." *Mathematical structures in computer science* 1.3 (1991): 255-296.

[24] Lobo, Matheus P. "Microarticles." *OSF Preprints*, 28 Oct. 2019. https://doi.org/10.31219/osf.io/ejrct

[25] Lobo, Matheus P. "Simple Guidelines for Authors: Open Journal of Mathematics and Physics." *OSF Preprints*, 15 Nov. 2019. https://doi.org/10.31219/osf.io/fk836

[26] Lobo, Matheus P. "Open Journal of Mathematics and Physics (OJMP)." *OSF*, 21 Apr. 2020. https://osf.io/6hzyp/files

[27] https://zenodo.org/record/5594984

[28] Indrzejczak, Andrzej. *Sequents and Trees*. Springer International Publishing, 2021. https://doi.org/10.1007/978-3-030-57145-0

The Open Mathematics Collaboration

Matheus Pereira Lobo[1,2,3] (lead author, mplobo@uft.edu.br)
https://orcid.org/0000-0003-4554-1372

[1]Federal University of Tocantins (Brazil)
[2]Federal University of Northern Tocantins (Brazil)
[3]Universidade Aberta (UAb, Portugal)

APPENDIX

Quantum Logics: Introduction

941. [11, 12]

942. *What logical structures one may hope to find in physical theories which, like quantum mechanics, do not conform to classical logic?* [13]

943. Phase-space is a mathematical concept present in both classical and quantum theories.

944. $\mathcal{S}$:= physical system

945. Σ := phase-space

946. a `point` in Σ := the "state" of $\mathcal{S}$ (ascertainable by "maximal" observations)

947. `pure states` := maximal pieces of information about $\mathcal{S}$
(cannot be consistently extended to a richer knowledge)

948. `mixtures` := non maximal pieces of information

949. P := experimental proposition about $\mathcal{S}$

950. X := all the pure states for which P holds

951. $X \subseteq \Sigma$

952. `events` (*physical qualities*) := subsets of Σ

953. X := event

954. $\mathcal{P}$:= set of all *experimental propositions*

955. $\mathcal{E}$:= set of all *events*

956. The correspondence between $\mathcal{P}$ and $\mathcal{E}$ is many-to-one.

957. $p :=$ pure state

958.

$$(\mathcal{S} \text{ in state } p \texttt{ verifies} \text{ both } X \text{ and } P) \equiv (p \in X)$$

959. *What is the structure of all events?*

960. The power-set of any set is a Boolean algebra.

961.

$$\mathcal{B} = \langle \mathcal{F}(\Sigma), \subseteq, \cap, \cup, -, \mathbf{1}, \mathbf{0} \rangle$$

962. $\mathcal{B} :=$ Boolean algebra

963. $\mathcal{F}(\Sigma) :=$ set of all *measurable events*

964. $\subseteq :=$ set-theoretic inclusion relation

965. $\cap :=$ intersection of sets ("and")

966. $\cup :=$ union of sets ("or")

967. $- :=$ relative complement of a set ("not")

968. $\mathbf{1} := \Sigma$ (total space)

969. $\mathbf{0} := \emptyset$ (empty space)

970. `Classical semantic behaviour:`

$$\begin{aligned} (i) &\ (p \text{ verifies } X \cap Y) \leftrightarrow (p \in X \cap Y) \leftrightarrow (p \text{ verifies both members}) \\ (ii) &\ (p \text{ verifies } X \cup Y) \leftrightarrow (p \in X \cup Y) \leftrightarrow (p \text{ verifies at least one member}) \\ (iii) &\ (p \text{ verifies } -X) \leftrightarrow (p \notin X) \leftrightarrow (p \text{ does not verify } X) \end{aligned}$$

971. points of $\Sigma :=$ wave-functions

972. $\Sigma \equiv$ *function-space* (usually the Hibert space)

973. In classical mechanics, the *excluded middle principle* holds, i.e.,

$$p \in X \ \veebar \ p \notin X.$$

974. Quantum theory is essentially probabilistic.

975. $\psi :=$ pure state (wave function) of a quantum system

976. In a quantum system, the experimental proposition P, for instance, can be "the spin value in a certain direction is up".

977. We have the following cases for the assignment of probability-values:

   (i) $\psi(P) = 1$, P is true,

   (ii) $\psi(P) = 0$, P is false,

   (iii) $\psi(P) \neq 0, 1$, P is *semantically indetermined.*

978. *Which mathematical representative would best describe quantum experimental propositions?*

979. `closed subspace` $:=$ *closed linear subspace of Hilbert space* $:=$ mathematical representative of P in a quantum system

980. `complete metric` $:=$ *metric* in which *every Cauchy sequence is convergent*

981.

$$\texttt{Hilbert space } (\mathcal{H}) := \text{vector space over a division ring}$$
$$(h \in \mathcal{H} \ \rightarrow \ h \in \mathbb{R} \ \vee \ h \in \mathbb{C} \ \vee \ h \in \mathbb{H}) \text{ such that}$$

   (i) an *inner product* is defined,

   (ii) $\mathcal{H}$ is *metrically complete.*

982. $\mathbb{H} :=$ set of quaternion numbers

983.

$$(\mathcal{H} := \texttt{separable}) \ \leftrightarrow \ (\mathcal{H} \text{ admits a countable basis})$$

984. Hereafter, let

$$\mathcal{H} := \text{separable Hilbert space}$$

such that its *unitary vectors* correspond to wave functions of a quantum system.

985. `closed subspaces` of $\mathcal{H}$:= subsets of $\mathcal{H}$ (closed under *linear combinations* and *Cauchy sequences*)

986. (985) contains the mathematical representatives of experimental propositions that are closed under finite and infinite linear combinations.

987. `quantum events` := mathematical representatives of experimental propositions of a quantum system

988. quantum mechanics $\rightsquigarrow$ linear combinations of p $\rightsquigarrow$ new pure states

989. $C(\mathcal{H})$:= set of all quantum events

990. `negation` of a *quantum event* := **orthogonal complement** of the *event*

991. `orthogonal complement` of a subspace V of the vector space := set of vectors orthogonal to all elements of V

992. X, X', Y := quantum events (closed subspaces)

993. X' := orthogonal complement of X

994. $X, X', Y \subseteq \mathcal{H}$

995. $\psi \in X' \;\leftrightarrow\; \psi \perp X \;\leftrightarrow\; \forall \phi \in X : (\psi, \phi) = 0$

996. (ψ, ϕ) := inner product of ψ and ϕ

997. `orthocomplement` := *orthogonal complement*

998.

$$\forall X\ \forall \psi\,(\text{pure states}) : \psi(X) = 1 \;\leftrightarrow\; \psi(X') = 0$$

999.
$$\forall X\ \forall\psi\ (\text{pure states}) : \psi(X) = 0\ \leftrightarrow\ \psi(X') = 1$$

1000.
$$\psi \text{ verifies } X \cap Y\ \leftrightarrow\ \psi \text{ verifies both members}$$

1001. union of two closed subspaces $\not\equiv$ closed subspace

1002. `supremum` $\sim\succ$ connective ***or***

1003. $X \sqcup Y$:= *supremum* of X and Y (the smallest closed subspace including both closed subspaces X and Y)

1004. $X \cup Y \subset X \sqcup Y$

1005.
$$\mathcal{C}(\mathcal{H}) = \langle C(\mathcal{H}), \sqsubseteq, \sqcap, \sqcup, ', \mathbf{1}, \mathbf{0} \rangle$$

1006. $\sqsubseteq, \sqcap$:= set-theoretic inclusion and intersection

1007. $\sqcup$:= supremum

1008. $'$:= orthogonal complement

1009. $\mathbf{1} := \mathcal{H}$ (total space)

1010. $\mathbf{0}$:= null subspace [the singleton of the null vector (smallest subspace)]

1011. `projections` := *idempotent* and *self-adjoint* linear operators

1012. $\mathfrak{P}(\mathcal{H})$:= set of all projections P of $\mathcal{H}$

1013. $\cong$:= isomorphism

1014. $\mathfrak{P}(\mathcal{H}) \cong$ closed subspaces

1015. $\mathcal{C}(\mathcal{H})$ is <u>not</u> a Boolean algebra, it simulates a "quasi-Boolean behaviour".

1016. $\mathcal{C}(\mathcal{H})$ is a (not distributive) *orthocomplemented orthomodular lattice*,

$$X \sqcap (Y \sqcup Z) \neq (X \sqcap Y) \sqcup (X \sqcap Z).$$

1017. $X \sqcup Y$ **may be true** *even if neither member is true.*

1018. It is possible for a pure state ψ that

$$\psi \notin X \wedge \psi \notin Y \ \rightarrow \ \psi \in X \sqcup Y.$$

1019. (1016) is connected with (1018) (the superposition principle).

1020. uncertainty principle $\rightsquigarrow$ *incompatible* quantities $\rightsquigarrow$ strongly undetermined (cannot be simultaneously measured)

1021. `standard quantum logic` := (complete orthomodular lattice + closed subspaces in $\mathcal{H}$) $\rightsquigarrow$ particular example of an algebraic structure