Logic of Proof Assistants

Prof. Floris van Doorn *

University of Bonn

4th May, 2025

Contents

---

* LaTeX-realization by Hannah Scholz

1 Introduction

The topics of this class are:

(i) First-order Logic/Set Theory

(ii) Lambda Calculus

(iii) Simple Type Theory (Higher-Order Logic)

(iv) Dependent Type Theory/Homotopy Type Theory

Example 1.1. Here are examples of proof assistants for these different types of logics:

(i) First-order Logic/Set Theory: Mizar, Metamath

(ii) Simple Type Theory: Isabelle/HoL, HoL Light

(iii) Dependent Type Theory: Lean, Rocq (formerly Coq), Agda

(iv) Homotopy Type Theory: cubicaltt, rzk

Remark 1.2. You might want to have the following criteria for a logic:

(i) Appropriate (You can encode mathematical arguments.)

(ii) Simple (It is relatively easy to understand.)

(iii) Expressive (Mathematical arguments are convenient to express.)

Theorem 1.3. *Let* π *be the prime counting function, i.e.* $\pi\colon \mathbb{R} \to \mathbb{N}$, $x \mapsto |\{p \leq x \mid p \text{ prime}\}|$. *Then* $\lim_{x\to\infty} \frac{\pi(x)}{x/\log(x)} = 1$.

Remark 1.4. When formalizing/stating this theorem in a formal logic there are a few things that you need to think about:

(i) What do you do about division by zero?

(ii) What does division even mean? (Do you define division for $\mathbb{R}$ explicitly? Do you define it generally for a field? Or even for a group? How do you ensure that the "correct" field structure on $\mathbb{R}$ gets used?)

(iii) How do you define a limit? (Do you define a limit for $\mathbb{R}$ explicitly? Or for every topological space? How do you ensure the "correct" topology on $\mathbb{R}$ gets used? How do you deal with potentially non-unique limits (for example in non-Hausdorff spaces)?)

Remark 1.5. You can make the following design choices for "a logic":

(i) Is the logic typed or untyped?

(ii) Is the logic constructive or classical?

(iii) Does the logic support computation?

Remark 1.6. In logic there is the **object language** and we reason about it in a **meta-language** ("ordinary mathematical reasoning").

1.1 Inductive Definitions

Example 1.7. The natural numbers are inductively defined by $0 \in \mathbb{N}$ and $S\colon \mathbb{N} \to \mathbb{N}$, $n \mapsto n+1$.

Definition 1.8. Let U be a set and $\mathcal{C} \subseteq \bigcup_{n\in\mathbb{N}}(U^n \to U)$ a set of **constructors**. Where $c\colon U^n \to U$ is called an **n-ary function** and $(U^n \to U)$ is the collection on n-ary functions.

(i) $A \subseteq U$ is **closed under $\mathcal{C}$** if for any n-ary $c \in \mathcal{C}$ and for all $x_1, \dots, x_n \in A$ we have that $c(x_1, \dots, x_n) \in A$.

(ii) $A \subseteq U$ is **generated by $\mathcal{C}$** or **inductively defined by $\mathcal{C}$** if A is the smallest set that is closed under $\mathcal{C}$, i.e. $A = \bigcap\{B \subseteq U \mid B \text{ is closed under } \mathcal{C}\}$.

(iii) $A \subset U$ is **freely generated by $\mathcal{C}$** if

(a) each constructor is injective on A and

(b) the images of different constructors are disjoint.

Remark 1.9. $\varnothing$ is closed under $\mathcal{C}$ iff $\mathcal{C}$ has no nullary constructors.

Exercise 1.10. $\bigcap\{B \subseteq U \mid B \text{ is closed under } \mathcal{C}\}$ *is closed under* $\mathcal{C}$.

Example 1.11.

(i) The free group.

(ii) The σ-algebra generated by a collection of subsets. (This is not freely generated.)

(iii) The topology generated by a collection of subsets. (This is not freely generated.)

Theorem 1.12 (Structural Induction)**.** *If $A \subset U$ is generated by $\mathcal{C}$ and $P \colon A \to \{\top, \bot\}$ is a predicate on A, to prove $\forall a \in A, P(a)$ it suffices to show: for any n-ary $c \in \mathcal{C}$ and any $x_1, \ldots, x_n \in A$ if $P(x_1), \ldots, P(x_n)$ then $P(c(x_1, \ldots, x_n))$.*

Proof. Exercise. □

Remark 1.13. The base case of the induction is given by nullary constructors.

Theorem 1.14 (Structural Recursion)**.** *If $A \subset U$ is freely generated by $\mathcal{C}$, B is a set and for any n-ary $c \in \mathcal{C}$ we have a $g_c \colon B^n \to B$ then there is a unique function $f \colon A \to B$ such that $f(c(a_1, \ldots, a_n)) = g_c(f(a_1), \ldots, f(a_n))$ for every $c \in \mathcal{C}$ and $a_1, \ldots a_n \in A$.*

Proof. Exercise. □

Example 1.15. For $A = \mathbb{N}$ this reduces to $f(0) \coloneqq g_0$ and $f(S(n)) \coloneqq g_s(f(n))$.

2 First-Order Logic

Definition 2.1. A (first-order) **language** $\mathcal{L}$ is a triple $(\mathcal{F}, \mathcal{R}, a)$ where $\mathcal{F}$ is a set of function symbols, $\mathcal{R}$ is a set of relation symbols, $\mathcal{F}$ and $\mathcal{R}$ are disjoint and $a\colon \mathcal{F} \cup \mathcal{R} \to \mathbb{N}$ is the arity function.

Example 2.2. A language for groups $\mathcal{L}_{\text{Group}}$ has $\mathcal{F} \coloneqq \{\cdot, {}^{-1}, 1\}$, $\mathcal{R} \coloneqq \varnothing$, $a(\cdot) = 2$, $a({}^{-1}) = 1$ and $a(1) = 0$.

Definition 2.3. We fix an infinite set of **variables** $\mathcal{V} \coloneqq \{x_0, x_1, \dots\}$.

Remark 2.4. We use x for variables, f and g for functions and R and S for relations.

Definition 2.5. We can define the **terms** $T_{\mathcal{L}}$ in the language $\mathcal{L}$ using the **Backus–Naur form (BNF)** :

$$s, t \Coloneqq x \mid f(t_1, \dots, t_n)$$

where f is an n-ary function symbol.

Definition 2.6. Formally, we define the **terms** $T_{\mathcal{L}}$ in the language $\mathcal{L}$ in the following way. We define the set of **symbols** $S \coloneqq \mathcal{F} \mathbin{\dot\cup} \mathcal{V} \mathbin{\dot\cup} \{\text{“(”}, \text{“)”}, \text{“,”}\}$ and the set of finite sequences of symbols S^*. Let $\mathcal{C}$ be defined as:

(i) for each variable $x \in \mathcal{V}$ there is a nullary constructor $c_x \coloneqq x$

(ii) for each n-ary function symbol f there is an n-ary constructor $c_f\colon (S^*)^n \to S^*$, $c_f(t_1, \dots, t_n) \coloneqq f\text{“(”}t_1\text{“,”}\dots\text{“,”}t_n\text{“)”}$

Then $T_{\mathcal{L}} \subseteq S^*$ is the set generated by $\mathcal{C}$.

Example 2.7.

(i) “(”“)”“,”f is in S^* but not in $T_{\mathcal{L}}$.

(ii) If f is binary then $f\text{“(”}x_0\text{“,”}x_1\text{“)”}$ is in $T_{\mathcal{L}}$.

Remark 2.8. Technically, the brackets and commas are not necessary. They are however necessary when you use infix notation. (For example the meaning of $a \cdot b + c$ is unclear.)

Definition 2.9. First-order **formulas** $\Phi_{\mathcal{L}}$ are specified by

$$\varphi, \psi ::= \bot \mid s = t \mid R(t_1, \dots t_n) \mid (\varphi \wedge \psi) \mid (\varphi \vee \psi) \mid (\varphi \rightarrow \psi) \mid (\forall x.\varphi) \mid (\exists x.\varphi)$$

where $\mathcal{R}$ is an n-ary relation symbol and $t_1, \dots, t_n \in T_{\mathcal{L}}$.

Remark 2.10. In classical logic one could omit the rules $(\varphi \wedge \psi)$ and $(\varphi \vee \psi)$ (as they can be defined using the other rules). They are however necessary for constructive logic.

Remark 2.11. We can define other connectives:

(i) $\neg\varphi \coloneqq (\varphi \rightarrow \bot)$

(ii) $\varphi \leftrightarrow \psi \coloneqq ((\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi))$

Remark 2.12. When writing formulas we omit some parentheses:

(i) $\varphi \rightarrow \psi \rightarrow \theta$ means $\varphi \rightarrow (\psi \rightarrow \theta)$

(ii) $\forall x.\varphi \rightarrow \psi$ means $\forall x.(\varphi \rightarrow \psi)$

Remark 2.13. We want $\forall x.x = x$ and $\forall y.y = y$ to mean the same thing. Options to achieve this are:

(i) Define $(\forall x.x = x) \equiv_\alpha (\forall y.y = y)$ to be **α-equivalent**. And then define the set of formulas to be $\Phi_{\mathcal{L}} / \equiv_\alpha$.

(ii) We could not use variable names for bound variables and use **de Bruijn indices** instead.

Remark 2.14. $\forall x.x = y$ has **bound variables** $\{x\}$ and **free variables** $\{y\}$. For a formula φ or a term t we also write **fv**(φ) and **fv**(t) for the set of free variables in φ and t.

Definition 2.15. A **sentence** is a formula without free variables.

Definition 2.16. **Substitution** $s[t/x]$ of x by t in a term s is defined recursively by

(i) $y[t/x] \coloneqq \begin{cases} t & \text{if } y = x \\ y & \text{otherwise} \end{cases}$

(ii) $f(s_1, \dots, s_n)[t/x] \coloneqq f(s_1[t/x], \dots, s_n[t/x])$

Example 2.17. Defining substitution in formulas is a little bit harder as we need to avoid **variable capture**: $(\exists x.x \leq z)[(x+1)/z]$ should not be $\exists x.x \leq x+1$ but $\exists y.y \leq x+1$.

Definition 2.18. For a formula φ **substitution** $\varphi[t/x]$ is defined as:

(i) $s = s'[t/x] \coloneqq s[t/x] = s'[t/x]$

(ii) $R(t_1, \dots, t_n)[t/x] \coloneqq R(t_1[t/x], \dots, t_n[t(x)])$

(iii) $(\varphi \vee \psi)[t/x] \coloneqq \varphi[t/x] \vee \psi[t/x]$

(iv) $(\varphi \wedge \psi)[t/x] \coloneqq \varphi[t/x] \wedge \psi[t/x]$

(v) $(\varphi \to \psi)[t/x] \coloneqq \varphi[t/x] \to \psi[t/x]$

(vi) $(\forall y.\varphi) \coloneqq \begin{cases} \forall y.\varphi & \text{if } y = x \\ \forall z.\varphi[z/y][t/x] & \text{otherwise} \end{cases}$

(vii) $(\exists y.\varphi) \coloneqq \begin{cases} \exists y.\varphi & \text{if } y = x \\ \exists z.\varphi[z/y][t/x] & \text{otherwise} \end{cases}$

Definition 2.19. **α-equivalence** is the **congruence closure** of

(i) $(\forall x.\varphi) \equiv_\alpha (\forall y.\varphi[y/x])$

(ii) $(\exists x.\varphi) \equiv_\alpha (\exists y.\varphi[y/x])$

i.e. it is the smallest equivalence relation containing these two rules and respecting the connectives:

(i) $(\varphi_1 \wedge \varphi_2) \equiv_\alpha (\psi_1 \wedge \psi_2)$ for $\varphi_1 \equiv_\alpha \psi_1$ and $\varphi_2 \equiv_\alpha \psi_2$

(ii) $(\varphi_1 \vee \varphi_2) \equiv_\alpha (\psi_1 \vee \psi_2)$ for $\varphi_1 \equiv_\alpha \psi_1$ and $\varphi_2 \equiv_\alpha \psi_2$

(iii) $(\varphi_1 \to \varphi_2) \equiv_\alpha (\psi_1 \to \psi_2)$ for $\varphi_1 \equiv_\alpha \psi_1$ and $\varphi_2 \equiv_\alpha \psi_2$

(iv) $(\forall x.\varphi) \equiv_\alpha (\forall x.\psi)$ if $\varphi \equiv_\alpha \psi$

(v) $(\exists x.\varphi) \equiv_\alpha (\exists x.\psi)$ if $\varphi \equiv_\alpha \psi$

Remark 2.20. We will treat α-equivalence as an equivalence relation. You could also define the formulas as $\Phi_{\mathcal{L}}/\equiv_\alpha$ and thus treat α-equivalence as equality.

2.1 Provability

Definition 2.21. Let Γ be a set of formulas and φ a formula. Then $\boldsymbol{\Gamma} \vdash \varphi$ (read : "Γ proves φ") is defined inductively by

(i) $\dfrac{}{\Gamma, \varphi \vdash \varphi}$ (assumption rule)

(ii) $\dfrac{\Gamma \vdash \varphi \qquad \Gamma \vdash \psi}{\Gamma \vdash \varphi \land \psi}$ ($\land$-introduction)

(iii) $\dfrac{\Gamma \vdash \varphi_1 \land \varphi_2}{\Gamma \vdash \varphi_i}$ for $i = 1, 2$ ($\land$-elimination)

(iv) $\dfrac{\Gamma \vdash \varphi_i}{\Gamma \vdash \varphi_1 \lor \varphi_2}$ for $i = 1, 2$ ($\lor$-introduction)

(v) $\dfrac{\Gamma \vdash \varphi \lor \psi \qquad \Gamma, \varphi \vdash \theta \qquad \Gamma, \psi \vdash \theta}{\Gamma \vdash \theta}$ ($\lor$-elimination)

(vi) $\dfrac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \to \psi}$ ($\to$-introduction)

(vii) $\dfrac{\Gamma \vdash \varphi \to \psi \qquad \Gamma \vdash \varphi}{\Gamma \vdash \psi}$ ($\to$-elimination)

(viii) $\dfrac{\Gamma, \neg\varphi \vdash \bot}{\Gamma \vdash \varphi}$ (proof by contradiction)

(ix) $\dfrac{\Gamma \vdash \varphi}{\Gamma \vdash \forall x.\varphi}$ for $x \notin \mathrm{fv}(\Gamma)$ ($\forall$-introduction)

(x) $\dfrac{\Gamma \vdash \forall x.\varphi}{\Gamma \vdash \varphi[t/x]}$ ($\forall$-elimination)

(xi) $\dfrac{\Gamma \vdash \varphi[t/x]}{\Gamma \vdash \exists x.\varphi}$ ($\exists$-introduction)

(xii) $\dfrac{\Gamma \vdash \exists x.\varphi \qquad \Gamma, \varphi \vdash \psi}{\Gamma \vdash \psi}$ for $x \in \mathrm{fv}(\Gamma, \psi)$ ($\exists$-elimination)

(xiii) $\dfrac{}{\Gamma \vdash t = t}$ ($=$-introduction)

(xiv) $\dfrac{\Gamma \vdash s = t \qquad \Gamma \vdash \varphi[t/x]}{\Gamma \vdash \varphi[s/x]}$ ($=$-elimination)

(xv) $\dfrac{\Gamma \vdash \varphi}{\Gamma \vdash \psi}$ for $\varphi \equiv_\alpha \psi$ (α-equivalence)

Remark 2.22. Read " $\dfrac{\mathrm{A}}{\mathrm{B}}$ " as: "Under the assumptions A we can prove B". With "Γ, φ" we really mean $\Gamma \cup \{\varphi\}$.

Example 2.23. If φ and ψ are formulas then we can show $\vdash (\varphi \wedge \psi) \to (\psi \wedge \varphi)$ using the following **proof tree** :

$$\dfrac{\dfrac{\dfrac{}{\varphi \wedge \psi \vdash \varphi \wedge \psi}\text{Assump.}}{\varphi \wedge \psi \vdash \psi}\wedge\text{-elim.} \qquad \dfrac{\dfrac{}{\varphi \wedge \psi \vdash \varphi \wedge \psi}\text{Assump.}}{\varphi \wedge \psi \vdash \varphi}\wedge\text{-elim.}}{\dfrac{\varphi \wedge \psi \vdash \psi \wedge \varphi}{\vdash (\varphi \wedge \psi) \to (\psi \wedge \varphi)}\to\text{-intro.}}\wedge\text{-intro.}$$

2.2 Semantics

Definition 2.24. An **$\mathcal{L}$-structure** $\mathcal{M}$ consists of

(i) a non-empty set $|\mathcal{M}|$

(ii) for any n-ary function symbol f a function $f_{\mathcal{M}}\colon |\mathcal{M}|^n \to |\mathcal{M}|$

(iii) for any n-ary relation symbol R a set $R_{\mathcal{M}} \subseteq |\mathcal{M}|^n$

Definition 2.25. If t is an $\mathcal{L}$-term and $\sigma\colon \mathcal{V} \to |\mathcal{M}|$ we define $[\![t]\!]_{\mathcal{M},\sigma}$ as:

(i) $[\![x]\!]_{\mathcal{M},\sigma} \coloneqq \sigma(x)$

(ii) $[\![f(t_1,\ldots,t_n)]\!]_{\mathcal{M},\sigma} \coloneqq f_{\mathcal{M}}([\![t_1]\!]_{\mathcal{M},\sigma},\ldots,[\![t_n]\!]_{\mathcal{M},\sigma})$

For formulas we define $\mathcal{M} \models_\sigma \varphi$ holds as

(i) $\mathcal{M} \models_\sigma R(t_1,\ldots,t_n)$ iff $R_{\mathcal{M}}([\![t_1]\!]_{\mathcal{M},\sigma},\ldots,[\![t_n]\!]_{\mathcal{M},\sigma})$

(ii) $\mathcal{M} \models_\sigma \perp$ never holds

(iii) $\mathcal{M} \models_\sigma s = t$ iff $[\![s]\!]_{\mathcal{M},\sigma} = [\![t]\!]_{\mathcal{M},\sigma}$

(iv) $\mathcal{M} \models_\sigma \varphi \wedge \psi$ iff $\mathcal{M} \models_\sigma \varphi$ and $\mathcal{M} \models_\sigma \psi$

(v) $\mathcal{M} \models_\sigma \varphi \vee \psi$ iff $\mathcal{M} \models_\sigma \varphi$ or $\mathcal{M} \models_\sigma \psi$

(vi) $\mathcal{M} \models_\sigma \varphi \to \psi$ iff $\mathcal{M} \models_\sigma \varphi$ implies $\mathcal{M} \models_\sigma \psi$

(vii) $\mathcal{M} \models_\sigma \forall x.\varphi$ iff for all $a \in |\mathcal{M}|$ we know that $\mathcal{M} \models_{\sigma,\mathrm{x}\mapsto\mathrm{a}} \varphi$ where

$$(\sigma, x \mapsto a)(y) \coloneqq \begin{cases} a & y = x \\ \sigma(y) & \text{otherwise} \end{cases}$$

(viii) $\mathcal{M} \models_\sigma \exists x.\varphi$ iff there is $a \in |\mathcal{M}|$ such that $\mathcal{M} \models_{\sigma,\mathrm{x}\mapsto\mathrm{a}} \varphi$

Remark 2.26. We write $\varphi(\bar{x})$ to mean that $\mathrm{fv}(\varphi) \subseteq \bar{x}$ and $\varphi(\bar{t})$ for $\varphi[\bar{t}/\bar{x}]$.

Remark 2.27. $[\![t]\!]_{\mathcal{M},\sigma}$ and $\mathcal{M} \models_\sigma \varphi$ only depend on the values $\sigma(x)$ where $x \in \mathrm{fv}(t)$ and $x \in \mathrm{fv}(\varphi)$ respectively. If φ is a sentence then $\mathcal{M} \models_\sigma \varphi$ does not depend on σ and is denoted $\mathcal{M} \models \varphi$ (read: "$\mathcal{M}$ realizes φ").

Definition 2.28. If Γ is a set of formulas and φ is a formula then $\Gamma \models \varphi$ means that for any $\mathcal{L}$-structure $\mathcal{M}$ and assignment $\sigma\colon \mathcal{V} \to |\mathcal{M}|$ such that $\mathcal{M} \models_\sigma \psi$ for all $\psi \in \Gamma$ we have $\mathcal{M} \models_\sigma \varphi$.

Theorem 2.29 (Soundness theorem)**.** *If* $\Gamma \vdash \varphi$ *then* $\Gamma \models \varphi$.

Theorem 2.30 (Completeness theorem)**.** *If* $\Gamma \models \varphi$ *then* $\Gamma \vdash \varphi$.

Theorem 2.31 (Compactness theorem)**.** *If* $\Gamma \models \varphi$ *then for some finite* $\Gamma' \subseteq \Gamma$ *we have* $\Gamma' \models \varphi$.

2.3 Definite descriptions

Definition 2.32. $\exists! x.\varphi(x, \bar{z}) \coloneqq \exists x.(\varphi(x, \bar{z}) \wedge \forall y.\varphi(y, \bar{z}) \to y = x)$.

Definition 2.33. Suppose Γ is a set of $\mathcal{L}$-sentences, Γ' a set of $\mathcal{L}'$-sentences and $\mathcal{L} \subseteq \mathcal{L}'$. Then **$\Gamma'$ is conservative over Γ** if $\Gamma \subseteq \Gamma'$ and for all $\mathcal{L}$-formulas ψ such that $\Gamma' \vdash \psi$ we have $\Gamma \vdash \psi$.

Theorem 2.34. *Suppose that* $\Gamma \vdash \forall \bar{x}.\exists! \varphi(\bar{x}, y)$ *and that* f *is a fresh function symbol (i.e. not among the function symbols of* $\mathcal{L}$*) then* $\Gamma \cup \{\forall \bar{x}.\varphi(\bar{x}, f(\bar{x}))\}$ *is conservative over* Γ.

Definition 2.35 (Axioms of ZFC)**.** $\mathcal{L}_{ZFC}$ has no function symbol and one binary relation "$\in$". The axioms of ZFC are

(i) Extensionality : $\forall x \forall y.(\forall z.z \in x \leftrightarrow z \in y) \to x = y$

(ii) Pairing: $\forall x \forall y \exists z \forall w.w \in z \leftrightarrow w = x \vee w = y$ ("$z = \{x, y\}$")

This allows us to define $\{x\} \coloneqq \{x, x\}$.

(iii) Union: $\forall x \exists y \forall z. z \in y \leftrightarrow \exists w.(w \in x \land z \in w)$ ("$y = \bigcup x$")

This allows us to define $x \cup y \coloneqq \bigcup\{x, y\}$.

(iv) Power set: $\forall x \exists y \forall z. z \in y \leftrightarrow w \in x$ ("$y = \mathcal{P}(x)$")

(v) Separation (axiom schema): for any formula $\varphi(\bar{x}, y)$ we have $\forall \bar{x} \forall y \exists z \forall w. w \in z \leftrightarrow (w \in y \land \varphi(\bar{x}, w))$ ("$z = \{w \in y \mid \varphi(\bar{x}, w)\}$")

(vi) Infinity: $\exists x. \varnothing \in x \land \forall y. y \in x \rightarrow y \cup \{y\} \in x$ where $\varnothing \coloneqq \{w \in y \mid \bot\}$

(vii) Foundation: $\forall x.(\exists y. y \in x) \rightarrow \exists y. y \in x \land \forall z. z \in x \rightarrow z \notin y$ ("Every set x contains an element y disjoint from x")

(viii) Replacement (axiom schema): For any formula $\varphi(z, w, \bar{y})$ we have $\forall x \forall \bar{y}(\forall z. z \in x \rightarrow \exists! w \varphi(z, w, \bar{y})) \rightarrow \exists u \forall w. w \in u \leftrightarrow \exists z. z \in x \land \varphi(z, w, \bar{y})$ ("If φ is a function with domain x then the image of φ is a set.")

(ix) Choice: $\forall x. \varnothing \notin x \rightarrow \exists f. f \in (x \rightarrow \bigcup x) \land \forall y. y \in x \rightarrow f(y) \in y$

where we define $(x, y) \coloneqq \{\{x\}, \{x, y\}\}$,

$A \times B \coloneqq \{z \in \mathcal{P}(\mathcal{P}(A \cup B)) \mid \exists x \in A \exists y \in B. z = (x, y)\}$,

$(A \rightarrow B) \coloneqq \{f \in \mathcal{P}(A \times B) \mid \forall x \in A \exists! y.(x, y) \in f\}$ and

$$f(x) \coloneqq \begin{cases} y & \text{if } (x, y) \in f \\ \varnothing & \text{if no such } y \text{ exists} \end{cases}$$

Remark 2.36. The existence of at least one set is provable and therefore the empty set also exists. Nonetheless, the existence of the empty set is often added as an axiom.