FIRST PRINCIPLES OF
MATHEMATICS

Forrest C. Taylor, M.S.

ABSTRACT

A dependent type theory is proposed as the foundation of mathematics. The formalism preserves the structure of mathematical thought, making it natural to use. The logical calculus of the type theory is proved to be syntactically complete. Therefore it does not suffer from the limitations imposed by Gödel’s incompleteness theorems. In particular, the concept of mathematical truth can be defined in terms of provability.
COPYRIGHT

Copyright © 2023 by Forrest C. Taylor under a Creative Commons BY-NC-ND 4.0 International license. To view a copy of this license, visit

http://creativecommons.org/licenses/by-nc-nd/4.0/

or send a letter to Creative Commons at

PO Box 1866
Mountain View, CA 94042, USA.

To summarize the license, you are free to copy and redistribute the material in any medium or format, under the following terms:

1  Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

2  Non-commercial — You may not use the material for commercial purposes.

3  No derivatives — If you remix, transform, or build upon the material, you may not distribute the modified material.

You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.
# TABLE OF CONTENTS

## FIRST PRINCIPLES

<table>
<thead>
<tr>
<th>Letter</th>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>the foundation of mathematics</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>the syntax of mathematics</td>
<td>4</td>
</tr>
<tr>
<td>C</td>
<td>the language of mathematics</td>
<td>10</td>
</tr>
<tr>
<td>D</td>
<td>the use of variables</td>
<td>16</td>
</tr>
</tbody>
</table>

## FORMALIZATION OF MATHEMATICS

<table>
<thead>
<tr>
<th>Number</th>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>mathematical operations</td>
<td>23</td>
</tr>
<tr>
<td>2</td>
<td>natural numbers</td>
<td>31</td>
</tr>
<tr>
<td>3</td>
<td>values of an operation</td>
<td>35</td>
</tr>
<tr>
<td>4</td>
<td>assemblies</td>
<td>39</td>
</tr>
<tr>
<td>5</td>
<td>mathematical propositions</td>
<td>43</td>
</tr>
<tr>
<td>6</td>
<td>predicate logic</td>
<td>48</td>
</tr>
<tr>
<td>7</td>
<td>mathematical induction</td>
<td>55</td>
</tr>
<tr>
<td>8</td>
<td>finite sequences</td>
<td>61</td>
</tr>
<tr>
<td>9</td>
<td>binary quantification</td>
<td>66</td>
</tr>
<tr>
<td>10</td>
<td>constructing subtypes</td>
<td>71</td>
</tr>
<tr>
<td>11</td>
<td>negative propositions</td>
<td>79</td>
</tr>
<tr>
<td>12</td>
<td>equating mathematical objects</td>
<td>85</td>
</tr>
</tbody>
</table>

## CONCLUDING REMARKS

<table>
<thead>
<tr>
<th>Letter</th>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>mathematical structures</td>
<td>95</td>
</tr>
<tr>
<td>F</td>
<td>philosophical implications</td>
<td>97</td>
</tr>
</tbody>
</table>
This book is logically self-contained. No previous knowledge of mathematics or logic is assumed. The reader will only need the English language and the ability to recognize patterns. The only exceptions are the abstract and the preface. In both cases, the reader is assumed to be familiar with the existing literature on the foundations of mathematics.

WHY I WROTE THIS BOOK

As a student, I looked everywhere for a book that explained mathematics from first principles: a modern version of Euclid's *Elements*. I never found one. The reason for this can be summarized by a diagram of the logical dependencies in modern textbooks:

![Diagram showing the vicious circle in modern textbooks.](image)

There is a vicious circle. The naive concept of a “set” leads to contradictions, as demonstrated by Russell’s paradox. The purpose of axiomatic set theory is to restrict the concept of “set,” and thereby resolve the contradictions in naive set theory. All of the modern books on axiomatic set theory assume knowledge of first order logic. Unfortunately, all of the modern books on first order logic use naive set theory for their semantics, completing the vicious circle.

There are a few older books on metamathematics, such as Bourbaki [1970] and Kleene [1952], that avoid the vicious circle. However, all such books use unstated assumptions in their metamathematical proofs, believing them to be...
“obvious” or “intuitively true.” When Euclid uses unstated assumptions in the *Elements*, we criticize him for not being rigorous. But he could say exactly the same thing to us. And I doubt that Euclid could tolerate the vicious circle.

Recent efforts to formalize mathematics in digital libraries are not immune to this criticism. The software is designed using mathematical principles, and these principles must be justified outside of the software.

In order to understand mathematics, we must start at the beginning. This book is my attempt to do so. I have aimed for clarity and beauty above all else. Please feel free to contact me with any criticisms.

Oak Harbor, Washington

Forrest C. Taylor

April 2023
PART I

FIRST PRINCIPLES
THE FOUNDATION OF MATHEMATICS

MATHEMATICAL ASSERTIONS

DEFINITION / assertion

• If $x$ is a symbol and $\delta$ is a definition, then the statement

  "$x$ satisfies $\delta$"

is defined to be an assertion.

• If the statements $\alpha$ and $\beta$ are assertions, then the statements

  "$\alpha$ and $\beta$" and "If $\alpha$, then $\beta$"

are defined to be assertions.

REMARK Assertions may be rephrased in transparent ways, as illustrated by the following examples.

EXAMPLE The definition of a number is given in the next chapter. If $n$ is a symbol, then the statement

• "$n$ satisfies the definition of a number"

is an assertion. It can be rephrased as "$n$ is a number."

EXAMPLE It is often convenient to separate an assertion into a sequence of statements, as described in table 1. Let $\alpha$, $\beta$, and $\gamma$ be assertions.

<table>
<thead>
<tr>
<th>SEQUENCE OF STATEMENTS</th>
<th>INTERPRETATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Suppose that $\alpha$; Then $\beta$</td>
<td>If $\alpha$, then $\beta$</td>
</tr>
<tr>
<td>Suppose that $\alpha$; If $\beta$, then $\gamma$</td>
<td>If $\alpha$ and $\beta$, then $\gamma$</td>
</tr>
</tbody>
</table>

Table 1. Interpretation of sequences of assertions.
DEFINITION  Let \( \alpha, \beta, \) and \( \gamma \) be assertions. The statements defined in table 2 abbreviate sequences of assertions.

<table>
<thead>
<tr>
<th>SEQUENCE OF ASSERTIONS</th>
<th>ABBREVIATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>If ( \alpha ), then ( \beta ); If ( \beta ), then ( \alpha )</td>
<td>Then ( \alpha ) if and only if ( \beta )</td>
</tr>
<tr>
<td>If ( \alpha ), then ( \gamma ); If ( \gamma ), then ( \alpha )</td>
<td>If ( \alpha ) or ( \beta ), then ( \gamma )</td>
</tr>
</tbody>
</table>

TABLE 2. Abbreviations for sequences of assertions.

THE FIRST PRINCIPLES OF MATHEMATICS

INTUITION  Let \( x \) denote a symbol, \( \delta \) a definition, and \( \alpha \) and \( \beta \) assertions. The meanings of the assertions

“\( x \) satisfies \( \delta \),” “\( \alpha \) and \( \beta \),” and “If \( \alpha \), then \( \beta \)”

elude definition. Instead, we state the rules for using mathematical assertions to construct proofs.

A.1 / DEFINITION APPLICATION

Let \( x \) be a symbol and \( \delta \) a definition.

1. From “\( x \) is defined to satisfy \( \delta \),” we may conclude that “\( x \) satisfies \( \delta \).”
2. From “If \( \alpha \), then \( x \) is defined to satisfy \( \delta \),” we may conclude that “If \( \alpha \), then \( x \) satisfies \( \delta \).”

A.2 / CONJUNCTION INTRODUCTION

Let \( \alpha, \beta, \) and \( \gamma \) be assertions.

1. From “\( \beta \)” and “\( \gamma \),” we may conclude “\( \beta \) and \( \gamma \).”
2. From “If \( \alpha \), then \( \beta \)” and “If \( \alpha \), then \( \gamma \),” we may conclude that “If \( \alpha \), then \( \beta \) and \( \gamma \).”

A.3 / CONJUNCTION ELIMINATION

Let \( \alpha \) and \( \beta \) be assertions.

1. We may conclude that “If \( \alpha \) and \( \beta \), then \( \alpha \).”
2. We may conclude that “If \( \alpha \) and \( \beta \), then \( \beta \).”
A.4 / IMPLICATION INTRODUCTION
Let $x$ be a symbol, $\delta$ a definition, and $\alpha$ an assertion. We may conclude that “If $\alpha$, then $\alpha$.”

A.5 / IMPLICATION ELIMINATION
Let $\alpha$ and $\beta$ be assertions. From “$\alpha$” and “If $\alpha$, then $\beta$,” we may conclude “$\beta$.”

A.6 / HYPOTHETICAL SYLLOGISM
Let $\alpha$, $\beta$, and $\gamma$ be assertions. From “If $\alpha$, then $\beta$” and “If $\beta$, then $\gamma$,” we may conclude that “If $\alpha$, then $\gamma$.”

DEFINITION Principles A.1–A.6 are the first principles of mathematics.

THEOREMS AND PROOFS

DEFINITION An assertion is said to be true if it has been concluded using the first principles of mathematics. True assertions are called theorems.

INTUITION Theorems express truth by definition. A proof of a theorem is an explanation of why it is true.

REMARK It is not necessary to cite the first principles in proofs, because it is not difficult to understand how they have been used.

INTUITION A theorem is called:

- a corollary of a given theorem if it requires little or no additional proof
- a lemma if it is useful, but not interesting in its own right.

REMARK Writing the symbol $\Box$ against the right-hand margin indicates the end of a proof or the omission of a trivial proof.

HOW TO READ THIS BOOK Think of each theorem as an exercise, and each proof as a series of hints. Strive to complete all of the exercises using as few hints as possible. This is the most enjoyable way to learn mathematics, and it guarantees a deep understanding.
THE SYNTAX OF MATHEMATICS

FUNDAMENTAL CONCEPTS

DEFINITION / number

- The symbol 0 is defined to be a number. It is called zero.
- If \( n \) is a number, then the symbol \( s(n) \) is defined to be a number. It is called the successor of \( n \).

EXAMPLE The symbols \( s(0) \) and \( s(s(0)) \) are numbers.

DEFINITION Let \( x \) and \( y \) be symbols. The symbol

\[
x := y
\]

means that \( x \) denotes \( y \). In other words, \( x \) is a name for \( y \), or represents \( y \).

NOTATION If \( n \) is a number, then \( 0n := n \). If \( x \) is a symbol and \( 0x \) denotes a number, then \( x := 0x \).

NOTATION Table 3 defines the Hindu-Arabic notation for numbers, where \( n \) is a number and \( x \) is a symbol:

<table>
<thead>
<tr>
<th>( n )</th>
<th>( s(n) )</th>
<th>( n )</th>
<th>( s(n) )</th>
<th>( n )</th>
<th>( s(n) )</th>
<th>( n )</th>
<th>( s(n) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x0 )</td>
<td>( x1 )</td>
<td>( x3 )</td>
<td>( x4 )</td>
<td>( x6 )</td>
<td>( x7 )</td>
<td>( x9 )</td>
<td>( s(x)0 )</td>
</tr>
<tr>
<td>( x1 )</td>
<td>( x2 )</td>
<td>( x4 )</td>
<td>( x5 )</td>
<td>( x7 )</td>
<td>( x8 )</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( x2 )</td>
<td>( x3 )</td>
<td>( x5 )</td>
<td>( x6 )</td>
<td>( x8 )</td>
<td>( x9 )</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TABLE 3. Hindu-Arabic notation.

If the left-hand symbol denotes \( n \), then the right-hand symbol denotes \( s(n) \). The reader is assumed to be familiar with this notation, and with the cardinal and ordinal words for numbers.
EXAMPLE The successor of zero is called one. Since zero is denoted by 00, it follows that one is denoted by 01, and therefore by 1.

REMARK Notation is immaterial to the logical structure of mathematics. Its purpose is to streamline the use of symbols.

DEFINITION Let \( n \) be a number. The six symbols

\[ \lor \quad b \quad c \quad s \quad x \quad \land(x(n)) \]

are defined to be prefixes. The first five are called simple prefixes.

INTUITION By the end of chapter 1, the meaning of each prefix will be clear. In other words, it will be clear how each prefix is used in mathematics.

DEFINITION / term
Let \( \alpha \) be a prefix.

- Numbers are defined to be terms.
- If \( t \) is a term, then the symbol \( \alpha(t) \) is defined to be a term.
- If \( t \) and \( u \) are terms, then the symbol \( t(u) \) is defined to be a term.

INTUITION Some terms have meaning, while some do not. Meaningful terms are called mathematical objects. The next chapter gives the precise definition.

DEFINITION If \( n \) is a number, then \( x_n \) is a term. It is called the variable with index \( n \). The symbol \( \land_x \) denotes the prefix \( \land(x) \), where \( x \) is understood to be a variable.

EXAMPLE If \( t, u, \) and \( v \) are terms, then \( t(u)(v) \) and \( t(u(v)) \) are terms.

DEFINITION / list of terms

- If \( t \) is a term, then \( t \) is defined to be a list of terms.
- If \( t \) is a term and \( \mathcal{L} \) is a list of terms, then the symbol \( \mathcal{L}, t \) is defined to be a list of terms.

NOTATION Let \( t \) be a term, \( \mathcal{L} \) a list of terms, and \( \beta \) a prefix or a term. Then

\[ \beta(\mathcal{L}, t) := \beta(\mathcal{L})(t). \]

The symbol \( \beta_{\mathcal{L}} := \beta(\mathcal{L}) \) is called the application of \( \beta \) to \( \mathcal{L} \), or simply \( \beta \) of \( \mathcal{L} \).
example  Let $t$, $u$, and $v$ be terms. Each of the four symbols
\[
t(u,v) \quad t_u(v) \quad t(u)_v \quad t_{u,v}
\]
denotes $t(u)(v)$.

**B.1 / Lemma**

Let $t$ be a term, $\mathcal{L}$ a list of terms, and $\beta$ a prefix or a term. If $\beta(\mathcal{L})$ is a term, then $\beta(\mathcal{L}, t)$ is a term.

**Variables and Contexts**

**Definition / Context / Length of a Context**

Let $t$ be a term.

- The symbol $\bullet$ is defined to be a context of length zero. It is called the empty context.
- The symbol $x_0 : t$ is defined to be a context of length one.
- If $n$ is a number and $\Gamma$ is a context of length $s(n)$, then the symbol $\Gamma, x_{s(n)} : t$
  is defined to be a context of length $s(s(n))$.

**Intuition**  In type theory, a type of mathematical object, such as integer or function, is itself a mathematical object, and therefore a term.

**Intuition**  A context declares its variables to be objects of certain types, by labeling them with the corresponding terms. In other words, contexts assign meanings to variables.

**Intuition**  The variable $x_n$ only has meaning in a context of length $s(n)$. It may have different meanings in different contexts.

**Example**  Let $t$, $u$, and $v$ be terms. The symbol

\[(B.2) \quad x_0 : t, \ x_1 : u(x_0), \ x_2 : v(x_0, x_1)\]

is a context of length three.

**Notation**  Let $\Gamma$ be a context and $t$ a term. In the symbol

\[\Gamma, x : t\]
it is understood that $x$ is the variable with index $n$, where $n$ is the length of $\Gamma$. If $\Gamma$ is the empty context, then $\Gamma, x : t$ denotes the context $x : t$.

**DEFINITION / labeling of variables**

Let $y$ be a variable, $t$ and $u$ terms, and $\Gamma$ a context.

- The context $\Gamma, x : t$ is defined to label $x$ with $t$.
- If $\Gamma$ labels $y$ with $u$, then $\Gamma, x : t$ is defined to label $y$ with $u$.

**INTUITION** The context $\Gamma, x : t$ assigns the same labels as $\Gamma$, and then labels $x$ with $t$. The empty context does not label any variables.

**EXAMPLE** The context $(B.2)$ labels $x_0$ with $t$ and $x_1$ with $u(x_0)$.

**INTUITION** Let $x$ be a variable. If the context $\Gamma$ labels $x$ with the type $X$, then $x$ is an indeterminate object of type $X$ in the context $\Gamma$.

**JUDGMENTS AND TEXTS**

**DEFINITION** Let $t$ and $u$ be terms and $\Gamma$ a context. The symbols

\[
\Gamma \xrightarrow{\text{ctx}} \Gamma \vdash t : u \quad t \equiv u
\]

are called *judgments*. They are denoted by (or interpreted as) statements:

<table>
<thead>
<tr>
<th>JUDGMENT</th>
<th>INTERPRETATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\Gamma \xrightarrow{\text{ctx}}$</td>
<td>The context $\Gamma$ is well-formed</td>
</tr>
<tr>
<td>$\Gamma \vdash t : u$</td>
<td>The label $u$ applies to $t$ in the context $\Gamma$</td>
</tr>
<tr>
<td>$t \equiv u$</td>
<td>The term $t$ is substitutable for $u$</td>
</tr>
</tbody>
</table>

**Table 4.** Interpretation of judgments.

**DEFINITION / text**

- Judgments are defined to be *texts*.
- If $H$ and $K$ are texts, then the symbol

\[
H \quad K
\]

is defined to be a text, called the *conjunction of $H$ and $K$*. This text is interpreted as the statement $H$ and $K$. 
The conjunction of $H$ and $K$ is denoted by $H \cdot K$. This notation is used if $H$ and $K$ are assumed, rather than proved, to be texts.

**Example** Let $m$ be a number, $X$ and $Y$ terms, and $\Gamma$ a context. The text

$$\Gamma \text{ ctx } \Gamma \vdash X : c_0(m) \quad \Gamma, x : X \vdash Y : c_0(m)$$

will be important in chapter 1. Its interpretation is provided in chapter B.

**Definition / subtext**

Let $H$, $K$, and $L$ be texts.

- The text $H$ is defined to be a subtext of itself,
- If $H$ is a subtext of $K$ or $L$, then $H$ is defined to be a subtext of $K \cdot L$.

**Intuition** The text $H$ is a subtext of $K$ if every judgment that occurs in $H$ occurs in $K$.

**B.3 / Lemma**

Let $H$ and $K$ be texts. Then $H$ is a subtext of $H \cdot K$ and $K \cdot H$. \hfill \Box

**Inferences and Truth**

**Definition** For texts $H$ and $K$, the inference from $H$ to $K$ is the symbol

$$\frac{H}{K}$$

where $H$ is its hypothesis and $K$ its result. Inferences 0.1–0.41 are called the postulates of mathematics.

**Notation** The inference from $H$ to $K$ is denoted by $H/K$. This notation is used when $H$ and $K$ are assumed, rather than proved, to be texts.

**Definition / valid inference**

Let $H$, $K$, and $L$ be texts and $M$ the text $K \cdot L$.

- If $K$ is a subtext of $H$, then $H/K$ is defined to be valid.
- If $H/K$ is a postulate, then $H/K$ is defined be valid.
- If $H/K$ and $K/L$ are valid, then $H/L$ is defined to be valid.
• If $H/K$ and $H/L$ are valid, then $H/M$ is defined to be valid.

**Definition** Let $H$ and $K$ be texts. If the inference $H/K$ is valid, then $K$ is said to be *derivable from* $H$.

**Definition** The judgment $\bullet \text{ctx}$ is called the *axiom of mathematics*. The text $H$ is said to be *valid* if it is derivable from $\bullet \text{ctx}$.

**Intuition** Mathematics is ultimately concerned with *valid judgments*. The axiom and postulates constitute the *first principles of the type theory*.

**B.4 / Theorem**

Let $H$ and $K$ be texts. If $H$ is valid and $H/K$ is valid, then $K$ is valid.

**B.5 / Theorem**

Let $H$, $K$, and $L$ be texts. Suppose that $K$ is derivable from $H$.

1. If $H$ is a subtext of $L$, then $K$ is derivable from $L$.
2. If $L$ is a subtext of $K$, then $L$ is derivable from $H$.

**Intuition** The hypothesis of a valid inference can be *strengthened*, and its conclusion can be *weakened*.

**B.6 / Corollary**

Let $H$, $K$, and $L$ be texts. Then $K \cdot L$ is derivable from $H$ if and only if both $K$ and $L$ are derivable from $H$.

**Proof** By B.3 and B.5.

**B.7 / Corollary**

The texts $H$ and $K$ are both valid if and only if $H \bullet K$ is valid.

**B.8 / Lemma**

Let $H$, $K$, and $L$ be texts. Then $K \bullet L$ is derivable from $H$ if and only if $L \bullet K$ is derivable from $H$.

**Proof** By B.6.

**Remark** The next chapter provides a method for expressing valid inferences using natural language. Refer to postulates 0.1–0.4, 0.7, 0.12–0.14, and 0.16 for examples.
IMPLICIT HYPOTHESES

**Definition** This book is divided into *entries*, which are labeled using small capitals. An entry may include *sub-entries* (such as *proofs* of theorems), which are labeled using italics.

**Definition** Let $K$ and $L$ be texts. The statements

- $K$ implies $L$
- Assume that $K$
- Conclude that $L$

are called *hypothetical statements*.

**Definition / implicit hypothesis**

Let $H$, $K$, and $L$ be texts and $S$ a hypothetical statement in the entry $E$.

- If $S$ is the first hypothetical statement in $E$, then $\Gamma$ ctx is defined to be the *implicit hypothesis* of $S$ in $E$, where $\Gamma$ is an arbitrary context.
- Otherwise, let $S'$ be the hypothetical statement directly before $S$ in $E$.
- If $H$ is the *implicit hypothesis* of $S'$ in $E$ and $S'$ is the statement
  - Assume that $K$
  then the text $H \cdot K$ is defined to be the *implicit hypothesis* of $S$ in $E$.
- If $H \cdot K$ is the *implicit hypothesis* of $S'$ in $E$ and $S'$ is the statement
  - Conclude that $L$
  where the statement “$K$ implies $L$” precedes $S'$ in $E$, then $H$ is defined to be the *implicit hypothesis* of $S$ in $E$.
- Otherwise, the *implicit hypothesis* of $S'$ in $E$ is defined to be the *implicit hypothesis* of $S$ in $E$. 
INTUITION The assertion that $K$ implies $L$ is proved in two steps:

1. Assume that $K$. This adds $K$ to the implicit hypothesis.
2. Conclude that $L$. This removes $K$ from the implicit hypothesis.

The implicit hypothesis does not change unless an assumption is made or an implication is proved.

DEFINITION Let $H$, $K$, and $L$ be texts. In table 5, let each statement $S$ in the left-hand column have the implicit hypothesis $H$ in the entry $E$.

<table>
<thead>
<tr>
<th>STATEMENT</th>
<th>INTERPRETATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conclude that $L$</td>
<td>$L$ is derivable from $H$</td>
</tr>
<tr>
<td>$K$ implies $L$</td>
<td>$L$ is derivable from $H \cdot K$</td>
</tr>
</tbody>
</table>

Table 5. The meanings of hypothetical statements.

Then the interpretation of $S$ in $E$ is given in the right-hand column.

NOTATION Let $X$ be a statement and $S$ a hypothetical statement. If these statements have the same meaning in ordinary language, then $X$ denotes $S$.

EXAMPLE Let $K$ and $L$ be texts.

- The statements “let $K$” and “suppose that $K$” mean “assume that $K$.”
- The statements “then $L$” and “therefore $L$” mean “conclude that $L$.”

The statement “$K$ implies $L$” is denoted by

$L$ if $K$ • $L$ is necessary for $K$ • If $K$, then $L$ • $K$ is sufficient for $L$.

DEFINITION Let $H$ and $L$ be texts. If the inference $H/L$ is a postulate, then $L$ is said to be derivable from $H$ by postulate.

NOTATION Let $H$ and $L$ be texts. The statement

• It is postulated that $L$

means “It follows that $L$ by postulate.” In other words, $H/L$ is defined to be a postulate, where $H$ is the implicit hypothesis.

NOTATION Let $H$, $K$, and $L$ be texts. The statement

• Then $K$ if and only if $L$

means “$K$ implies $L$ and $L$ implies $K$.” In other words, $K$ is derivable from $H \cdot K$ and $L$ is derivable from $H \cdot L$, where $H$ is the implicit hypothesis.
Implicit Contexts

Definition Let $x$ be a variable and $t$, $u$, and $v$ terms. The statements

- The implicit context is well-formed
- The label $u$ applies to $t$
- Assign the label $v$ to $x$

are called contextual statements.

Definition / bound variable / depth of a bound variable
Let $k$ be a number, $t$ and $u$ terms, and $\beta$ a simple prefix or a term.

- The variable $x$ is defined to be bound with depth zero in $\wedge_x(t)$
- If the variable $y$ is bound with depth $k$ in $t$, then $y$ is defined to be bound with depth $s(k)$ in $\wedge_x(t)$, and bound with depth $k$ in $\beta(t)$ and $t(u)$.

Definition The symbol $\Gamma$ is called the fixed context. For the rest of the book, $\Gamma$ denotes a context.

Definition / implicit context
Let $t$, $u$, and $v$ be terms and $\Delta$ a context of length $n$. Let $S$ be a contextual statement in the entry $E$.

- If $S$ is the first contextual statement of $E$, then $\Gamma$ is defined to be the implicit context of $S$ in $E$.
- Otherwise, let $S'$ be the contextual statement directly before $S$ in $E$.
- If $\Delta$ is the implicit context of $S'$ in $E$ and $S'$ is the statement
  - Assign the label $t$ to $x$
  where $x$ is the variable with index $n$, then the context $\Delta, x : t$ is defined to be the implicit context of $S$ in $E$.
- If $\Delta, x : t$ is the implicit context of $S'$ in $E$ and $S$ states that
  - The label $v$ applies to $u$,
  where $x$ is bound with depth zero in $u$ and/or $v$, then $\Delta$ is defined to be the implicit context of $S$ in $E$.
- Otherwise, the implicit context of $S'$ in $E$ is defined to be the implicit context of $S$ in $E$. 
The language of mathematics

**Definition**  Let \( t, u, \) and \( v \) be terms. In table 6, let each statement \( S \) in the left-hand column have the implicit context \( \Delta \) in the entry \( E \).

<table>
<thead>
<tr>
<th>Statement</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>The implicit context is well-formed</td>
<td>The context ( \Delta ) is well-formed</td>
</tr>
<tr>
<td>The label ( u ) applies to ( t )</td>
<td>The label ( u ) applies to ( t ) in the context ( \Delta )</td>
</tr>
</tbody>
</table>

*Table 6. The meanings of contextual statements.*

Then the interpretation of \( S \) in \( E \) is given in the right-hand column.

**Types and mathematical objects**

**Definition**  Let \( n \) be a number. The term \( c_n \) is called the *constructor with index* \( n \). The symbol \( U \) denotes the constructor with index zero.

**Intuition**  *Constructors* are used to distinguish different *types of mathematical object*, in order to manipulate them by different rules.

**Definition**  Let \( m \) be a number and \( X \) a term. The term \( U(m) \) is denoted by \( U_m \) and called the *type universe of order* \( m \). The statement that

- \( X \) is a type of order \( m \)

means that the label \( U_m \) applies to \( X \).

**Definition**  Let \( a \) be a term and \( X \) a type of order \( m \). The statements

- \( a \) is an object of type \( X \)
- \( a \) has type \( X \)

mean that the label \( X \) applies to \( a \), written \( a : X \).

**Definition**  Let \( X \) be a type of order \( m \) and \( S \) the statement

- Declare \( x \) as an object of type \( X \).

Then \( S \) means “assign the label \( X \) to \( x \).” If the implicit context of \( S \) in the entry \( E \) has length \( n \), then \( x \) denotes the variable with index \( n \) in \( E \).

**Intuition**  If the variable \( x \) has been declared as an object of type \( X \), then \( x \) represents an *indeterminate object of type* \( X \).
0.1 / ACCUMULATION OF TYPE UNIVERSES

Let $m$ be a number. It is postulated that:

1. The universe of order $m$ is a type of order $s(m)$
2. If $X$ is a type of order $m$, then $X$ is a type of order $s(m)$.

Remark In other words, the following inferences are defined to be postulates:

$$
\frac{\Gamma \text{ctx}}{\Gamma \vdash \mathbb{U}_m : \mathbb{U}_{s(m)}} \quad \frac{\Gamma \vdash X : \mathbb{U}_m}{\Gamma \vdash X : \mathbb{U}_{s(m)}}
$$

intuition Type universes are objects of higher-order type universes. Types accumulate in higher-order type universes.

definition The symbol $\nu$ is called the fixed number. For the rest of the book, $\nu$ denotes a number.

- The phrase “of order $\nu$” is suppressed. For example, the type universe of order $\nu$ is simply called the type universe. It is denoted by $\mathbb{U}$.
- The phrase “of order $s(\nu)$” is replaced with the phrase of higher order.

example Let $X$ be a term. The meaning of each statement in the left-hand column of table 7 is given in the right-hand column.

<table>
<thead>
<tr>
<th>STATEMENT</th>
<th>MEANING</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X$ is a type</td>
<td>$X$ is a type of order $\nu$</td>
</tr>
<tr>
<td>$X$ is a higher-order type</td>
<td>$X$ is a type of order $s(\nu)$</td>
</tr>
</tbody>
</table>

Table 7. Use of the fixed number.

If the term $a$ is assumed to be an object of type $X$, then $X$ is understood to be a type of order $\nu$ unless otherwise specified.

example Let $H$ and $K$ be texts, $\Delta$ a context, and $S$ the statement

"If $K$, then $a$ has type $X."$

If $S$ has implicit hypothesis $H$ and implicit context $\Delta$ in the entry $E$, then $S$ means that the judgment

$$
\Delta \vdash a : X
$$

is derivable from $H \cdot K$. 
Let $X$ and $Y$ be types. Assume that the variable $x$ is bound with depth zero in the term $y$. The implicit context can be changed:

- from $\Delta$ to $\Delta$, $x : X$ by declaring $x$ as an object of type $X$.
- from $\Delta$, $x : X$ to $\Delta$ by proving that $y$ is an object of type $Y$.

The implicit context does not change unless a variable is declared or a declared variable becomes bound.
THE USE OF VARIABLES

DECLARATION OF VARIABLES

DEFINITION Let $\Delta$ be a context. If the judgment $\Delta$ ctx is true, then $\Delta$ is said to be well-formed.

**D.1 / Lemma**
The empty context is well-formed. \hfill $\square$

**Intuition** Let $\Delta$ be a well-formed context which assigns the label $X$ to the variable $x$. Postulates 0.2 through 0.4 guarantee that the judgments

$$\Delta \vdash X : \mathbb{U} \quad \text{and} \quad \Delta \vdash x : X$$

are true. Thus a well-formed context assigns types to variables.

**0.2 / Context Extension**
Suppose that $X$ is a type. Declare $x$ as an object of type $X$. It is postulated that the implicit context is well-formed.

**Remark** In other words, the inference

$$\Gamma \ (\text{ctx}) \quad \Gamma \vdash X : \mathbb{U} \quad \Gamma, x : X \ (\text{ctx})$$

is defined to be a postulate.

**D.2 / Corollary**
If the context $\Delta$ is well-formed and $\Delta \vdash X : \mathbb{U}$ is true, then $\Delta, x : X$ is well-formed.

**Proof** By 0.2 and B.4. \hfill $\square$

**Intuition** Every well-formed context is constructed using D.1 and D.2.
0.3 / DECLARATION OF VARIABLES

Suppose that \( X \) is a type. Declare \( x \) as an object of type \( X \). It is postulated that \( x \) is an object of type \( X \).

Remark. In other words, the inference

\[
\Gamma \text{ ctx } \Gamma \vdash X : \mathbb{U} \\
\Gamma, x : X \vdash x : X
\]

is defined to be a postulate.

0.4 / CONTEXT WEAKENING

Suppose that \( X \) is a type and the label \( u \) applies to \( t \). Declare \( x \) as an object of type \( X \). It is postulated that the label \( u \) applies to \( t \).

Remark. In other words, the following inference is defined to be a postulate:

\[
\Gamma \text{ ctx } \Gamma \vdash X : \mathbb{U} \quad \Gamma \vdash t : u \\
\Gamma, x : X \vdash t : u
\]

Intuition. Since \( \Gamma \) does not declare \( x \), neither \( t \) nor \( u \) depends on \( x \).

SUBSTITUTION OF TERMS IN JUDGMENTS

DEFINITION. Suppose that the symbol \( R \) satisfies the following condition:

- If \( t \) and \( u \) are terms, then \( t R u \) is interpreted as a statement.

Then \( R \) is said to express a relationship between terms.

DEFINITION / chain of relationships / last term

Let \( t \) and \( u \) be terms and let \( R \) express a relationship between terms.

- The symbol \( t R u \) is defined to be a chain of relationships with \( u \) as its last term.
- If \( c \) is a chain of relationships with \( t \) as its last term, then the symbol

\[
c R u
\]

is defined to be a chain of relationships with \( u \) as its last term. It is interpreted as the statement \( c \) and \( t R u \).
EXAMPLE If $t$, $u$, and $v$ are terms, then the symbol

$$ t \equiv u \equiv v $$

is a chain of relationships. It is interpreted as stating that $t \equiv u$ and $u \equiv v$.

DEFINITION Let $t$ and $u$ be terms. The symbol $t \equiv u$ is called an identity, and the statement that

- The identity $t \equiv u$ is satisfied

means that $t$ is substitutable for $u$. It is sometimes said that $t$ is identical to $u$ or that $t$ is $u$. This is an abuse of language if $t$ and $u$ are distinct symbols.

D.3 / INTUITION

Let $t$ and $u$ be terms and $J_t$ a judgment. Let $J_u$ be the judgment constructed from $J_t$ by substituting $u$ for $t$. If the judgments

$$ J_t \quad \text{and} \quad u \equiv t $$

are derivable from the text $H$, then $J_u$ is derivable from $H$. This is guaranteed by postulates 0.5 through 0.7.

0.5 / SUBSTITUTION IN IDENTITIES

Let $t$, $u$, and $v$ be terms. It is postulated that:

1. The term $t$ is substitutable for itself.
2. If $t \equiv u$, then $u \equiv t$.
3. If $t \equiv u \equiv v$, then $t \equiv v$.

D.4 / THEOREM

Let $t$, $u$, and $v$ be terms.

1. If $t \equiv u$ and $t \equiv v$, then $u \equiv v$.
2. If $u \equiv t$ and $v \equiv t$, then $u \equiv v$.

Proof In each case, $u \equiv t$ and $t \equiv v$ by 0.4.2 and B.6, so $u \equiv v$ by 0.4.3.

INTUITION Let $J_t$ be the identity $t \equiv v$ or $v \equiv t$ in D.3. If $J_t$ and $u \equiv t$ are derivable from $H$, then $J_u$ is derivable from $H$ by 0.5 and D.4.

0.6 / SUBSTITUTION IN TERMS

Let $\alpha$ be a prefix and $t_1$, $t_2$, $u_1$, and $u_2$ are terms. Suppose that $t_1 \equiv t_2$ and $u_1 \equiv u_2$. It is postulated that $\alpha(t_1) \equiv \alpha(t_2)$ and $t_1(u_1) \equiv t_2(u_2)$.
Suppose that \( t_1 \equiv t_2 \) and \( u_1 \equiv u_2 \), where \( t_1, t_2, u_1, \) and \( u_2 \) are terms. If the label \( u_1 \) applies to \( t_1 \), it is postulated that the label \( u_2 \) applies to \( t_2 \).

**Remark** In other words, the symbol

\[
\Gamma \text{ ctx } t_1 \equiv t_2 \quad u_1 \equiv u_2 \quad \Gamma \vdash t_1 : u_1 \\
\Gamma \vdash t_2 : u_2
\]

is defined to be a postulate.

**D.5 / Corollary**

Suppose that \( t_1, t_2, u_1, \) and \( u_2 \) are terms such that the label \( u_1 \) applies to \( t_1 \).

1. If \( t_1 \equiv t_2 \), then the label \( u_1 \) applies to \( t_2 \).
2. If \( u_1 \equiv u_2 \), then the label \( u_2 \) applies to \( t_1 \).

**Proof** By 0.6 and 0.4.1.

**Intuition** With the notation of D.3, let \( J_t \) be either of the judgments

\[
\Delta \vdash t : v \quad \text{or} \quad \Delta \vdash v : t.
\]

If \( J_t \) and \( u \equiv t \) are derivable from \( H \), then \( J_u \) is derivable from \( H \) by D.5.

**Intuition** Let \( \Delta \) be a well-formed context and \( J_t \) the judgment

\[
\Delta, x : t \ \text{ctx}
\]

If \( J_t \) and \( u \equiv t \) are true, then \( J_u \) is true. Indeed, since \( J_t \) is true, \( t \) is a type in the context \( \Delta \) (intuitively, by 0.2). Therefore \( u \) is a type by D.5. It follows from 0.2 that the context \( \Delta, x : u \) is well-formed.

**Substitution of Terms for Variables**

**Definition** If \( n \) is a number, then \( b_n \) is a term. It is called the placeholder with index \( n \).

**Intuition** Let \( X \) be a type. A mathematical operation defined on \( X \) is an object \( f \) which encodes a method of constructing an output object \( f(a) \) from an input object \( a \) of type \( X \). The encoding uses variables, but can be simplified by replacing them with placeholders.
**Definition**  Numbers, constructors, and the *successor symbol* $s$ are defined to be constants.

**0.8 / Incrementing Placeholders**

Let $n$ be a number, $t$ and $u$ terms, and $x$ a variable. Let $c$ be a variable other than $x$ or a constant. It is postulated that the following identities are satisfied:

\[ \land_x(c) \equiv c, \quad \land_x(x) \equiv b_0, \quad \land_x(b_n) \equiv b_{s(n)}, \quad \land_x(t(u)) \equiv \land_x(t, \land_x(u)). \]

*Intuition*  The term $\land_x(t)$ is constructed from $t$ by *substituting* $b_0$ for $x$ and *incrementing* the indices of the other placeholders.

**0.9 / Decrementing Placeholders**

Let $n$ be a number, $a$, $t$, and $u$ terms, and $c$ a variable or a constant. It is postulated that the following identities are satisfied:

\[ \lor_a(c) \equiv c, \quad \lor_a(b_0) \equiv a, \quad \lor_a(b_{s(n)}) \equiv b_n, \quad \lor_a(t(u)) \equiv \lor_a(t, \lor_a(u)). \]

*Intuition*  The term $\lor_a(t)$ is constructed from $t$ by *substituting* $a$ for $b_0$ and *decrementing* the indices of the other placeholders.

**Notation**  Let $a$ and $t$ be terms. Define

\[ \left[ \frac{a}{x} \right](t) := \lor_a(\land_x(t)). \]

*Intuition*  According to 0.8 and 0.9, the term $\left[ \frac{a}{x} \right](t)$ is constructed from $t$ by *substituting* $a$ for $x$.

**D.6 / Theorem**

Let $a$ be a term, $c$ a constant, $n$ a number, and $x$ a variable. Then

\[ \left[ \frac{a}{x} \right](c) \equiv c, \quad \left[ \frac{a}{x} \right](b_n) \equiv b_n, \quad \left[ \frac{a}{x} \right](x) \equiv a. \]

*Proof*  Let $n$ be a number. It follows from 0.8 and 0.9 that

\[ \lor_a(\land_x(b_n)) \equiv \lor_a(b_{s(n)}) \equiv b_n \quad \text{and} \quad \lor_a(\land_x(x)) \equiv \lor_a(b_0) \equiv a. \]

**D.7 / Theorem**

Let $x$ be a variable and $a$, $t$, and $u$ terms. Then

\[ \left[ \frac{a}{x} \right](t(u)) \equiv \left[ \frac{a}{x} \right](t) \left[ \frac{a}{x} \right](u). \]
Proof It follows from 0.8 and 0.9 that
\[ \forall a\left(\land x(t(u))\right) \equiv \forall a\left(\land x(t, \land x(u))\right) \equiv \forall a\left(\land x(t), \forall a(\land x(u))\right). \]

\[ \vdash \square \]

0.10 / REFLEXIVE SUBSTITUTION

Let \( t \) be a term and \( x \) a variable. It is postulated that
\[ \left[ x \right](t) \equiv t. \]

0.11 / INDEPENDENT SUBSTITUTION

Suppose that \( X \) is a type, \( a \) is an object of type \( X \), and the label \( u \) applies to \( t \). Declare \( x \) as an object of type \( X \). It is postulated that
\[ \left[ a \right](t) \equiv t. \]

Intuition As in context weakening (0.4), the term \( t \) does not depend on \( x \). Therefore \( t \) is not changed by substituting \( a \) for \( x \).

D.8 / COROLLARY

Suppose that \( X \) and \( Y \) are types, \( a \) is an object of type \( X \), and \( b \) is an object of type \( Y \). Declare \( x \) as an object of type \( X \). Then
\[ \left[ a \right](b(x)) \equiv b(a) \quad \text{and} \quad \left[ a \right](x(b)) \equiv a(b). \]

Proof By D.6, D.7, and 0.11. \[ \square \]

D.12 / CONSTRUCTION BY SUBSTITUTION

Assume that \( X \) is a type and \( a \) is an object of type \( X \). Declare \( x \) as an object of type \( X \) and suppose that the label \( u \) applies to \( t \). It is postulated that the label \( \left[ a \right](u) \) applies to \( \left[ a \right](t) \).

Remark In other words, the symbol
\[ \Gamma \mid \mid \quad \Gamma \vdash X : U \quad \Gamma \vdash v : X \quad \Gamma, x : X \vdash t : u \]
\[ \Gamma \vdash \left[ a \right](t) : \left[ a \right](u) \]

is defined to be a postulate.

D.9 / COROLLARY

Assume that \( X \) is a type and \( a \) is an object of type \( X \). Declare \( x \) as an object of type \( X \). Suppose that \( Y \) is a type. Then \( \left[ a \right](Y) \) is a type.

Proof Since \( \left[ a \right](U) \equiv U \) by 0.12, the result follows from D.5. \[ \square \]
PART II

FORMALIZATION OF MATHEMATICS
MATHEMATICAL OPERATIONS

PRODUCT TYPES

NOTATION The symbols \( \land \) and \( \Pi \) denote the constructors with indices one and two, respectively.

NOTATION If \( X \) is a type, then \( \land_X \{ \land_x (y) \} \) is denoted by
\[
\land_{(x:X)} y \quad \text{or by} \quad x : X \mapsto y.
\]
The variable \( x \) is bound with depth zero in this term.

0.13 / CONSTRUCTION OF PRODUCT TYPES

Let \( X \) be a type. Declare \( x \) as an object of type \( X \). Suppose that \( Y \) is a type. It is postulated that
\[
\Pi_{(x:X)} Y := \Pi(X, \land_{(x:X)} Y) : U.
\]

Remark In other words, the following inference is defined to be a postulate:
\[
\Gamma \vdash X : U \quad \Gamma, x : X \vdash Y : U
\quad \frac{}{\Gamma \vdash \Pi_{(x:X)} Y : U}.
\]

DEFINITION The type constructed above is called a product type. An object of a product type is called a mathematical operation, or simply an operation.

NOTATION Let \( k \) and \( n \) be numbers. Define
\[
s^0(n) := n \quad \text{and} \quad s^{s(k)}(n) := s(s^k(n)).
\]

Let \( x \) be a variable and \( Y \) and \( y \) terms, where \( x \) is bound with depth \( k \) in \( Y \) and/or \( y \). If the implicit context of the statement
\[
\bullet \ y \text{ is an object of type } Y
\]
has length \( n \) in the entry \( E \), then \( x \) is understood to have index \( s^k(n) \) in \( E \).
EXAMPLE  Let $X$, $Y$, and $Z$ be terms. If the implicit context of the statement

$$\prod_{(x:X)} \left( \prod_{(y:Y)} Z \right) : U$$

has length $n$ in the entry $E$, then the variables $x$ and $y$ are understood to have indices $n$ and $s(n)$, respectively, in $E$.

INTUITION  Bound variables can be declared in order of increasing depth.

1.1 / THEOREM

Let $X$ and $Y$ be types. Then

$$X \rightarrow Y := \prod_{(x:X)} Y : U.$$  

Proof  Declare $x$ as an object of type $X$. Then $Y$ is a type by 0.4. Hence the result by 0.13. \qed

Definition  An object of type $X \rightarrow Y$ is called an operation from $X$ to $Y$ or a family of objects of type $Y$ indexed by $X$.

Notation  If $f$ is assumed to be an operation from $X$ to $Y$, then $X$ and $Y$ are understood to be types. The symbol $Y^X$ denotes the type $X \rightarrow Y$.

0.14 / APPLICATION OF AN OPERATION

Let $a$ be an object of type $X$. Declare $x$ as an object of type $X$. Suppose that $Y$ is a type and $f$ is an operation of type $\prod_{(x:X)} Y$. It is postulated that $f(a)$ is an object of type $\left[ \frac{a}{x} \right] (Y)$.

Remark  In other words, the inference

\[
\Gamma \ 	ext{ctx} \quad \Gamma \vdash X : U \quad \Gamma \vdash a : X \quad \Gamma, x : X \vdash Y : U \quad \Gamma \vdash f : \prod_{(x:X)} Y \\
\Gamma \vdash f(a) : \left[ \frac{a}{x} \right] (Y)
\]

is defined to be a postulate.

Definition  The operation $f$ is said to be defined on the type $X$, which is called the domain of $f$. The object $f(a)$ is called the image of $a$ under $f$ or the value of $f$ at $a$.

1.2 / THEOREM

Let $f$ be an operation from $X$ to $Y$ and $a$ an object of type $X$. Then $f(a)$ is an object of type $Y$. 
Proof Declare $x$ as an object of type $X$. Then $Y$ is a type by 0.4, so it follows from 0.14 and D.5 that

$$f(a) : \left[\frac{a}{x}\right](Y) \equiv Y.$$ 

\[\square\]

Intuition An operation $f$ from $X$ to $Y$ encodes a method of constructing an object $f(a)$ of type $Y$ from an object $a$ of type $X$.

1.3 / Theorem

Let $X$ be a type. Then $X \rightarrow \mathbb{U}$ is a higher-order type.

Proof The objects $X$ and $\mathbb{U}$ are higher-order types by 0.1. Since the implicit number $\nu$ is arbitrary, the result follows from 1.1. \[\square\]

Definition Let $m$ be a number and $X$ a type of order $m$. An operation from $X$ to $U_m$ is called a type family of order $m$ indexed by $X$.

1.4 / Lemma

Let $F$ be a type family indexed by $X$. Declare $x$ as an object of type $X$. Then the object $F_x$ is a type.

Proof It follows from 0.3 and 0.4 that $x$ has type $X$ and $F$ is a type family indexed by $X$. Hence the result by 1.2, since $\Gamma$ is an arbitrary context. \[\square\]

Notation Let $t$, $u$ and $v$ be terms and $\Lambda$ a constructor. Then

$$\Lambda_{(x:t)} u(v) := \Lambda_{(x:t)} (u(v)).$$

1.5 / Corollary

If $F$ is a type family indexed by $X$, then $\Pi_{(x:X)} F_x$ is a type.

Proof By 1.4 and 0.13. \[\square\]

Definition The type $\Pi_{(x:X)} F_x$ is denoted by $\Pi_X(F)$ or by

$$\prod_{x:X} F_x.$$ 

An object of type $\Pi_X(F)$ is called a selection of $F$.

Notation If $f$ is assumed to have type $\Pi_X(F)$, then $X$ is understood to be a type and $F$ is understood to be a type family indexed by $X$. 
1.6 / THEOREM

Let \( f \) be an operation of type \( \prod_X (F) \) and \( a \) an object of type \( X \). Then \( f(a) \) is an object of type \( F_a \).

Proof Declare \( x \) as an object of type \( X \). Then \( F(x) \) is a type by 1.4, so \( f(a) \) is an object of type \( [a_x](F(x)) \) by 0.14. Hence the result by D.5 and D.8. \( \square \)

Intuition An operation \( f \) of type \( \prod_X (F) \) encodes a method of constructing an object \( f(a) \) of type \( F_a \) from an object \( a \) of type \( X \).

0.15 / UNIQUENESS OF DOMAINS

Suppose that \( f \) has types \( \prod_{X_1} (F_1) \) and \( \prod_{X_2} (F_2) \). Then \( X_1 \equiv X_2 \).

Intuition Either \( X_1 \) or \( X_2 \) can be referred to as the domain of \( f \).

CONSTRUCTION OF MATHEMATICAL OPERATIONS

0.16 / CONSTRUCTION OF OPERATIONS

Let \( X \) be a type. Declare \( x \) as an object of type \( X \). Suppose that \( Y \) is a type and \( t \) is an object of type \( Y \). It is postulated that

\[ \lambda_{(x:X)} t : \prod_{(x:X)} Y \]

Remark In other words, the inference

\[
\Gamma \text{ ctx} \quad \Gamma \vdash X : U \quad \Gamma, x : X \vdash Y : U \quad \Gamma, x : X \vdash t : Y
\]

\[ \Gamma \vdash \lambda_{(x:X)} t : \prod_{(x:X)} Y \]

is defined to be a postulate.

DEFINITION Let \( x \) and \( y \) be variables and \( X, Y, \) and \( t \) terms. If the term

\[ x : X \mapsto t \]

is an operation, it is said to be defined for \( x \) of type \( X \) by the value \( t \).

0.17 / EVALUATION OF AN OPERATION

Let \( a \) be an object of type \( X \). Declare \( x \) as an object of type \( X \). If \( Y \) is a type and \( \lambda_X (u) \) has type \( \prod_{(x:X)} Y \), it is postulated that \( (\lambda_X (u))(a) \equiv \land_a (u) \).
1.7 / COROLLARY

Let $a$ be an object of type $X$. Declare $x$ as an object of type $X$. Suppose that $Y$ is a type and $t$ is an object of type $Y$. Then

$$(\lambda (x : X) \ t)(a) \equiv \left[ \frac{a}{x} \right](t).$$

Proof By 0.16 and 0.17.

Remark In particular, it follows from 0.10 that $(\lambda (x : X) \ t)(x) \equiv t$.

1.8 / CONSTANT OPERATION

Let $X$ and $Y$ be types and $b$ an object of type $Y$. Construct an operation

$$\kappa_X(b) : X \rightarrow Y$$

such that $\kappa_X(b, a) \equiv b$ if $a$ is an object of type $X$.

Solution Define $\kappa_X(b)$ as the term

$$x : X \mapsto b.$$

It has the required type by 0.4 and 0.16. Given an object $a$ of type $X$,

$$\kappa_X(b, a) \equiv \left[ \frac{a}{x} \right](b) \equiv b$$

by 1.7 and 0.11.

Definition The object $\kappa_X(b)$ is called a constant operation.

Remark By definition, $X \rightarrow Y \equiv \prod_X (\kappa_X(Y))$.

1.9 / IDENTITY OPERATION

Let $X$ be a type. Construct an operation $1_X : X \rightarrow X$ such that $1_X(a) \equiv a$ if $a$ is an object of type $X$.

Solution Define $1_X$ as the term

$$x : X \mapsto x.$$

It has the required type by 0.3 and 0.16. Given an object $a$ of type $X$,

$$1_X(a) \equiv \left[ \frac{a}{x} \right](x) \equiv a.$$

Definition The object $1_X$ is called the identity operation of $X$. 


1.10 / EVALUATION OPERATOR

Let $F$ be a type family indexed by $X$ and $a$ an object of type $X$. Construct

$ev_a : \prod_X (F) \to F(a)$

such that $ev_a(f) \equiv f(a)$ if $f$ is a selection of $F$.

Proof Define $ev_a$ as the term

$f : \prod_X (F) \mapsto f(a).$

It has the required type by 1.6 and 0.16. Given a selection $f$ of $F$,

$ev_a(f) \equiv \left[ \frac{f}{x} \right](x(a)) \equiv f(a). \qed$

DEFINITION The object $ev_a$ is called an evaluation operator.

1.11 / LEMMA

Let $f$ be an operation from $X$ to $Y$ and $G$ a type family indexed by $Y$. Then

$\prod_{x:X} G(f(x)) : U.$

Proof Declare $x$ as an object of type $X$. It follows from 1.2 that $f(x)$ is an object of type $Y$ and $G(f(x))$ is a type. Hence the result by 0.13. \qed

1.12 / COMPOSITION OF OPERATIONS

Given an operation $f$ from $X$ to $Y$ and an operation $g$ of type $\prod_Y (G)$, construct

$g \circ f : \prod_{x:X} G(f(x))$

such that $g \circ f(a) \equiv g(f(a))$ if $a$ is an object of type $X$.

Solution Define $g \circ f$ as the term

$x : X \mapsto g(f(x)).$

Declare $x$ as an object of type $X$. It follows from 1.2 that $f(x)$ has type $Y$, so $g(f(x))$ has type $G(f(x))$ by 1.6. Given an object $a$ of type $X$,

$g \circ f(a) \equiv \left[ \frac{a}{x} \right](g(f(x))) \equiv g(f(a)).$

by 1.7, D.7, 0.11, and D.8. \qed
Definition. The operation \( g \circ f \) is called the \textit{composition of} \( g \) \textit{with} \( f \).

1.13 / Corollary

If \( f \) is an operation from \( X \) to \( Y \) and \( g \) is an operation from \( Y \) to \( Z \), then

\[
g \circ f : X \rightarrow Z.
\]

\textbf{Proof} It follows from 1.12 and the definitions that \( g \circ f \) is a selection of

\[
\mathcal{K}_Y(Z) \circ f \equiv \lambda_{(x : X)} \left( \mathcal{K}_Y(Z, f(x)) \right) \equiv \lambda_{(x : X)} Z \equiv \mathcal{K}_X(Z).
\]

1.14 / Corollary

Let \( f \) be an operation from \( X \) to \( Y \) and \( g \) an operation of type \( \prod_Y(G) \). Then

\[
g \circ f : \prod_Y(G \circ f).
\]

1.15 / Theorem

Let \( f \) be an operation from \( X \) to \( Y \) and \( g \) an operation from \( Y \) to \( Z \). If \( H \) is a type family indexed by \( Z \) and \( h \) is a selection of \( H \), then

\[
h \circ (g \circ f) : \prod_X (H \circ (g \circ f)) \quad \text{and} \quad (h \circ g) \circ f : \prod_X ((H \circ g) \circ f)
\]

\textbf{Proof} By 1.13 and 1.14.

Uniqueness of Operations and Domains

0.18 / Uniqueness of Operations

Let \( X \) be a type and let \( f \) and \( g \) be operations defined on \( X \). Declare \( x \) as an object of type \( X \). If \( f(x) \equiv g(x) \), it is postulated that \( f \equiv g \).

\textbf{Remark} More precisely, \( f \) is a selection of \( F \) and \( g \) is a selection of \( G \), where \( F \) and \( G \) are type families.

\textbf{Intuition} The operations \( f \) and \( g \) are identical if and only if \( f(x) \equiv g(x) \) for all \( x \) of type \( X \).

1.16 / Corollary

Let \( F \) be a type family indexed by \( X \) and \( f \) a selection of \( F \). Then

\[
x : X \mapsto f(x)
\]

is a selection of \( F \). Furthermore, it is identical to \( f \).
Proof Let \( g \) denote the given formal operator. Then

\[
g(x) \equiv \left[ \frac{x}{x} \right] (f(x)) \equiv f(x)
\]

by 1.7, 0.10, and 0.11. Hence the result by 1.6, D.5, 0.16, and 0.18. \( \square \)

1.17 / COROLLARY

Let \( f \) be an operation from \( X \) to \( Y \). Then

\[
f \circ 1_X \equiv f \equiv 1_Y \circ f.
\]

Proof Declare \( x \) as an object of type \( X \). By 1.9 and 1.12,

\[
f(1_X(x)) \equiv f(x) \equiv 1_Y(f(x)).
\] \( \square \)

1.18 / COROLLARY

Let \( f \) be an operation from \( X \) to \( Y \) and \( g \) an operation from \( Y \) to \( Z \). If \( H \) is a type family indexed by \( Z \) and \( h \) is a selection of \( H \), then

\[
h \circ (g \circ f) \equiv (h \circ g) \circ f.
\]

Proof Declare \( x \) as an object of type \( X \). By 1.12 and 0.18,

\[
h(g \circ f(x)) \equiv h(g(f(x))) \equiv h \circ g(f(x)).
\] \( \square \)

Notation The operation \( h \circ (g \circ f) \) is denoted by \( h \circ g \circ f \).
2

NATURAL NUMBERS

THE NATURAL NUMBER TYPE

DEFINITION The constructor with index three is denoted by \( \mathbb{N} \) and called the
natural number type.

0.19 / CONSTRUCTION OF THE NATURAL NUMBER TYPE
It is postulated that \( \mathbb{N} \) is a type.

DEFINITION An object \( n \) of type \( \mathbb{N} \) is called a natural number. An operation
\( f \) defined on \( \mathbb{N} \) is called a sequence. Its value \( f_n \) at \( n \) is called its \( n \)th term.

DEFINITION A sequence of type \( \mathbb{N} \rightarrow X \) is called a sequence of objects of
type \( X \), or a sequence in \( X \). In particular:

- a sequence of types is called a type sequence
- a sequence of natural numbers is called a numerical sequence.

0.20 / CONSTRUCTION OF NATURAL NUMBERS
It is postulated that:

1. Zero is a natural number
2. If \( n \) is a natural number, then \( s(n) \) is a natural number.

Definition The object \( s(n) \) is called the successor of \( n \).

INTUITION Natural numbers can be thought of as numbers, since both are
constructed from zero by iterating the successor operation. The difference is
that variables can be declared as natural numbers.

DEFINITION Let \( C \) be a type sequence. A recursor of \( C \) is an object of type

\[
\prod_{(x:\mathbb{N})} \left( C_x \rightarrow C_{s(x)} \right).
\]

NOTATION The symbol \( \mathcal{R} \) denotes the constructor with index four.
Chapter 2

0.21 / Recursive Definition

Let \( C \) be a type sequence, \( T \) a recursor of \( C \), and \( c_0 \) an object of type \( C_0 \). It is postulated that \( R(C, T, c_0) \) is a selection of \( C \), and that

\[
R(C, T, c_0, 0) \equiv c_0.
\]

Declare \( x \) as a natural number. It is postulated that

\[
R(C, T, c_0, s(x)) \equiv T(x, R(C, T, c_0, x)).
\]

Intuition The sequence \( f := R(C, T, c_0) \) is defined as follows:

1. Define \( f(0) \) as the object \( c_0 \) of type \( C_0 \).
2. Given a natural number \( x \) and an object \( f(x) \) of type \( C_x \), define \( f(s(x)) \) as the object \( T(x, f(x)) \) of type \( C_{s(x)} \).

Since the natural numbers are constructed from zero by iterating the successor operation, this procedure defines \( f(x) \) for every natural number \( x \).

Definition It is said that \( f \) is defined recursively on \( C \) by the identities

\[
f(0) \equiv c_0 \quad \text{and} \quad f(s(x)) \equiv T(x, f(x))
\]

which are called its initial condition and recurrence relation, respectively. The value of \( f \) at zero is called its initial value.

Definition Let \( X \) be a type. A recursor \( T \) of the constant type sequence \( \kappa_N(X) \) is called a recursor of \( X \). If \( x_0 \) is an object of type \( X \), then

\[
R_{X}(X, T, x_0) := R(\kappa_N(X), T, x_0).
\]

This sequence is said to be defined recursively on \( X \).

Remark A recursor of \( X \) is a sequence in \( X \rightarrow X \).

Definition A recursor of \( \mathbb{N} \) is called a numerical recursor.

Recursive Definitions

2.1 / Predecessor Operation

Construct a numerical sequence \( pd \) such that

\[
pd(0) \equiv 0 \quad \text{and} \quad pd(s(n)) \equiv n
\]

if \( n \) is a natural number.
Solution  Let $T$ denote the term

$$x : \mathbb{N} \mapsto \kappa_{\mathbb{N}}(x)$$

Then $T$ is a recursor of $\mathbb{N}$ by 0.3, 1.8, and 0.16. Define

$$\text{pd} := \mathcal{R}_\kappa(\mathbb{N}, T, 0),$$

which is a numerical sequence by 0.21. Declare $x$ as a natural number. Then

$$\text{pd}(s(x)) \equiv T(x, \text{pd}(x)) \equiv \kappa_{\mathbb{N}}(x, \text{pd}(x)) \equiv x.$$  

**Definition**  The operation $\text{pd}$ is called the *predecessor operation*. If $n$ is a natural number, then $\text{pd}(n)$ is called the *predecessor of $n$*.

**Definition**  Let $x$ and $y$ be variables and $X$, $Y$, and $t$ terms. The term

$$x : X \mapsto (y : Y \mapsto t)$$

is denoted by $x : X, y : Y \mapsto t$.

If this term is a mathematical operation, it is said to be *defined for $x$ of type $X$ and $y$ of type $Y$ by the value $t$*.

### 2.2 / Iteration of an Operation

Let $f$ be an operation from $X$ to $X$. Construct a recursor $\text{itr} f$ of $X$ such that

$$\text{itr} f(0) \equiv 1_X \quad \text{and} \quad \text{itr} f(s(n)) \equiv f \circ (\text{itr} f(n))$$

if $n$ is a natural number.

Solution  Let $T_f$ denote the term

$$x : \mathbb{N}, y : X^X \mapsto f \circ y.$$ 

Then $T_f$ is a recursor of $X \mapsto X$ by 1.13. Define

$$\text{itr} f := \mathcal{R}_\kappa(X^X, T_f, 1_X).$$

Declare $x$ as a natural number. Then

$$\text{itr} f(s(x)) \equiv T_f(x, \text{itr} f(x)) \equiv f \circ (\text{itr} f(x)).$$

**Definition**  Let $n$ be a natural number. The operation

$$f^n := \text{itr} f(n)$$

is called the *$n$th iterate of $f$*. With this notation,

$$f^0 \equiv 1_X \quad \text{and} \quad f^{s(n)} \equiv f \circ f^n.$$
2.3 / COROLLARY

If \( f \) is an operation from \( X \) to \( X \), then
\[
\begin{align*}
f^1 & \equiv f, \\ f^2 & \equiv f \circ f, \\ f^3 & \equiv f \circ f \circ f.
\end{align*}
\]
\[\square\]

**NOTATION** Let \( x \) be a variable, \( t \) and \( X \) terms, and \( L \) a list of terms. Then
\[
L : X, x : X \mapsto t \quad \text{is denoted by} \quad L, x : X \mapsto t.
\]
If this symbol is a mathematical operation, it is said to be *defined for \( \mathcal{L} \) and \( x \) of type \( X \) by the value \( t \).*

2.4 / ADDITION OF NATURAL NUMBERS

**Construct** a numerical recursor \(+\) such that
\[
+m(0) \equiv m \quad \text{and} \quad +m(s(n)) \equiv s(m) + m(n) \quad \text{if} \quad n \quad \text{are natural numbers.}
\]

**Solution** Define \(+\) as the term
\[
x, y : \mathbb{N} \mapsto s^y(x),
\]
which is a numerical recursor by 0.20 and 2.2. It follows that
\[
+^x(0) \equiv s^0(x) \equiv x \\
+^x(s(y)) \equiv s^s(y)(x) \equiv s(s^y(x)) \equiv s^s(x) \\
\]
\[\square\]

**DEFINITION** If \( n \) and \( m \) are natural numbers, then
\[
+^m n \quad \text{is a natural number, called the} \quad \text{sum of} \quad n \quad \text{and} \quad m.
\]

With this notation,
\[
m + 0 \equiv m \quad \text{and} \quad m + s(n) \equiv s(m + n).
\]

The numerical recursor \(+\) is called *addition of natural numbers.*

2.5 / COROLLARY

If \( m \) and \( n \) are natural numbers, then
\[
m + 1 \equiv s(m) \quad \text{and} \quad m + (n + 1) \equiv (m + n) + 1.
\]
\[\square\]

**Example** The judgment \( 2 + 2 \equiv 4 \) is valid.
VALUES OF AN OPERATION

IMAGE TYPES

0.22 / CONSTRUCTION OF IMAGE TYPES
Let $X$ be a type and $f$ an operation defined on $X$. It is postulated that

$$\text{im} f := f(X) : \mathbb{U}.$$  

Definition The type $\text{im} f$ is called the *image of* $f$. An object of type $\text{im} f$ is called a *value of* $f$.

0.23 / CONSTRUCTION OF VALUES
Suppose that $X$ is a type, $f$ is an operation defined on $X$, and $a$ is an object of type $X$. It is postulated that $f(a)$ is a value of $f$.

3.1 / THEOREM
Let $X$ be a type and $f$ an operation defined on $X$. Then $f : X \to f(X)$.

Proof Declare $x$ as an object of type $X$. Then $f(x)$ is a value of $f$ by 0.23, so the result follows from 0.16. □

NOTATION The symbol $H$ denotes the constructor with index five.

DEFINITION Let $x$ be an object of type $X$ and $f$ an operation defined on $X$. If $f(x) \equiv y$, then $x$ is said to be a *preimage of* $y$ *under* $f$.

0.24 / THE PRINCIPLE OF CONSTRUCTIVE CHOICE
Let $X$ be a type and $f$ an operation defined on $X$. It is postulated that:

$$1 \quad f^* := H(X, f) : f(X) \to X$$

$$2 \quad f \circ f^* \equiv 1_{f(X)}.$$  

Definition The operation $f^*$ is called the *Hilbert inverse of* $f$. If $y$ is a value of $f$, then $f^*(y)$ is a preimage of $y$ under $f$. It is said to be *canonical*. 

35
Definition  The statement

- Choose an object $x$ of type $X$ such that $y \equiv f(x)$

means that $x$ denotes $f^*(y)$.

Intuition  The statement that $y$ is a value of $f$ means that $y \equiv f(x)$ for some object $x$ of type $X$.

3.2 / Theorem
Let $f$ be an operation from $X$ to $Y$ and $y$ a value of $f$. Then $y$ has type $Y$.

Proof  Choose an object $x$ of type $X$ such that $y \equiv f(f^*(x))$ by 0.24, so $y$ has type $Y$ by 1.2 and D.5. □

Definition  If $x$ is an object of type $X$, then the image of $\varepsilon_N(x)$ is denoted by $\{x\}$ and called the singleton type defined by $x$.

3.3 / Theorem
Let $x$ be an object of type $X$. Then $y : \{x\}$ if and only if $y \equiv x$. □

3.4 / Corollary
If $x$ is an object of type $X$ and $f$ is an operation defined on $\{x\}$, then

$$f \equiv \varepsilon_{\{x\}}(f(x)) \quad \text{and} \quad f(\{x\}) \equiv \{f(x)\}.$$ □

Definition  Let $X$ be a type. The canonical object of type $X$ is the term

$$\tau(X) := (\varepsilon_X(0))^*(0).$$

If $\tau(X)$ has type $X$, then $X$ is said to be inhabited.

3.5 / Theorem
Let $X$ be a type. If $x$ is an object of type $X$, then $X$ is inhabited.

Proof  Let $f := \varepsilon_X(0) : X \rightarrow N$. Then $f(x) \equiv 0$, so $\tau(X) \equiv f^*(f(x))$ has type $X$. □

Definition  If the type $X$ is inhabited, then the statement

- Choose an object $x$ of type $X$

means that $x$ denotes the canonical object of type $X$.

Intuition  Saying that $X$ is inhabited means that some term has type $X$. 
3.6 / Corollary

Let $f$ be an operation defined on $X$. If $f(X)$ is inhabited, then $X$ is inhabited.

Proof. By 3.5, since $f^*(\tau(f(X)))$ has type $X$ by 0.24 and 1.2. □

The Subtyping Relation

Definition. Let $A$ and $X$ be types. If $1_A : A \rightarrow X$, then $A$ is said to be a subtype of $X$, written $A \subseteq X$.

Intuition. The statement that $A$ is a subtype of $X$ means that every object of type $A$ is an object of type $X$.

Example. If $m$ is a number, then $U_m$ is a subtype of $U_{m+1}$ by 0.12. If $f$ is an operation from $X$ to $Y$, then $f(X) \subseteq Y$ by 3.2.

3.7 / Corollary

Let $f$ an operation defined on the type $X$. If $f(X) \subseteq Y$, then $f : X \rightarrow Y$.

Proof. By 1.13, since $f \equiv 1_{f(X)} \circ f$ by 3.1. □

3.8 / Theorem

If $X$ is a type, then $X$ is a subtype of itself. □

3.9 / Theorem

Let $X$, $Y$ and $Z$ be types. If $X \subseteq Y$ and $Y \subseteq Z$, then $X \subseteq Z$.

Proof. By 1.13, since $1_X \equiv 1_Y \circ 1_X$ by 1.20. □

Notation. If the term $A$ is assumed to be a subtype of $X$, then $A$ and $X$ are understood to be types.

Definition. Let $F$ be a type family indexed by $X$ and $f$ a selection of $F$. If $A$ is a subtype of $X$, then the operation

$$f \circ 1_A : \prod_{a : A} F_a$$

is denoted by $f \upharpoonright A$ and called the restriction of $f$ to $A$. Its image is denoted by $f[A]$ and called the image of $A$ through $f$. 

**Intuition**  The statement that \( y \) has type \( f[A] \) means that \( y \equiv f(a) \) for some object \( a \) of type \( A \).

**3.10 / Lemma**

Let \( X \) be a type, \( f \) an operation defined on \( X \), and \( A \) a subtype of \( X \). If \( a \) is an object of type \( A \), then \( f(a) \) has type \( f[A] \).

**Notation**  If \( x \) has type \( X \), then \( f[x] \) denotes the image of \( \{x\} \) through \( f \).

**3.11 / Theorem**

Let \( X \) be a type and \( f \) an operation defined on \( X \). Then \( f[X] \equiv f(X) \).

**Proof**  Since \( f \equiv f \circ 1_X \) by 1.20.

**0.25 / Substitution of Types**

Let \( X \) and \( Y \) be types. It is postulated that \( X \equiv Y \) if \( X \subseteq Y \) and \( Y \subseteq X \).

**Remark**  If \( X \equiv Y \), then \( X \subseteq Y \) and \( Y \subseteq X \) by 0.5 and 3.8.

**3.12 / Corollary**

If \( A \) is a subtype of \( X \), then \( 1_X[A] \equiv A \).

**Proof**  By 0.25. Indeed, if \( a \) is an object of type \( A \), then \( 1_X \circ 1_A(a) \equiv a \).

**3.13 / Corollary**

Let \( f \) be an operation from \( X \) to \( Y \) and \( g \) an operation from \( Y \) to \( Z \). If \( A \) is a subtype of \( X \), then

\[
g[f[A]] \equiv (g \circ f)[A].
\]

**Proof**  By 0.25. Indeed, if \( a \) is an object of type \( A \), then

\[
g \circ 1_{f[A]}(f \circ 1_A(a)) \equiv ((g \circ f) \circ 1_A)(a).
\]

**3.14 / Corollary**

Let \( X \) be a type and \( f \) an operation defined on \( X \). If \( A \subseteq B \) and \( B \subseteq X \), then

\[
f[A] \subseteq f[B] \quad \text{and} \quad (f|B)[A] \equiv f[A].
\]

**Proof**  Declare \( y \) as an object of type \( f[A] \). Choose an object \( a \) of type \( A \) such that \( y \equiv f(a) \). Then \( a \) has type \( B \), so \( y \) has type \( f[B] \). Furthermore

\[
(f \circ 1_B)[A] \equiv f[1_B[A]] \equiv f[A]
\]

by the previous two corollaries.
ASSEMBLIES

COPRODUCT TYPES

NOTATION The symbol $\Pi$ denotes the constructor with index six.

0.26 / CONSTRUCTION OF COPRODUCT TYPES

If $F$ is a type family indexed by $X$, it is postulated that $\Pi_X(F)$ is a type.

Definition The type $\Pi_X(F)$ is denoted by $\Pi_{(x:X)}F_x$ or

$$\Pi_{x:X}F_x$$

and called a coproduct type. An object of this type is called an assembly.

NOTATION The symbol $A$ denotes the constructor with index seven.

0.27 / CONSTRUCTION OF ASSEMBLIES

Let $F$ be a type family indexed by $X$. If $x$ is an object of type $X$ and $y$ an object of type $F_x$, it is postulated that

$$[x, y] := A(X, F, x, y) : \Pi_X(F).$$

Definition The object constructed in 0.27 is called the assembly of $x$ with $y$. It is said to consist of an object $x$ of type $X$ and an object $y$ of type $F_x$.

INTUITION The assembly $[x, y]$ combines $x$ and $y$ into a single object.

4.1 / THEOREM

Let $f$ be an operation from $X$ to $Y$ and $G$ a type family indexed by $Y$. Given a selection $g$ of the type family $G \circ f$, construct an operation

$$[f, g] : X \rightarrow \Pi_Y(G)$$

such that $[f, g](a) \equiv [f(a), g(a)]$ if $a$ is an object of type $X$. 
Proof Define \([f, g]\) as the term
\[x : X \mapsto [f(x), g(x)].\]
Declare \(x\) as an object of type \(X\). Then \(f(x)\) is an object of type \(Y\) and \(g(x)\) is an object of type \(G(f(x))\). Hence the result by 0.27, 0.16, and 1.7. □

4.2 / Canonical Inclusion

Let \(F\) be a type family indexed by \(X\) and \(x\) is an object of type \(X\). Construct
\[\iota_x : F_x \rightarrow \prod_X (F) \quad \text{such that} \quad \iota_x(y) = [x, y]\]
for all objects \(x\) of type \(F_x\).

Proof Define \(\iota_x\) as the operation \([\pi_{F_x}(x), 1_{F_x}]\). □

Definition The object \(\iota_x\) is called the canonical inclusion of \(F_x\) into \(\prod_X (F)\).

Separators and Combinators

Notation The symbol \(C\) denotes the constructor with index eight.

Notation If \(f\) is assumed to have type \(\prod_X (F)\), then \(F\) is understood to be a type family indexed by the type \(X\). If \(x, y\) and \(h\) are terms, then
\[h[x, y] := h([x, y]).\]

Definition Let \(H\) be a type family indexed by \(\prod_X (F)\). An object \(S\) of type
\[\prod_{x : X} \prod_{y : F_x} H[x, y],\]
is called a separator of \(H\). The term \(C(X, F, H, S)\) is denoted by
\[\prod_X (S) \quad \text{or} \quad \prod_{x : X} S(x)\]
and called the combinator of \(S\).

0.28 / Construction of Combinators

If \(H\) is a type family indexed by \(\prod_X (F)\) and \(S\) is a separator of \(H\), it is postulated that
\[\prod_X (S) : \prod_{\prod_X (F)} (H).\]
0.29 / Evaluation of Combinators

Let $S$ be a separator of $H$, where $H$ is a type family indexed by $\coprod_X (F)$. Declare $x$ as an object of type $X$ and $y$ as an object of type $F_x$. It is postulated that

$$\left( \coprod_X (S) \right)[x, y] \equiv S(x, y).$$

**Definition** Given a selection $h$ of $H$, the operation

$$x : X, y : F_x \mapsto h[x, y]$$

is denoted by $\text{sep } h$ and called the *separator of $H$ defined by $h$.*

4.3 / Corollary

Let $S$ a separator of $H$, where $H$ is a type family indexed by $\coprod_X (F)$. Then $\text{sep } \left( \coprod_X (S) \right) \equiv S$.

4.4 / First Component of an Assembly

Given a type family $F$ indexed by $X$, construct an operation

$$\sigma_1 : \coprod_X (F) \rightarrow X \quad \text{such that} \quad \sigma_1 [x, y] \equiv x$$

for all objects $x$ of type $X$ and all objects $y$ of type $F_x$.

**Solution** Define $\sigma_1$ as the combinator

$$\coprod_{x : X} (\kappa_{F_x}(x)).$$

4.5 / Second Component of an Assembly

Given a type family $F$ indexed by $X$, construct an operation

$$\sigma_2 : \coprod_{\coprod_X (F)} (F \circ \sigma_1) \quad \text{such that} \quad \sigma_2 [x, y] \equiv y$$

for all objects $x$ of type $X$ and all objects $y$ of type $F_x$.

**Solution** Define $\sigma_2$ as the combinator

$$\coprod_{x : X} 1_{F_x}.$$

**Definition** If $z$ is an object of the coproduct type $\coprod_X (F)$, then $\sigma_1(z)$ is called the *first component of $z$* and $\sigma_2(z)$ is called the *second component of $z*. 

\[ \]
0.30 / UNIQUENESS OF COMBINATORS

Let $h$ be a selection of $H$, where $H$ is a type family indexed by $\prod_X(F)$. It is postulated that

$$h \equiv \prod_{x:X} (\text{sep } h)(x).$$

4.6 / COROLLARY

Let $H$ be a type family indexed by $\prod_X(F)$. If $h_1$ and $h_2$ are selections of $H$ such that $\text{sep } h_1 \equiv \text{sep } h_2$, then $h_1 \equiv h_2$. □

4.7 / THEOREM

Let $F$ be a type family indexed by $X$ and $z$ an object of type $\prod_X(F)$. Then

$$z \equiv [\sigma_1(z), \sigma_2(z)].$$

Proof. Declare $x$ as an object of type $X$ and $y$ as an object of type $F_x$. Then

$$[\sigma_1, \sigma_2][x, y] \equiv [\sigma_1[x, y], \sigma_2[x, y]] \equiv [x, y]$$

by 4.1, 4.4, and 4.5. Hence the result by 4.6. □

Notation. The first component of $z$ is denoted by $z_1$ and the second component by $z_2$. Therefore $z \equiv [z_1, z_2]$. 
MATHEMATICAL PROPOSITIONS

PROOF TYPES AND TRUTH

NOTATION  The symbols $V$ and $\Box$ denote the constructors with indices nine and ten, respectively. If $m$ is a number, then $V(m)$ is denoted by $\mathcal{V}_m$.

Definition  The term $\mathcal{V}_m$ is called the propositional universe of order $m$. An object of type $\mathcal{V}_m$ is called a proposition of order $m$.

0.31 / CONSTRUCTION OF PROPOSITIONAL UNIVERSES
Let $m$ be a number. It is postulated that $\mathcal{V}_m$ is a type of order $m+1$ and a subtype of $\mathcal{V}_{m+1}$.

INTUITION  Propositions can be thought of as mathematical statements that can be manipulated using logical operations.

0.32 / CONSTRUCTION OF PROOF TYPES
Let $m$ be a number and $P$ a proposition of order $m$. It is postulated that $\Box(P)$ is a type of order $m$.

Definition  The type $\Box(P)$ is called the proof type of $P$. If it is inhabited, then $P$ is said to be true. An object of type $\Box(P)$ is said to prove $P$, and is called a proof of $P$ or a proof that $P$ is true.

Notation  The canonical proof of $P$ is denoted by $\Diamond P$. Thus $P$ is true if and only if $\Diamond P$ proves $P$. The symbol $P$ denotes the statement that “$P$ is true.”

NOTATION  The symbol $S$ denotes the constructor with index eleven.

0.33 / SUBTYPING PROPOSITIONS
Let $X$ and $Y$ be types. It is postulated that $S(X,Y)$ is a proposition and that its proof type is a subtype of $X \rightarrow Y$. If $X$ is a subtype of $Y$, it is postulated that $1_X$ proves $S(X,Y)$. 

43
Definition  The proposition $S(X, Y)$ is called a subtyping proposition.

0.34 / Uniqueness of Subtyping Proofs
Let $X$ and $Y$ be types, and $p$ a proof of $S(X, Y)$. It is postulated that $p \equiv 1_X$.

Intuition  If the proposition $S(X, Y)$ is true, then $1_X$ is its only proof.

5.1 / Corollary
Let $X$ and $Y$ be types. Then $X \subseteq Y$ if and only if $S(X, Y)$ is true.

Proof  Immediate from the previous two postulates.

5.2 / Theorem
Let $x$ be an object of type $X$ and $A$ a subtype of $X$. Then $x$ has type $A$ if and only if $\{x\} \subseteq A$.

Proof  It follows from 3.4 that $1_{\{x\}} \equiv x_{\{x\}}(x)$. Hence the result by 1.9.

Notation  Let $X$ and $Y$ be types and $A$ a subtype of $X$. Let $x$ be an object of type $X$ and $y$ a term. Subtyping propositions are interpreted as follows:

<table>
<thead>
<tr>
<th>Proposition</th>
<th>Interpretation</th>
<th>Notation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S(X, Y)$</td>
<td>$X$ is a subtype of $Y$</td>
<td>$X \subseteq Y$</td>
</tr>
<tr>
<td>$S({x}, A)$</td>
<td>$x$ is an object of type $A$</td>
<td>$x : A$</td>
</tr>
<tr>
<td>$S({x}, {y})$</td>
<td>$x$ is substitutable for $y$</td>
<td>$x \equiv y$</td>
</tr>
</tbody>
</table>

Table 8. Interpretation of subtyping propositions.

These interpretations are justified by theorems 5.1, 5.2, and 3.3, respectively.

Intuition  The judgments given in table 8 can be interpreted as propositions.

Notation  Suppose that $x$ and $y$ are both natural numbers, both sequences, or both types. Then the proposition $x \equiv y$ is denoted by $x = y$.

Definition  Let $X$ be a type of order $m$. An operation from $X$ to $\forall_m$ is called a predicate of order $m$ on $X$.

Example  Let $A$ be a subtype of $X$ and $f$ and $g$ operations on $X$. The terms $x : X \maps X : A$ and $x : X \mapsto f(x) \equiv g(x)$ are predicates on $X$. The former is called the predicate on $X$ defined by $A$. 
QUANTIFICATION OF A PREDICATE

NOTATION The constructor with index twelve is denoted by \( \forall \) and called the universal quantifier. For terms \( P \) and \( X \), the term \( \forall_X(P) \) is denoted by

\[
\forall_{(x:X)} P(x).
\]

0.35 / UNIVERSAL QUANTIFICATION

Let \( X \) be a type and \( P \) a predicate on \( X \). It is postulated that \( \forall_{(x:X)} P(x) \) is a proposition, and that

\[
\square(\forall_{(x:X)} P(x)) = \prod_{(x:X)} \square(P(x)).
\]

Definition The proposition \( \forall_X(P) \) is called the universal quantification of \( P \). It is interpreted as the statement

- For all \( x \) of type \( X \), the proposition \( P(x) \) is true.

The phrases for all, for every, and for each have the same meaning.

Intuition A proof of \( \forall_X(P) \) encodes a method of constructing a proof of the proposition \( P(a) \) from an object \( a \) of type \( X \).

5.3 / THE GENERALIZATION THEOREM

Let \( X \) be a type and \( P \) a predicate on \( X \). If either of the following statements expresses a valid inference, then so does the other:

1. If \( a \) is an object of type \( X \), then \( P(a) \) is true
2. The proposition \( \forall_{(x:X)} P(x) \) is true.

Proof If 1 is valid, then 2 is valid by 0.3, 0.16, and 3.5. If 2 is valid, then 1 is valid by 1.6 and 3.5.

5.4 / UNIVERSAL INSTANTIATION

Let \( X \) be a type, \( P \) a predicate on \( X \), and \( a \) an object of type \( X \). If \( P(x) \) is true for all \( x \) of type \( X \), then \( P(a) \) is true.

Proof Choose a proof \( f \) of \( \forall_X(P) \). Then \( f(a) \) proves \( P(a) \) by 1.6 and 7.1.

5.5 / COROLLARY

Let \( X \) and \( Y \) be types. Then \( X \subseteq Y \) if and only if every object of type \( X \) is an object of type \( Y \).
Proof Necessity follows from 5.3 and sufficiency from 5.4.

5.6 / corollary

Let $X$ be a type and $f$ and $g$ operations defined on $X$. Then $f \equiv g$ if and only if $f(x) \equiv g(x)$ for all $x$ of type $X$.

Proof Necessity follows from 0.5 and 5.3, sufficiency from 5.4 and 0.18.

5.7 / corollary

Let $F$ and $G$ be type families indexed by $X$. If $F_x \subseteq G_x$ for all $x$ of type $X$, then

$$\prod_{x:X} F_x \subseteq \prod_{x:X} G_x$$

and

$$\prod_{x:X} F_x \subseteq \prod_{x:X} G_x.$$ 

Proof of the first proposition Declare $f$ as an object of type $\prod_X (F)$ and $x$ as an object of type $X$. Then $f(x)$ has type $G_x$ by 1.6 and 5.4. Hence the result by 0.16 and 5.5.

Proof of the second proposition Declare $z$ as an object of type $\prod_X (F)$. Then $z_1$ has type $X$ by 4.4 and $z_2$ has type $G_{z_1}$ by 4.5 and 5.4. Hence the result by 4.7 and 5.5.

notation The constructor with index thirteen is denoted by $\exists$ and called the existential quantifier. For terms $P$ and $X$, the term $\exists_X (P)$ is denoted by

$$\exists_{(x:X)} P(x).$$

0.36 / existential quantification

Let $X$ be a type and $P$ a predicate on $X$. It is postulated that $\exists_{(x:X)} P(x)$ is a proposition, and that

$$\square (\exists_{(x:X)} P(x)) = \prod_{(x:X)} \square (P(x)).$$

Definition The proposition $\exists_X (P)$ is called the existential quantification of $P$. It is interpreted as the statements

- There exists $x$ of type $X$ such that $P(x)$ is true
- For some $x$ of type $X$, the proposition $P(x)$ is true.

The phrases there exists and there is have the same meaning.

Intuition A proof of $\exists_X (P)$ is an assembly consisting of an object $a$ of type $X$ and a proof of the proposition $P(a)$. 
5.8 / EXISTENTIAL GENERALIZATION

Let \( a \) be an object of type \( X \) and \( P \) a predicate on \( X \). If the proposition \( P(a) \) is true, then \( P(x) \) is true for some \( x \) of type \( X \).

Proof Choose a proof \( p \) of \( P(a) \). Then \([a, p]\) proves \( \exists_X(P) \) by 0.36. \( \square \)

Definition Let \( a \) be an object of type \( X \) and \( P \) a predicate on \( X \). If \( P(a) \) is true, then \( a \) is called a witness of \( P \) and is said to satisfy \( P \).

Notation Let \( X \) be a type and \( P \) a predicate on \( X \). The term

\[
\sigma_1 \left( \exists (x : X) \ P(x) \right)
\]

is denoted by \( \varepsilon_X(P) \) or by \( \varepsilon_{(x : X)} P(x) \) and called the canonical witness of \( P \).

Definition Suppose that \( P(x) \) is true for some \( x \) of type \( X \). The statement

- Choose a witness \( a \) of \( P \)

means that \( a \) denotes the canonical witness of \( P \).

5.9 / EXISTENTIAL INSTANTIATION

Let \( X \) be a type and \( P \) a predicate on \( X \). The proposition \( \exists_X(P) \) is true if and only if \( \varepsilon_X(P) \) has type \( X \) and \( P(\varepsilon_X(P)) \) is true.

Proof The condition is necessary by 4.4 and 4.5. It is sufficient by 5.8 \( \square \)

5.10 / COROLLARY

Let \( X \) and \( Y \) be types, \( A \) a subtype of \( X \), and \( f \) an operation from \( X \) to \( Y \). Then \( y \) belongs to \( f[A] \) if and only if \( y \equiv f(a) \) for some object \( a \) of type \( A \).

Proof Necessity follows from 0.24 and 5.8, sufficiency from 5.9 and 3.10. \( \square \)

5.11 / COROLLARY

The type \( X \) is inhabited if and only if \( x : X \) for some object \( x \) of type \( X \).

Proof The condition is necessary by 5.8 and sufficient by 5.9 and 3.5. \( \square \)

Definition Let \( m \) be a number. The term \( x \) is called a mathematical object of order \( m \) if there exists a type \( X \) of order \( m \) such that \( x \) has type \( X \).

Intuition There are five kinds of mathematical object: types, operations, natural numbers, assemblies, and propositions. No object falls within two of these categories.
6

PREDICATE LOGIC

LOGICAL IMPLICATION AND LOGICAL EQUIVALENCE

DEFINITION  Let \( P \) and \( Q \) be propositions. The proposition

\[
P \implies Q := \forall \Box P (\Box P (Q))
\]

is called the **conditional from \( P \) to \( Q \)**. It has **antecedent** \( P \) and **consequent** \( Q \). The proposition \( Q \implies P \) is called the **converse** of \( P \implies Q \).

6.1 / DEDUCTION THEOREM

Let \( P \) and \( Q \) be propositions. If either of the following statements is true, then so is the other:

1. If \( P \) is true, then \( Q \) is true
2. The proposition \( P \implies Q \) is true.

**Proof**  By the generalization theorem.

**Notation**  The statement \( P \) implies \( Q \) means that \( P \implies Q \) is true.

6.2 / COROLLARY

If \( P \) and \( Q \) are propositions, then the proof type of \( P \implies Q \) is

\[
\Box (P) \rightarrow \Box (Q).
\]

**Proof**  By 0.35 and 1.9.

6.3 / MODUS PONENS

Let \( P \) and \( Q \) be propositions. If \( P \) is true and \( P \) implies \( Q \), then \( Q \) is true.

**Proof**  By 5.4.

6.4 / CONDITIONAL TAUTOLEG

Let \( P \) be a proposition. Then \( P \) implies \( P \).
6.5 / Hypothetical Syllogism

Let \( P, Q, \) and \( R \) be propositions, where \( P \) implies \( Q \) and \( Q \) implies \( R \). If \( P \) is true, then \( R \) is true.

Proof Suppose that \( f \) proves \( P \implies Q \) and \( g \) proves \( Q \implies R \). Then \( g \circ f \) proves \( P \implies R \) by 1.13. Hence the result by the deduction theorem. \( \square \)

6.6 / Corollary

Let \( P, Q, \) and \( R \) be propositions. If \( P \) implies \( Q \), then

1. \( R \implies P \implies R \)
2. \( Q \implies R \implies R. \)

6.7 / Lemma

Let \( X \) be a type and \( P \) a proposition. Then

1. \( \exists X (\kappa X (P)) \) implies \( P \)
2. \( P \) implies \( \forall X (\kappa X (P)). \)

Proof By 5.9 and 5.3, respectively. \( \square \)

6.8 / Corollary

Let \( P \) and \( Q \) be propositions. If \( Q \) is true, then \( P \) implies \( Q \).

Proof By 6.7.2. \( \square \)

6.9 / Lemma

Let \( a \) be an object of type \( X \) and \( P \) a predicate on \( X \). Then

1. \( \forall_{(x:X)} P(x) \) implies \( P(a) \)
2. \( P(a) \) implies \( \exists_{(x:X)} P(x). \)

Proof This is a restatement of 5.4 and 5.8. \( \square \)

6.10 / Corollary

Let \( X \) be an inhabited type and \( P \) a proposition. Then

1. \( \forall X (\kappa X (P)) \) is equivalent to \( P \)
2. \( P \) is equivalent to \( \exists X (\kappa X (P)). \)

Definition Let \( P \) and \( Q \) be propositions. If \( P \) implies \( Q \) and \( Q \) implies \( P \), then \( P \) is said to be equivalent to \( Q \), written \( P \iff Q. \)
6.11 / THE EQUIVALENCE THEOREM
Let $P$ and $Q$ be propositions. If either of the following statements is true, then so is the other:

1. $P$ is true if and only if $Q$ is true
2. $P$ is equivalent to $Q$.

Proof By the deduction theorem. \qed

6.12 / THEOREM
Let $P$, $Q$, and $R$ be propositions.

1. The proposition $P$ is equivalent to itself
2. If $P \iff Q$, then $Q \iff P$
3. If $P \iff Q \iff R$, then $P \iff R$.

Proof By 6.4 and 6.5. \qed

6.13 / LEMMA
Let $P$, $Q$, and $R$ be propositions. If $P$ is equivalent to $Q$, then

1. $R \implies P$ is equivalent to $R \implies Q$
2. $Q \implies R$ is equivalent to $P \implies R$.

Proof By 6.6. \qed

TRANSFER OF IMPLICATION ACROSS QUANTIFIERS

6.14 / THEOREM
Let $P$ and $Q$ be predicates defined on the type $X$. Then:

1. $\forall_{(x : X)} (P(x) \implies Q(x))$ implies $\forall_{(x : X)} P(x) \implies \forall_{(x : X)} Q(x)$
2. $\forall_{(x : X)} (P(x) \implies Q(x))$ implies $\exists_{(x : X)} P(x) \implies \exists_{(x : X)} Q(x)$.

Proof of 1 Declare $f$ as a proof of the antecedent, $g$ as a proof of $\forall_X (P)$, and $x$ as an object of type $X$. Then $f(x, g(x))$ proves $Q(x)$ by 5.4. Hence the result by the generalization theorem.

Proof of 2 Declare $f$ as a proof of the antecedent and $z$ as a proof of $\exists_X (P)$. Then $f(\sigma_2(z))$ proves of $Q(\sigma_2(z))$. Hence the result by 5.8 and 6.1. \qed
6.15 / Corollary

Let $X$ be a type, $P$ a predicate on $X$, and $Q$ a proposition.

1. $\forall (x:X) \ (Q \implies P(x))$ implies $Q \implies \forall (x:X) \ P(x)$
2. $\forall (x:X) \ (P(x) \implies Q)$ implies $\exists (x:X) \ P(x) \implies Q$.

Proof By 6.14, 6.7, and 6.5.

6.16 / Theorem

Let $P$ and $Q$ be predicates defined on the type $X$. Then

$$\exists (x:X) \ (P(x) \implies Q(x)) \implies \forall (x:X) \ P(x) \implies \exists (x:X) \ Q(x).$$

Proof Declare $p$ as a proof of the antecedent and $f$ as a proof of $\forall_X (P)$. Then $f(\sigma_1(p))$ proves $P(\sigma_1(p))$ by 5.9 and 5.4, so

$$\sigma_2(p, f(\sigma_1(p)))$$

proves $Q(\sigma_1(p))$ by 5.9 and 6.3.

6.17 / Corollary

Let $X$ be a type, $P$ a predicate on $X$, and $Q$ a proposition.

1. $\exists (x:X) \ (P(x) \implies Q)$ implies $\forall (x:X) \ P(x) \implies Q$.
2. $\exists (x:X) \ (Q \implies P(x))$ implies $Q \implies \exists (x:X) \ P(x)$.

Proof By 6.16, 6.7, and 6.5.

6.18 / Corollary

Let $X$ be a type, $P$ a predicate on $X$, and $Q$ a proposition.

1. $\forall (x:X) \ (Q \implies P(x))$ is equivalent to $Q \implies \forall (x:X) \ P(x)$
2. $\forall (x:X) \ (P(x) \implies Q)$ is equivalent to $\exists (x:X) \ P(x) \implies Q$

Proof of 1 Declare $f$ as a proof of the right-hand proposition, $x$ as an object of type $X$, and $q$ as a proof of $Q$. Then $f(q, x)$ proves $P(x)$ by 5.4.

Proof of 2 Declare $g$ as a proof of the right-hand proposition, $x$ as an object of type $X$, and $p$ as a proof of $P(x)$. Then $g[x,p]$ proves $Q$ by 5.8 and 6.3. Hence the result by 6.15.
MATHEMATICAL RELATIONS

**Definition** Let \( m \) be a number, \( X \) a type of order \( m \), and \( F \) a type family of order \( m \) indexed by \( X \). An object \( R \) of type

\[
\prod_{x:X} (F_x \to \forall_m).
\]

is called a *relation of order \( m \) on \( F \).* Let \( a \) be an object of type \( A \) and \( b \) an object of type \( F_a \). If \( R(a,b) \) is true, then \( a \) is said to *have the relation \( R \) to \( b \).*

**Definition** Let \( X \) and \( Y \) be types. A relation on \( \kappa_X(Y) \) is called a *relation from \( X \) to \( Y \).* A relation from \( X \) to \( X \) is called a *relation on \( X \), or a relation between objects of type \( X \).*

**Example** The following operation is a relation of order \( m \) between types:

\[
x, y : U_m \mapsto x \subseteq y.
\]

It is called the *subtyping relation of order \( m \).*

**Example** Let \( X \) be a type. The operation defined for \( x \) and \( y \) of type \( X \) by the proposition \( x \equiv y \) is a relation on \( X \), called its *identity relation.*

**Example** Let \( X \) and \( Y \) be types and suppose that \( R \) is a relation on \( Y \). The operation defined for \( f \) and \( g \) of type \( X \rightarrow Y \) by the proposition

\[
\forall_{(x:X)} \left( R(f(x), g(x)) \right)
\]

is a relation on \( X \rightarrow Y \). It is said to be *induced by \( R \).*

**Notation** Let \( F \) be a type family indexed by \( X \) and \( R \) a relation on \( F \).

<table>
<thead>
<tr>
<th>Proposition</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>[\forall_{(x:X)} \left( \forall_{(y:F_x)} R(x, y) \right)]</td>
<td>For all ( x ) of type ( X ) and all ( y ) of type ( F_x ), the proposition ( R(x, y) ) is true</td>
</tr>
<tr>
<td>[\exists_{(x:X)} \left( \exists_{(y:F_x)} R(x, y) \right)]</td>
<td>There exist ( x ) of type ( X ) and ( y ) of type ( F_x ) such that ( R(x, y) ) is true</td>
</tr>
</tbody>
</table>

**Table 9.** Interpretation of double quantification.

The interpretation of each proposition in the left-hand column of table 9 is given in the right-hand column.
6.19 / Lemma

Let $X$ be a type and $R$ a relation on $X$.

1. $\forall (x : X) \left( \forall (y : X) R(x, y) \right)$ implies $\forall (x : X) R(x, x)$
2. $\exists (x : X) R(x, x)$ implies $\exists (x : X) \left( \exists (y : X) R(x, y) \right)$.

Proof By 6.14 and 6.9.

6.20 / Interchange of Universal Quantifiers

Let $X$ and $Y$ be types and $R$ a relation from $X$ to $Y$. Then

$$\forall (x : X) \left( \forall (y : Y) R(x, y) \right) \iff \forall (y : Y) \left( \forall (x : X) R(x, y) \right).$$

Proof Declare $f$ as a proof of the left-hand proposition. Then the operation

$$y : Y, x : X \mapsto f(x, y)$$

proves the right-hand proposition by 0.35. Since the operation

$$y : Y, x : X \mapsto R(x, y)$$

is a relation from $Y$ to $X$, the proof is completed by interchanging the roles of $X$ and $Y$. This is called a proof by symmetry.

6.21 / Interchange of Antecedents

Let $P$, $Q$ and $R$ be propositions. Then

$$P \Rightarrow (Q \Rightarrow R) \text{ is equivalent to } Q \Rightarrow (P \Rightarrow R).$$

6.22 / Interchange of Existential Quantifiers

Let $X$ and $Y$ be types and $R$ a relation from $X$ to $Y$. Then

$$\exists (x : X) \left( \exists (y : Y) R(x, y) \right) \iff \exists (y : Y) \left( \exists (x : X) R(x, y) \right).$$

Proof Either implication is proved by two applications of 5.9, followed by two applications of 5.8. Hence the result by symmetry.

6.23 / Interchange of Distinct Quantifiers

Let $X$ and $Y$ be types and $R$ a relation from $X$ to $Y$. Then

$$\exists (x : X) \left( \forall (y : Y) R(x, y) \right) \Rightarrow \forall (y : Y) \left( \exists (x : X) R(x, y) \right).$$

Proof Choose a witness $a$ of the antecedent. Declare $y$ as an object of type $Y$. It follows from 5.4 that $R(a, y)$ is true, so $\exists (x : X) R(x, y)$ is true by 5.8. Hence the result by 5.3.
6.24 / THE CLASSICAL AXIOM OF CHOICE

Let $F$ be a type family indexed by $X$ and $R$ is a relation on $F$. Then

$$\forall (x : X) \left( \exists (y : F_x) R(x, y) \right) \implies \exists (f : \Pi_X F) \left( \forall (x : X) R(x, f(x)) \right)$$

Proof. Declare $p$ as a proof of the antecedent. Let $f$ denote the operation

$$x : X \mapsto \sigma_1(p(x))$$

Then $f$ is a selection of $F$. Declare $x$ as an object of type $X$. Then $\sigma_2(p(x))$ proves $R(x, f(x))$ by 5.9. Hence the result by 5.3 and 5.8. \qed
7

MATHEMATICAL INDUCTION

ITERATION AND ADDITION

7.1 / THE PRINCIPLE OF MATHEMATICAL INDUCTION

Let $P$ be a sequence of propositions such that

$$P_0 \quad \text{and} \quad \forall (x: \mathbb{N}) (P_x \implies P_{x+1})$$

are true. Then $P_x$ is true for all natural numbers $x$.

Proof Define a proof $p$ of the result recursively by the identities

$$p_0 \equiv \Diamond P_0 \quad \text{and} \quad p_{x+1} \equiv \Diamond (P_x \implies P_{x+1})(p_x).$$

Definition An application of 7.1 is called a proof by induction on $x$. Its basis, inductive hypothesis, and inductive step are the propositions

$$P_0, \ P_x, \ \text{and} \ \forall (x: \mathbb{N}) (P_x \implies P_{x+1}),$$

respectively. Thus a proof by induction is accomplished by proving its basis and its inductive step.

7.2 / COROLLARY

If $x$ is a natural number, then $x = s^x(0)$.

Proof By induction on $x$. The basis follows from 2.3. If $x = s^x(0)$, then

$$x + 1 = s(x) = s(s^x(0)) = s^{x+1}(0).$$

7.3 / COROLLARY

Let $P$ be a sequence of propositions. If $P_0$ and $\forall (x: \mathbb{N}) P_{x+1}$ are true, then $P_x$ is true for all natural numbers $x$.

Proof By 7.1, 6.8, and 5.3.
**7.4 / Lemma**

Let \( f \) be an operation from \( X \) to \( X \). Then \( f \circ f^n \equiv f^n \circ f \) for all natural numbers \( n \).

**Proof** By induction. The base case follows from 1.20. Declare \( n \) as a natural number and suppose that \( f \circ f^n \equiv f^n \circ f \). Then

\[
f \circ f^{n+1} \equiv f \circ f \circ f^n \equiv f \circ f^n \circ f \equiv f^{n+1} \circ f.
\]

by 1.21 and the inductive hypothesis.

**7.5 / Theorem**

Let \( f \) be an operation from \( X \) to \( X \). Then \( f^{m+n} \equiv f^m \circ f^n \) for all natural numbers \( m \) and \( n \).

**Proof** By induction. Declare \( m \) and \( n \) as natural numbers, and suppose that \( f^{m+n} \equiv f^m \circ f^n \). According to 7.4 and 1.21,

\[
f \circ f^{m+n} \equiv f \circ f^m \circ f^n \equiv f^m \circ f \circ f^n.
\]

**Definition** Let \( f \) be a numerical recuser. It is said that \( f \) is commutative (resp. associative) if the identity

\[
f(x, y) = f(y, x) \quad \text{ (resp. } f(f(x, y)) = f(x) \circ f(y))
\]

is satisfied for all natural numbers \( x \) and \( y \).

**7.6 / Corollary**

Addition of natural numbers is commutative.

**Proof** Declare \( x \) and \( y \) as natural numbers. By 2.4 and 7.2,

\[
x + y = s^y \circ s^x(0) = s^{y+x}(0) = y + x.
\]

**7.7 / Corollary**

Let \( f \) be an operation from \( X \) to \( X \). Then \( f^m \circ f^n \equiv f^n \circ f^m \) for all natural numbers \( m \) and \( n \).

**7.8 / Corollary**

Addition of natural numbers is associative.

**Proof** Declare \( x, y, \) and \( z \) as natural numbers. By 2.9 and 1.21,

\[
x + (y + z) = s^x \circ s^y \circ s^z(0) = (x + y) + z.
\]

**Notation** The natural number \( x + (y + z) \) is denoted by \( x + y + z \).
CUTOFF SUBTRACTION AND FINITE TYPES

**Definition** If $m$ and $n$ are natural numbers, then $\text{pd}^m(n)$ is denoted by $m \div n$ and called the **cutoff difference of $m$ and $n$**. The operation

$$x, y : \mathbb{N} \mapsto y \div x$$

is a numerical recursor, called **cutoff subtraction**.

**Remark** If $n$ is a natural number, then $n \div 0 = n$ and $n \div 1 = \text{pd}(n)$.

**7.9 / Theorem**

*For all natural numbers $x$, $y$, and $z$,*

1. $x \div (y + z) = (x \div y) \div z$
2. $(x + y) \div y = x$
3. $(x + z) \div (y + z) = x \div y$.

**Proof** The first identity is an application of 7.4. For the second, declare $x$ as a natural number. The proof is by induction on $y$, where

$$(x + (y + 1)) \div (y + 1) = ((x + y) + 1) \div y = (x + y) \div y$$

by 7.5 7.7.1, and 2.3. Therefore 7.7.2 follows from 5.3. It follows that

$$(x + z) \div (y + z) = ((x + z) \div z) \div y = x \div y,$$

which completes the proof. \qed

**7.10 / Corollary**

*Let $x$ and $y$ be natural numbers. If $x + z = y$, then $x = y \div z$.* \qed

**7.11 / Additive Cancellation**

*Let $x$, $y$, and $z$ be natural numbers. If $x + z = y + z$, then $x = y$.* \qed

**7.12 / Theorem**

*If $x$ is a natural number, then $0 \div x = 0 = x \div x$.*

**Proof** The first identity is proved by induction, where

$$0 \div (x + 1) = (0 \div 1) \div x = 0 \div x$$

by 7.5, 7.1.1, and 2.3. The second follows from 7.7.2, since $x \equiv 0 + x$. \qed
DEFINITION Let $m$ be a natural number. The image of the operation
\[ x : \mathbb{N} \mapsto m \div x \]
is denoted by $F_m$ and called the finite type generated by $m$. It is a subtype of the natural number type. An object of type $F_m$ is called a segment of $m$.

REMARK Let $m$ and $n$ be natural numbers. It follows from 5.11 that $m$ is a segment of $n$ if and only if $m = n \div x$ for some natural number $x$.

THE STANDARD ORDERING OF THE NATURAL NUMBERS

DEFINITION Let $x$ and $y$ be natural numbers. The proposition
\[ \exists (m : \mathbb{N}) (x + m = y) \]is denoted by $x \leq y$ or by $y \geq x$.
If $x \leq y$, then $x$ is said to be less than or equal to $y$, and $y$ is said to be greater than or equal to $x$. The relation
\[ m, n : \mathbb{N} \mapsto m \leq n \]
is called the standard ordering of natural numbers.

Definition If $x \geq 1$, then $x$ is called a positive number.

7.13 / THEOREM
If $x$ and $y$ are natural numbers, then $x \leq x + y$. \hfill \square

7.14 / COROLLARY
If $x$ is a natural number, then $0 \leq x$ and $x \leq x$.

Proof By 7.13. \hfill \square

7.15 / THEOREM
Let $x$ and $y$ be natural numbers. Then $x \leq y$ if and only if $x + (y \div x) = y$.

Proof By 7.10. \hfill \square

7.16 / THEOREM
Let $x$ and $y$ be natural numbers. If $x \leq y$, then $x = y \div (y \div x)$. 

Proof By 7.15 and 7.10. \hfill \square
7.17 / COROLLARY

Let \( x, y, \) and \( z \) be natural numbers. Then \( x \leq y \) if and only if \( x + z \leq y + z \).

**Proof** By 7.15, using 7.6, 7.8, 7.9.3, and 7.11.

7.18 / LEMMA

Let \( x, y, \) and \( z \) be natural numbers. If \( x \leq y \) and \( y \leq z \), then \( x \leq z \) and

\[
(z \div y) + (y \div x) = z \div x.
\]

**Proof** According to 7.15 and 7.8,

\[
z = y + (z \div y) = (x + (y \div x)) + (z \div y)
\]

Therefore \( z \leq x \) by 7.8, and the result follows from 7.15, 7.11, and 7.6.

7.19 / COROLLARY

Let \( x \) and \( y \) be natural numbers. If \( x \leq y \) and \( y \leq x \), then \( x = y \).

**Proof** Suppose that \( x \leq y \) and \( y \leq x \). Then

\[
y \div x = (x \div x) \div (x \div y) = 0
\]

by 7.18, 7.10, and 7.12. Therefore \( x = y \) by 7.15.

7.20 / LEMMA

If \( x \) is a natural number, then \( x \div 1 \leq x \).

**Proof** By induction on \( x \), using 7.12 and 7.13.

7.21 / COROLLARY

Let \( x \) and \( y \) be natural numbers. Then \( x \div y \leq x \).

**Proof** By induction on \( y \). The base step is trivial. To prove the inductive step, declare \( y \) as a natural number and suppose that \( x \div y \leq x \). Since

\[
x \div (y + 1) = (x \div y) \div 1 \leq x \div y
\]

by 7.20, the result follows from 7.18 and the inductive hypothesis.

7.22 / THE FINITE TYPE THEOREM

Let \( x \) and \( y \) be natural numbers. Then \( x : \mathbb{F}_y \) if and only if \( x \leq y \).

**Proof** By 7.15 and 7.16.
Definition  If $x \leq y$, then the natural number $y \div x$ is denoted by $y - x$ and called the difference of $y$ and $x$.

**Definition**  Subtraction of natural numbers is the operation

$$x : \mathbb{N}, \ y : \mathbb{F}_x \mapsto x - y.$$  

**7.23 / Theorem**  
Let $x$ and $y$ be natural numbers. If $x \leq y$, then

$$x + (y - x) = y \quad \text{and} \quad x = y - (y - x).$$

**Proof**  By 7.15 and 7.16. \hfill \square

**7.24 / Corollary**  
Let $x$, $y$, and $z$ be natural numbers. If $x \leq y$ and $y \leq z$, then $y - x \leq z - x$.

**Proof**  Immediate from 7.18. \hfill \square

**7.25 / Theorem**  
Let $x$, $y$ and $z$ be natural numbers. If $z \leq y$, then

$$(x + y) - z = x + (y - z).$$

**Proof**  Since $z \leq x + y$ by 7.13, 7.6, and 7.17, it follows that

$$z + ((x + y) - z) = x + y = x + (z + (y - z)) = z + (x + (y - z))$$

by 7.23, 7.6, and 7.8. Hence the result by 7.11. \hfill \square

**7.26 / Corollary**  
Let $m$, $n$, and $x$ be natural numbers. If $m \leq x \leq m + n$, then

$$m + n = x + (n - (x - m)).$$

**Proof**  Since $x - m \leq n$ by 7.24 and 7.9.2, it follows from 7.23 and 7.8 that

$$m + n = m + ((x - m) + (n - (x - m))) = x + (n - (x - m)).$$  \hfill \square
8

FINITE SEQUENCES

EXTENSION OF AFINITE SEQUENCE

DEFINITION For natural numbers \( m \) and \( n \), the type \( +_n[F_m] \) is denoted by

\[
\mathbb{I}_m(n) \quad \text{or} \quad [m::m+n]
\]

and called the \textit{finite interval} from \( m \) to \( m+n \). It has \textit{left endpoint} \( m \), \textit{right endpoint} \( m+n \), and \textit{length} \( n \).

Remark The natural number \( x \) has type \( \mathbb{I}_m(n) \) if and only if \( x = n + (m-y) \) for some segment \( y \) of \( m \).

8.1 / COROLLARY

Let \( m, n, \) and \( x \) be natural numbers. Then \( x \) has type \( \mathbb{I}_m(n) \) if and only if

\[
n \leq x \leq n+m.
\]

Proof By 7.25 and 7.6. \qed

Notation Let \( F \) be a type family indexed by \( \mathbb{I}_m(n) \). Then

\[
\prod_{i=n}^{n+m} F_i := \prod_{i: \mathbb{I}_m(n)} F_i \quad \text{and} \quad \prod_{i=n}^{n+m} F_i := \prod_{i: \mathbb{I}_m(n)} F_i.
\]

8.2 / COROLLARY

If \( m \) is a natural number, then \( \mathbb{I}_m(0) = F_m \) and \( \mathbb{I}_0(m) = \{m\} \).

Proof By 0.24, using 8.1, 7.14, 7.18, and 3.3. \qed

Remark Note that \( [0::m] := \mathbb{I}_m(0) \) and \( [m::m] := \mathbb{I}_0(m) \).

DEFINITION Let \( m, n, \) and \( x \) be natural numbers. Define

\[
\uparrow_x(\mathbb{I}_m(n)) := +_x[\mathbb{I}_m(n)] \quad \text{and} \quad \downarrow_x(\mathbb{I}_m(n+x)) := pd^x[\mathbb{I}_m(n+x)]
\]

The former is called an \textit{upward shift} by \( n \), and the latter a \textit{downward shift} by \( n \).
**Definition** Let $f$ be an operation from $X$ to $Y$ and $g$ an operation from $Y$ to $X$. If $g \circ f \equiv 1_X$, then $g$ is said to cancel $f$.

**8.3 / Theorem**
Let $m$, $n$, and $x$ be natural numbers. Then the operations
\[ \uparrow_x (\mathbb{I}_m(n)) \quad \text{and} \quad \downarrow_x (\mathbb{I}_m(n + x)) \]
cancel each other.

*Proof* Declare $y$ as a segment of $m$. By 7.6, 7.8, 7.25, and 7.9.2,
\[
\left( \left( (n + x) + (m - y) \right) - x \right) + x = (n + x) + (m - y), \\
\left( (n + (m - y)) + x \right) - x = n + (m - y). 
\]

**8.4 / Lemma**
Let $f$ be an operation from $X$ to $Y$ and $g$ an operation from $Y$ to $X$. Suppose that $g$ cancels $f$. Then the image of $g$ is $X$.

*Proof* Declare $x$ as an object of type $X$. Then $x$ is a value of $g$, since
\[ x \equiv g(f(x)). \]

**8.5 / Corollary**
Let $m$, $n$, and $x$ be natural numbers. Then
\[ \text{im} \uparrow_x (\mathbb{I}_m(n)) = \mathbb{I}_m(n + x) \quad \text{and} \quad \text{im} \downarrow_x (\mathbb{I}_m(n + x)) = \mathbb{I}_m(n). \]

*Proof* By 8.4 and 8.3.

**Definition** Let $A$ be a subtype of $X$ and $f$ and $g$ operations on $X$. If
\[ f(x) \equiv g(x) \]
for all $x$ of type $A$, then $f$ is said to agree with $g$ on $A$.

**8.6 / Extension of a Finite Sequence**
Let $m$ be a natural number and $C$ a type family indexed by $\mathbb{F}_{m+1}$. Let $f$ be a selection of $C \mid \mathbb{F}_m$ and $c_{m+1}$ an object of type $C_{m+1}$. Construct an operation
\[ \text{ext} (f, c_{m+1}) : \prod_{i=0}^{m+1} C_i \]
that agrees with $f$ on $\mathbb{F}_m$ and has the value $c_{m+1}$ at $m+1$. 


Proof Let $D$ be the type sequence
\[ x : \mathbb{N} \mapsto C((m + 1) \div x) \]
and $g$ the selection of $D$ defined recursively by the identities
\[ g(0) = c_{m+1} \quad \text{and} \quad g(x + 1) = f(m \div x). \]
Define $\text{ext}(f, c_{m+1})$ as the finite sequence
\[ x : F_{m+1} \mapsto g((m + 1) - x). \]
Denote it by $h$. Then $h(0) = c_{m+1}$ by 7.12. Since
\[ m - x = (m + 1) - (x + 1) \]
by 7.9.3, it follows from 7.23 that $h$ agrees with $f$ on $F_m$. \qed

8.7 / Corollary
Let $m$ be a natural number, $C$ a type family indexed by $F_{m+1}$, and $f$ a selection of $C$. Then
\[ f = \text{ext}(f \mid F_m, f(m + 1)). \]
Proof Declare $x$ as a natural number. Then
\[ f((m + 1) \div x) = \text{ext}(f \mid F_m, f(m + 1))((m + 1) \div x) \]
by induction on $x$, using 8.6, 7.12, and 7.9.3. \qed

8.8 / Corollary
Let $l$ and $m$ be natural numbers and $Z$ a type family indexed by $\mathbb{I}_{m+1}(n)$. Let $x$ be a selection of $Z \mid \mathbb{I}_m(n)$ and $y$ an object of type $Z_{n+m+1}$. Construct an object
\[ (x, y) : \prod_{i=n}^{n+m+1} Z_i \]
that agrees with $x$ on $\mathbb{I}_m(n)$ and has the value $y$ at $n + m + 1$.
Proof Define $(x, y)$ as the operation
\[ \text{ext}(x \circ \uparrow_n (F_m), y) \circ \downarrow^n (\mathbb{I}_{m+1}(n)). \]
It has the required properties by 8.6 and 8.3. \qed

8.9 / Corollary
Let $l$ and $m$ be natural numbers, $Z$ a type family indexed by $\mathbb{I}_{m+1}(n)$, and $z$ a selection of $Z$. Then $z = (z \mid \mathbb{I}_m(n), z_{n+m+1})$. 
Proof. By 8.3, it is sufficient to show that
\[ z \circ \uparrow_n(F_{m+1}) = \text{ext}\left( (z \circ \uparrow_m(n) \circ \uparrow_n(F_m)), z_{n+m+1} \right). \]
Hence the result by 8.7 and 8.8. \hfill \Box

**Definition.** Let \( m \) be a positive number. An object of type \([1 :: m]\) is called an \textit{m-index}. An operation defined on \([1 :: m]\) is called an \textit{m-tuple}. If \( X \) is a type, then the type \([1 :: m] \rightarrow X\) is denoted by \( X^m \).

**8.10 / Corollary.**
Let \( n \) be a positive number, \( X \) a type, and \( A \) a subtype of \( X \). Then \( A^n \subseteq X^n \).

**Proof.** By 5.7. \hfill \Box

**Notation.** Let \( x \) be an object of type \( X \). The constant operation \( x_{(1)} \) is denoted by \( (x) \). It has type \( X^1 \) by 1.9.

**8.11 / Lemma.**
Let \( X \) be a type and \( P \) a predicate on \( X \). Then:
1. \( \forall_{(x:X)} P(x) \iff \forall_{(y:X^1)} P(y) \)
2. \( \exists_{(x:X)} P(x) \iff \exists_{(y:X^1)} P(y) \).

**Proof.** By 5.3 and 5.8. \hfill \Box

**8.12 / Lemma.**
Let \( n \) be a natural number, \( X \) a type, and \( P \) a predicate defined on \( X^{n+1} \). Then:
1. \( \forall_{(x:X^n)} (\forall_{(y:X)} P(x,y)) \iff \forall_{(z:X^{n+1})} P(z) \)
2. \( \exists_{(x:X^n)} (\exists_{(y:X)} P(x,y)) \iff \exists_{(z:X^{n+1})} P(z) \).

**Proof.** By 8.8, 5.3, and 5.8. \hfill \Box

**Ordered Pairs.**

**Definition.** A one-tuple (resp. two-tuple, three-tuple) is called an \textit{ordered singleton} (resp. \textit{ordered pair}, \textit{ordered triple}).

**Notation.** Let \( Z \) be an ordered pair of types, \( z_1 \) an object of type \( Z_1 \), and \( z_2 \) an object of type \( Z_2 \). The ordered pair \((z_1, z_2)\) is denoted by \((z_1, z_2)\).
Remark. It follows from 8.8 that \((z_1, z_2)\) is a selection of \((Z_1, Z_2)\).

8.13 / Theorem

Let \(Z\) be an ordered pair of types. If \(z\) is a selection of \(Z\), then \(z \equiv (z_1, z_2)\).

Proof. By 8.9.

Definition. Let \(Z_1\) and \(Z_2\) be types. Define

\[Z_1 \times Z_2 := \prod_{i=1}^{2} Z_i\]  
\[Z_1 + Z_2 := \prod_{i=1}^{2} Z_i.\]

The former is called the Cartesian product of \(Z_1\) and \(Z_2\), and the latter is called the coproduct of \(Z_1\) and \(Z_2\).

8.14 / Theorem

Suppose that \(X\) and \(Y\) are types. If \(x\) is an object of type \(X\) and \(y\) is an object of type \(Y\), then \((x, y)\) is an object of type \(X \times Y\).

Proof. By 8.8.

8.15 / Theorem

Let \(X\) and \(Y\) be types.

1. The term \([1, x]\) has type \(X + Y\) if and only if \(x\) has type \(X\).
2. The term \([2, y]\) has type \(X + Y\) if and only if \(y\) has type \(Y\).

Proof. By 0.27 and 4.6.

8.16 / The Fibonacci Sequence

Construct a numerical sequence \(F\) such that

\[F_0 = 0, \quad F_1 = 1, \quad \text{and} \quad F_{x+2} = F_x + F_{x+1} \]

for all natural numbers \(x\).

Proof. Define a sequence \(G\) in \(\mathbb{N} \times \mathbb{N}\) recursively by the identities

\[G_0 = (0, 1) \quad \text{and} \quad G_{x+1} = (G_{x,2}, G_{x,1} + G_{x,2}),\]

and define \(F := \pi_1 \circ G\). Declare \(x\) as a natural number. Then

\[F_x = G_{x,1}, \quad F_{x+1} = G_{x,2} \]
\[F_{x+2} = G_{x+1,2} = G_{x,1} + G_{x,2} = F_x + F_{x+1}.\]
CONJUNCTION AND DISJUNCTION

**Definition** Let $P_1$ and $P_2$ be propositions. The proposition
\[ \forall_{[1:2]}(P_1, P_2) \]
is denoted by $P_1 \land P_2$ and called the *conjunction* of $P_1$ and $P_2$.

**9.1 / Theorem**
Let $P_1$ and $P_2$ be propositions. Then the proof type of $P_1 \land P_2$ is
\[ \Box(P_1) \times \Box(P_2). \]

*Remark* A proof of $P_1 \land P_2$ is an ordered pair $(p_1, p_2)$, where $p_1$ proves $P_1$ and $p_2$ proves $P_2$.

**9.2 / The Conjunction Theorem**
The proposition $P_1 \land P_2$ is true if and only if $P_1$ and $P_2$ are both true.


**9.3 / Theorem**
Let $P$, $Q_1$, and $Q_2$ be propositions. Then $P \implies (Q_1 \land Q_2)$ is equivalent to
\[ (P \implies Q_1) \land (P \implies Q_2). \]

*Proof* By 6.18.1.

**9.4 / Exportation**
Let $P_1$, $P_2$, and $Q$ be propositions. Then
\[ P_1 \implies (P_2 \implies Q) \text{ is equivalent to } (P_1 \land P_2) \implies Q \]

*Proof of necessity* Declare $f$ as a proof of the left-hand proposition and $x$ as a proof of $P_1 \land P_2$. Then $f(x_1, x_2)$ is a proof of $Q$.  

---

66
**Proof of sufficiency** Declare \( g \) as a proof of the right-hand proposition, \( y \) as a proof of \( P_1 \), and \( z \) as a proof of \( P_2 \). Then \( g(y, z) \) is a proof of \( Q \).

### 9.5 / Theorem

Let \( P \) and \( Q \) be propositions. The proposition

\[
(P \implies Q) \land (Q \implies P)
\]

is true if and only if \( P \) is equivalent to \( Q \).

**Definition** The proposition given in 9.5 is denoted by \( P \iff Q \) and called the biconditional of \( P \) and \( Q \).

**Definition** Let \( P_1 \) and \( P_2 \) be propositions. The proposition

\[
\exists_{[1:2]}(P_1, P_2)
\]

is denoted by \( P_1 \lor P_2 \) and called the disjunction of \( P_1 \) and \( P_2 \).

**Notation** The proposition \( P_1 \lor P_2 \) is interpreted as the statement \( P_1 \) or \( P_2 \).

### 9.6 / Theorem

Let \( P_1 \) and \( P_2 \) be propositions. Then the proof type of \( P_1 \lor P_2 \) is

\[
\Box(P_1) + \Box(P_2).
\]

**Remark** A proof of \( P_1 \lor P_2 \) is an assembly which consists of a two-index \( i \) and a proof of \( P_i \).

### 9.7 / Disjunctive Generalization

Let \( P_1 \) and \( P_2 \) be propositions.

1. If \( P_1 \) is true, then \( P_1 \lor P_2 \) is true.
2. If \( P_2 \) is true, then \( P_1 \lor P_2 \) is true.

**Proof** By 8.15.

### 9.8 / Case Analysis

Let \( P_1 \), \( P_2 \), and \( Q \) be propositions. Then \( (P_1 \lor P_2) \implies Q \) is equivalent to

\[
(P_1 \implies Q) \land (P_2 \implies Q)
\]

**Proof** By 6.18.2.
9.9 / **Lemma**

Let $P$, $Q$, and $R$ be propositions. If $P$ implies $Q$, then

1. $P \land R \implies Q \land R$ and $R \land P \implies R \land Q$
2. $P \lor R \implies Q \lor R$ and $R \lor P \implies R \lor Q$.

**Proof** By 6.4 and 6.14. \qed

9.10 / **Corollary**

Let $P$, $Q$, and $R$ be propositions. If $P$ is equivalent to $Q$, then

1. $P \land R \iff Q \land R$ and $R \land P \iff R \land Q$
2. $P \lor R \iff Q \lor R$ and $R \lor P \iff R \lor Q$.

**Proof** By 9.5, 9.2, and 9.9. \qed

**The Logic of Binary Quantification**

9.11 / **Idempotent Laws**

Let $P$ be a proposition.

1. $P \land P \iff P$
2. $P \lor P \iff P$.

**Proof** By 6.10. \qed

9.12 / **Commutative Laws**

Let $P$ and $Q$ be propositions.

1. $P \land Q \iff Q \land P$
2. $P \lor Q \iff Q \lor P$.

**Proof** By 9.2, 9.7, and 9.8. \qed

**Remark** The next two theorems are proved in the same way.

9.13 / **Associative Laws**

Let $P$, $Q$, and $R$ be propositions.

1. $P \land (Q \land R) \iff (P \land Q) \land R$
2. $P \lor (Q \lor R) \iff (P \lor Q) \lor R$.

\qed
9.14 / Lemma
Let \( P \) and \( Q \) be propositions.

1. \( P \land (P \lor Q) \iff P \)
2. \( P \lor (P \land Q) \iff P. \)

9.15 / Theorem
Let \( X \) be a type and \( P_1 \) and \( P_2 \) predicates on \( X \).

1. \( \forall_{(x: X)} (P_1(x) \land P_2(x)) \iff (\forall_{(x: X)} P_1(x)) \land (\forall_{(x: X)} P_2(x)) \)
2. \( \exists_{(x: X)} (P_1(x) \lor P_2(x)) \iff (\exists_{(x: X)} P_1(x)) \lor (\exists_{(x: X)} P_2(x)). \)

Proof By 6.20 and 6.22, respectively.

9.16 / Corollary
Assume that \( P \) and \( Q \) are predicates on a type \( X \).

1. \( \forall_{(x: X)} P(x) \iff Q(x) \) implies \( \forall_{(x: X)} P(x) \iff \forall_{(x: X)} Q(x) \)
2. \( \forall_{(x: X)} P(x) \iff Q(x) \) implies \( \exists_{(x: X)} P(x) \iff \exists_{(x: X)} Q(x). \)


9.17 / Corollary
Let \( X \) be an inhabited type, \( P \) a predicate on \( X \), and \( Q \) a proposition.

1. \( Q \land (\forall_{(x: X)} P(x)) \iff (\forall_{(x: X)} Q(x)) \land (\forall_{(x: X)} P(x)) \)
2. \( Q \lor (\exists_{(x: X)} P(x)) \iff (\exists_{(x: X)} Q(x)) \lor (\exists_{(x: X)} P(x)). \)

Proof By 9.15 and 6.10.

9.18 / Distributive Laws
Let \( P, Q, \) and \( R \) be propositions.

1. \( P \land (Q \land R) \iff (P \land Q) \land (P \land R) \)
2. \( P \lor (Q \lor R) \iff (P \lor Q) \lor (P \lor R). \)

9.19 / Theorem
Let \( X \) be a type and \( P \) and \( Q \) predicates on \( X \). Then

1. \( \exists_{(x: X)} (P(x) \land Q(x)) \) implies \( \exists_{(x: X)} P(x) \land (\exists_{(x: X)} Q(x)). \)
2. \( (\forall_{(x: X)} P(x)) \lor (\forall_{(x: X)} Q(x)) \) implies \( \forall_{(x: X)} (P(x) \lor Q(x)). \)
Proof. By 6.23.

9.20 / Corollary

Let \( X \) be a type, \( P \) a predicate on \( X \), and \( Q \) a proposition.

1. \( Q \land (\exists_{(x:X)} P(x)) \) is equivalent to \( \exists_{(x:X)} (Q \land P(x)) \)
2. \( Q \lor (\forall_{(x:X)} P(x)) \) implies \( \forall_{(x:X)} (Q \lor P(x)) \).

Proof of 1. Declare \( p \) as a proof of the left-hand proposition. Then \( (p_1, \sigma_2(p_2)) \) is a proof of \( Q \land P(\sigma_1(p_2)) \). Hence the result by 5.8 and 6.23.


9.21 / Distributive Laws

Let \( P, Q, \) and \( R \) be propositions. Then

1. \( P \land (Q \lor R) \iff (P \land Q) \lor (P \land R) \)
2. \( P \lor (Q \land R) \iff (P \lor Q) \land (P \lor R) \).

Proof. The first equivalence follows from 9.20.1. The proposition

\[ ((P \lor Q) \land P) \lor ((P \lor Q) \land R) \]

is equivalent to the right-hand side of 2 by 1 and 9.10. It is equivalent to

\[ ((P \land P) \lor (Q \land P)) \lor ((P \land R) \lor (Q \land R)) \]


9.22 / Quantification over a Subtype

Let \( A \) be a subtype of \( X \) and \( P \) a predicate on \( X \).

1. \( \forall_{(x:A)} P(x) \) is equivalent to \( \forall_{(x:X)} ((x : A) \implies P(x)) \)
2. \( \exists_{(x:A)} P(x) \) is equivalent to \( \exists_{(x:X)} ((x : A) \land P(x)) \).

9.23 / Theorem

Let \( X \) be a type, \( B \) a subtype of \( X \), and \( A \) a subtype of \( B \). Then

1. \( \forall_{(x:B)} P(x) \implies \forall_{(x:A)} P(x) \)
2. \( \exists_{(x:A)} P(x) \implies \exists_{(x:B)} P(x) \)

Proof. By 5.3 and 5.8, respectively.
CONSTRUCTING SUBTYPES

DEFINING A SUBTYPE BY A PREDICATE

**Definition** Let $X$ be a type and $P$ a predicate on $X$. The type

$$\{ x : X \mid P(x) \} := \sigma_1(\exists (x : X) P(x))$$

is called the *type defined for $x$ of type $X$ by the proposition $P(x)$*. It is also denoted by $X \mid P$.

**Remark** It follows from 3.2 that $X \mid P$ is a subtype of $X$.

**10.1 / Theorem**

Let $x$ be an object of type $X$ and $P$ a predicate on $X$. Then $x$ has type $X \mid P$ if and only if $P(x)$ is true.

**Proof** The condition is necessary by 5.9 and sufficient by 5.8 and 4.4. □

**10.2 / Corollary**

If $A$ is a subtype of $X$, then $A = \{ x : X \mid x : A \}$.

**10.3 / Corollary**

Let $P$ and $Q$ be predicates on the type $X$.

1. $X \mid P \subseteq X \mid Q$ if and only if $\forall (x : X) (P(x) \implies Q(x))$
2. $X \mid P = X \mid Q$ if and only if $\forall (x : X) (P(x) \iff Q(x))$.

**Proof** By 10.1. □

**10.4 / Corollary**

Let $X$ be a type and $A$ and $B$ subtypes of $X$.

1. $A \subseteq B$ if and only if $\forall (x : X) (x : A \implies x : B)$
2. $A = B$ if and only if $\forall (x : X) (x : A \iff x : B)$.
Proof. By 10.2 and 10.3.

10.5 / Corollary
If \( n \) is a natural number, then \( \mathbb{P}_n = \{ x : \mathbb{N} \mid x \leq n \} \).

Proof. By 7.18.

Remark. This result can be generalized using 8.6.

Power Types

Definition. Let \( X \) be a type of order \( m \). An object of type

\[
\mathcal{P}_m(X) := \{ A : \mathbb{U}_m \mid A \subseteq X \}
\]

is called a subtype of \( X \) of order \( m \).

10.6 / Theorem
Let \( X \) be a type of order \( m \). Then \( \mathcal{P}_m(X) \) is a type of order \( m + 1 \).

Proof. Declare \( A \) as a type of order \( m \). Then the term \( A \subseteq X \) has type \( \mathbb{V}_{m+1} \) by 0.33 and 0.31. Since \( \mathbb{U}_m \) has type \( \mathbb{U}_{m+1} \) by 0.1.1, it follows that

\[
\prod_{(A : \mathbb{U}_m)} (A \subseteq X) : \mathbb{U}_{m+1}
\]

by 0.26 and 0.32. Hence the result by 0.22.

Definition. If \( X \) is a type, then \( \mathcal{P}_\nu(X) \) is a higher-order type. It is denoted by \( \mathcal{P}(X) \) and called the type of subtypes of \( X \) or the power type of \( X \).

Definition. Let \( K \) be a type. An operation \( A : K \rightarrow \mathcal{P}(X) \) is called a family of subtypes of \( X \) indexed by \( K \).

Notation. If \( A \) is assumed to be a family of subtypes of \( X \) indexed by \( K \), then it is understood that \( K \) and \( X \) are types.

10.7 / Lemma
Let \( X \) and \( Y \) be types. Then \( X \subseteq Y \) if and only if \( \mathcal{P}(X) \subseteq \mathcal{P}(Y) \).

Proof. By 3.8 and 3.9.
INTERSECTIONS AND UNIONS OF SUBTYPES

**Definition** Let $A$ be a family of subtypes of $X$ indexed by $K$. The types

$$
\bigcap_{\kappa: K} A_{\kappa} := \{ x : X \mid \forall_{(\kappa: K)} (x : A_{\kappa}) \}
$$

$$
\bigcup_{\kappa: K} A_{\kappa} := \{ x : X \mid \exists_{(\kappa: K)} (x : A_{\kappa}) \}
$$

are called the intersection of $A$ and the union of $A$, respectively.

**Remark** The next two theorems are corollaries of 6.18.

**10.8 / The Intersection Lemma**

Let $K$ and $X$ be types, $A$ a family of subtypes of $X$ indexed by $K$, and $B$ a subtype of $X$. Then $B \subseteq A_{\kappa}$ for all $\kappa$ of type $K$ if and only if

$$
B \subseteq \bigcap_{\kappa: K} A_{\kappa}.
$$

**10.9 / The Union Lemma**

Let $K$ and $X$ be types, $A$ a family of subtypes of $X$ indexed by $K$, and $B$ a subtype of $X$. Then $A_{\kappa} \subseteq B$ for all $\kappa$ of type $K$ if and only if

$$
\bigcup_{\kappa: K} A_{\kappa} \subseteq B.
$$

**10.10 / Lemma**

Let $A$ be a family of subtypes of $X$ indexed by $K$. If $\lambda$ has type $K$, then

$$
\bigcap_{\kappa: K} A_{\kappa} \subseteq A_{\kappa} \subseteq \bigcup_{\kappa: K} A_{\kappa} \quad \text{and} \quad \bigcap_{\kappa: \{\lambda\}} A_{\kappa} = A_{\lambda} = \bigcup_{\kappa: \{\lambda\}} A_{\kappa}.
$$

**Proof** The first proposition follows from 10.4.1, 5.4, and 5.9. The second is a corollary of the first by 10.8, 10.9, and 3.3.

**10.11 / Theorem**

Let $A$ and $B$ be families of subtypes of $X$ indexed by $K$. If $A_{\kappa} \subseteq B_{\kappa}$ for all $\kappa$ of type $K$, then

$$
\bigcap_{\kappa: K} A_{\kappa} \subseteq \bigcap_{\kappa: K} B_{\kappa} \quad \text{and} \quad \bigcup_{\kappa: K} A_{\kappa} \subseteq \bigcup_{\kappa: K} B_{\kappa}.
$$

**Proof** By 10.4.1 and 6.14.
10.12 / Corollary

Let $A$ be a family of subtypes of $X$ indexed by $K$ and $J$ a subtype of $K$. Then

$$\bigcap_{\kappa:K} A_\kappa \subseteq \bigcap_{\kappa:J} A_\kappa \quad \text{and} \quad \bigcup_{\kappa:J} A_\kappa \subseteq \bigcup_{\kappa:K} A_\kappa.$$ 

Proof. By 10.4.1 and 9.23.

Definition. Let $A$ be a family of subtypes of $X$. If

$$B \subseteq \bigcup_{\kappa:K} A_\kappa,$$

then $A$ is said to cover $B$, and is called a covering of $B$.

Remark. By the union lemma, $A$ covers $X$ if and only if $X$ is the union of $A$.

10.13 / Theorem

Let $A$ be a family of subtypes of $X$ indexed by $K$ and $J$ a family of subtypes of $K$ indexed by $L$. If $J$ covers $K$, then

1

$$\bigcap_{\kappa:K} A_\kappa = \bigcap_{\lambda:L} \left( \bigcap_{\kappa:J_\lambda} A_\kappa \right).$$

2

$$\bigcup_{\kappa:K} A_\kappa = \bigcup_{\lambda:L} \left( \bigcup_{\kappa:J_\lambda} A_\kappa \right).$$

Proof of 1. Declare $\lambda$ as an object of type $L$. It follows from 10.12 that

$$\bigcap_{\kappa:K} A_\kappa \subseteq \bigcap_{\kappa:J_\lambda} A_\kappa, \quad \text{so} \quad \bigcap_{\kappa:K} A_\kappa \subseteq \bigcap_{\lambda:L} \left( \bigcap_{\kappa:J_\lambda} A_\kappa \right)$$

by 10.8. Suppose that $x$ has the right-hand type in 1. Declare $\kappa$ as an object of type $K$. Choose $\lambda$ of type $L$ such that $\kappa$ has type $J_\lambda$. Then by 10.10,

$$x : \bigcap_{\mu:J_\lambda} A_\mu \subseteq A_\kappa.$$

Proof of 2. Declare $\lambda$ as an object of type $L$. It follows from 10.12 that

$$\bigcup_{\kappa:J_\lambda} A_\kappa \subseteq \bigcup_{\kappa:K} A_\kappa, \quad \text{so} \quad \bigcup_{\lambda:L} \left( \bigcup_{\kappa:J_\lambda} A_\kappa \right) \subseteq \bigcup_{\kappa:K} A_\kappa$$

by 10.9. Suppose that $x$ has the left-hand type in 2. Choose $\kappa$ of type $K$ and $\lambda$ of type $L$ such that $x$ has type $A_\kappa$ and $\kappa$ has type $J_\lambda$. Then by 10.10,

$$x : \bigcup_{\kappa:J_\lambda} A_\kappa \subseteq \bigcup_{\lambda:L} \left( \bigcup_{\kappa:J_\lambda} A_\kappa \right).$$
10.14 / Theorem

Let \( f \) be an operation from \( X \) to \( Y \) and \( A \) a subtype of \( X \). Then

\[
f[A] = \bigcup_{a : A} f[a] \quad \text{and} \quad X = \bigcup_{x : X} \{x\}.
\]

Proof The first identity follows from 10.4.2 and 3.10. The second follows from the first, using 3.11 and 3.12.

Notation The type \( f[A] \) is denoted by \( \{f(a) \mid a : A\} \).

10.15 / Corollary

Suppose that \( A \) is a family of subtypes of \( X \) indexed by \( L \) and \( f \) is an operation from \( K \) to \( L \). Then

\[
\bigcap_{\lambda : f[K]} A_\lambda = \bigcap_{\kappa : K} A_{f(\kappa)} \quad \text{and} \quad \bigcup_{\lambda : f[K]} A_\lambda = \bigcup_{\kappa : K} A_{f(\kappa)}.
\]

Proof According to 10.14, the operation

\[
\kappa : K \mapsto f[\kappa]
\]

is a covering of \( L \). Hence the result by 10.14 and 10.13.

10.16 / Theorem

Let \( f \) be an operation from \( X \) to \( Y \) and \( A \) a family of subtypes of \( X \) indexed by the type \( K \). Then

\[
\begin{align*}
1 & \quad f \left[ \bigcap_{\kappa : K} A_\kappa \right] \subseteq \bigcap_{\kappa : K} f[A_\kappa] \\
2 & \quad f \left[ \bigcup_{\kappa : K} A_\kappa \right] = \bigcup_{\kappa : K} f[A_\kappa].
\end{align*}
\]

Proof of 1 Let \( B \) denote the intersection of \( A \). If \( \kappa \) has type \( K \), then

\[
f[B] \subseteq f[A_\kappa]
\]

by 10.10 and 3.14. Hence the result by 10.8.


Notation Suppose that \( A \) is a family of subtypes of \( X \) indexed by \( \mathbb{I}_n(m) \), where \( l \) and \( m \) are natural numbers. Then

\[
\begin{align*}
\bigcap_{i = m}^{m+n} A_i & := \bigcap_{i : \mathbb{I}_n(m)} A_i \quad \text{and} \quad \bigcup_{i = m}^{m+n} A_i := \bigcup_{i : \mathbb{I}_n(m)} A_i.
\end{align*}
\]
**Definition** Suppose that $A_1$ and $A_2$ are subtypes of $X$. The types

$$A_1 \cap A_2 := \bigcap_{i=1}^{2} A_i \quad \text{and} \quad A_1 \cup A_2 := \bigcup_{i=1}^{2} A_i$$

are called the *intersection* and *union* of $A_1$ and $A_2$, respectively.

**10.17 / Theorem**

Let $A_1$ and $A_2$ subtypes of $X$ and $x$ an object of type $X$.

1. $x$ has type $A_1 \cap A_2$ if and only if $x$ has types $A_1$ and $A_2$
2. $x$ has type $A_1 \cup A_2$ if and only if $x$ has type $A_1$ or $A_2$.

**Proof** By 10.4.

**Remark** The laws for quantifiers proved in chapters 6 and 9 can be restated as identities for intersections and unions.

**10.18 / Theorem**

Let $x$ be a natural number. Then $x = 0$ or $x \geq 1$.

**Proof** Since $x + 1 \geq 1$, the result follows from 7.3 and 9.7.

**10.19 / Corollary**

Let $m$ be a natural number. Then $\mathbb{F}_m = \mathbb{F}_{m-1} \cup \{m\}$.

**Proof** By 7.21, 10.9, and 7.19, it is sufficient to prove that $\mathbb{F}_m \subseteq \mathbb{F}_{m-1} \cup \{m\}$. Suppose that $x$ has type $\mathbb{F}_m$. Choose a natural number $y$ such that

$$x = m \div y.$$ 

If $y = 0$, then $x = m$. If $y \geq 1$, then $y = z + 1$ for some natural number $z$, so

$$x = m \div (z + 1) = (m \div 1) \div z \leq m \div 1.$$ 

by 7.9 and 7.20. Therefore the result follows from 11.5 and 9.8.

**Definition** An object of type $\mathbb{F}_1$ is called a *binary digit*, or simply a *bit*. A sequence with domain $\mathbb{F}_1$ is called a *binary sequence*.

**10.20 / Corollary**

If $x$ is a binary digit, then $x = 0$ or $x = 1$.

**Proof** By 10.19, 8.1, and 3.3.
THE INVERSE IMAGE OF AN OPERATION

**Definition**  Let \( f \) be an operation from \( X \) to \( Y \). If \( B \) is a subtype of \( Y \), then
\[
f^{-1}[B] := \{ x : X \mid f(x) : B \}
\]
is called the **inverse image of** \( B \) **through** \( f \).

**10.21 / Theorem**
Let \( f \) be an operation from \( X \) to \( Y \) and \( B \) a family of subtypes of \( Y \) indexed by the type \( K \). Then
\[
1. \quad f^{-1}\left(\bigcap_{\kappa: K} B_{\kappa}\right) = \bigcap_{\kappa: K} f^{-1}[B_{\kappa}]
\]
\[
2. \quad f^{-1}\left(\bigcup_{\kappa: K} B_{\kappa}\right) = \bigcup_{\kappa: K} f^{-1}[B_{\kappa}].
\]

**Definition**  Let \( y \) be an object of type \( Y \). The inverse image of \( \{y\} \) through \( f \) is denoted by \( f^{-1}[y] \) and called the **inverse image of** \( y \) **through** \( f \).

**Remark**  Therefore \( x \) has type \( f^{-1}[y] \) if and only if \( f(x) = y \).

**10.22 / Corollary**
Let \( f \) be an operation from \( X \) to \( Y \) and \( B \) a subtype of \( Y \). Then
\[
f^{-1}[B] = \bigcup_{b: B} f^{-1}[b].
\]

**Proof**  By 10.14 and 10.21.

**10.23 / Theorem**
If \( f \) is an operation from \( X \) to \( Y \) and \( B \) is a subtype of \( Y \), then
\[
f[f^{-1}[B]] \subseteq B.
\]

**Proof**  If \( x \) has type \( f^{-1}[B] \), then \( f(x) \) has type \( B \).

**10.24 / Theorem**
If \( f \) is an operation from \( X \) to \( Y \) and \( A \) is a subtype of \( X \), then
\[
A \subseteq f^{-1}[f[A]].
\]

**Proof**  According to 3.10, if \( x \) has type \( A \), then \( f(x) \) has type \( f[A] \).
10.25 / COROLLARY
Let \( f \) be an operation from \( X \) to \( Y \). Suppose that \( A \) is a subtype of \( X \) and \( B \) is a subtype of \( Y \). Then \( A \subseteq f^{-1}[B] \) if and only if \( f[A] \subseteq B \).

Proof By 10.23, 10.24, and 3.9.

10.26 / COROLLARY
Let \( f \) be an operation from \( X \) to \( Y \). Then \( f^{-1}[Y] = X \).

Proof By 10.25 and 0.24, since \( f^{-1}[Y] \subseteq X \) and \( f[X] \subseteq Y \).

10.27 / THEOREM
Let \( f \) be an operation from \( X \) to \( Y \) and \( B_2 \) a subtype of \( Y \). If \( B_1 \subseteq B_2 \), then

\[
f^{-1}[B_1] \subseteq f^{-1}[B_2].
\]

Proof If \( x \) has type \( X \) and \( f(x) \) has type \( B_1 \), then \( f(x) \) has type \( B_2 \).
NEGATIVE PROPOSITIONS

ABSURDITIES AND CONTRADICTIONS

DEFINITION The proposition $0 = 1$ is denoted by $\bot$ and called falsity. A proof of $\bot$ is called an absurdity.

11.1 / EX FALSO QUODLIBET

Let $P$ be a proposition. Then $\bot$ implies $P$.

Proof Declare $x$ as an absurdity. Then $0 = 1$. It follows from 0.6 that $1 = 2$, and therefore $[1, x] = [2, x]$. Since $[1, x]$ proves $\bot \lor Q$, it follows from 8.15 that $x$ proves $Q$. □

DEFINITION Let $P$ be a proposition. The conditional $P \implies \bot$ is called the negation of $P$ and denoted by $\neg P$. It is interpreted as the statement

- It is not the case that $P$.

If the negation of $P$ is true, then $P$ is said to be false.

EXAMPLE Let $x$ and $y$ be natural numbers. Then the symbol

$$x \neq y \text{ denotes } \neg(x = y).$$

The proposition $\bot$ is false by 6.4. In other words, $1 \neq 0$.

11.2 / MODUS TOLLENS

Let $P$ and $Q$ be propositions. If $Q$ is false and $P$ implies $Q$, then $P$ is false.

Proof By 6.5. □

EXAMPLE Let $x$ be a natural number. If $x + 1 = 0$, then $1 = 0$ by 7.6, 7.9, and 7.12. Therefore $x + 1 \neq 0$ by 11.2.

DEFINITION If $P$ is a proposition, then $P \land \neg P$ is called a contradiction.
11.3 / THEOREM
Let \( P \) be a proposition. Then \( P \land \neg P \) is false.

Proof If \( x \) proves \( P \land \neg P \), then \( x_2(x_1) \) is an absurdity.

Intuition Every contradiction is false.

NOTATION The propositional universe \( \mathbb{V}_\nu \) is denoted by \( \mathbb{V} \).

11.4 / THE LAW OF NON-CONTRADICTION
The proposition \( \exists (x: \exists) (x \land \neg x) \) is false.

NOTATION Let \( x \) and \( y \) be objects of type \( X \). Then
\[
\{x, y\} := \{x\} \cup \{y\}.
\]

11.5 / THE COMPLETENESS THEOREM
Let \( P \) be a proposition. Then \( P \) is either true or false.

Proof Let the symbol \( A \) denote the binary sequence
\[
i : \mathbb{F}_1 \longmapsto \{j : \mathbb{F}_1 \mid P \lor (i = j)\}.
\]
Then each binary digit \( i \) has type \( A_i \). Declare \( X \) as an object of type \( \{A_0, A_1\} \).
Choose a proof \( p_X \) of \( (X = A_0) \lor (X = A_1) \) and let \( f \) denote the operation
\[
X : \{A_0, A_1\} \longmapsto \sigma_1(p_X).
\]
For each binary digit \( i \), the binary digit \( f(A_i) \) has type \( A_i \). Therefore
\[
\text{(11.6)} \quad P \lor \left( (f(A_0) = 0) \land (f(A_1) = 1) \right)
\]
is true by 9.21.2. It follows from 9.7 and 10.3 that \( P \) implies \( A_0 = F_1 = A_1 \).
Therefore \( P \) implies \( f(A_0) = f(A_1) \). If \( f(A_0) = 0 \) and \( f(A_1) = 1 \), then
\[
f(A_0) = f(A_1) \implies 0 = 1
\]
Hence the result by 9.9.2 and 11.2, using (11.6).

Remark This proof is due to Goodman and Myhill [1978], who expanded on the ideas of Diaconescu [1975].

11.7 / THE LAW OF EXCLUDED MIDDLE
The proposition \( \forall (x: \exists) (x \lor \neg x) \) is true.
THE LAWS FOR NEGATION

11.8 / THEOREM
Let $P$ and $Q$ be equivalent propositions. Then $P$ is false if and only if $Q$ is false.

Proof By 11.2 and 9.12.1. □

11.9 / DISJUNCTIVE SYLLOGISM
Let $P$ and $Q$ be propositions. If $P \lor Q$ is true and $P$ is false, then $Q$ is true.

Proof Suppose that $P$ is false. Then both $P$ and $Q$ imply $Q$, by 11.1 and 6.4. Therefore $P \lor Q$ implies $Q$ by 9.8. Hence the result by 9.4 and 9.12. □

11.10 / DOUBLE NEGATION
Let $P$ be a proposition. Then $P$ is true if and only if $P$ is not false.

Proof If $P$ is true, then $P$ is not false by 11.3 and 9.4. If $P$ is not false, then $P$ is true by 11.5 and 11.9. □

11.11 / THE LAW OF CONTRAPosition
Let $P$ and $Q$ be propositions. Then $P \implies Q$ is equivalent to $\neg Q \implies \neg P$.

Proof By 11.2, 11.8, and 11.10. □

Definition If $P$ and $Q$ are propositions, then the conditional $\neg Q \implies \neg P$ is called the contrapositive of $P \implies Q$.

11.12 / DE MORGAN’S LAWS
Let $X$ be a type and $P$ a predicate on $X$.

1. $\neg \forall_{(x:X)} P(x) \iff \exists_{(x:X)} \neg P(x)$
2. $\neg \exists_{(x:X)} P(x) \iff \forall_{(x:X)} \neg P(x)$.

Proof The first equivalence follows from 6.17.1 and 6.18.2. The second follows from the first, using 11.10 and 11.8. □

11.13 / COROLLARY
Let $P$ and $Q$ be propositions.

1. $\neg (P \land Q) \iff \neg P \lor \neg Q$
2. $\neg (P \lor Q) \iff \neg P \land \neg Q$. □
11.14 / Lemma

Let $P$ and $Q$ be propositions. Then $P \implies Q$ is equivalent to $Q \lor \neg P$.

Proof The proposition $Q \lor \neg P$ implies $P \implies Q$ by 9.8, using 6.8 and 11.1. The converse follows from 11.5 and 9.9.2. \hfill \Box

11.15 / Corollary

Let $P$ and $Q$ be propositions. Then $\neg (P \implies Q)$ is equivalent to $P \land \neg Q$.

Proof By 11.14, 11.13, and 9.12.1. \hfill \Box

11.16 / Corollary

Let $X$ be a type and $P$ and $Q$ predicates on $X$. Then $\exists_{(x:X)} (P(x) \implies Q(x))$ is equivalent to $\forall_{(x:X)} P(x) \implies \exists_{(x:X)} Q(x)$.

Proof By 11.8 and 11.10, it is sufficient to prove that these two propositions have equivalent negations. Hence the result by 11.12, 11.15, and 9.15. \hfill \Box

11.17 / Corollary

Let $X$ be an inhabited type, $P$ a predicate on $X$, and $Q$ a proposition.

1. $\exists_{(x:X)} (P(x) \implies Q)$ is equivalent to $\forall_{(x:X)} P(x) \implies Q$.
2. $\exists_{(x:X)} (Q \implies P(x))$ is equivalent to $Q \implies \exists_{(x:X)} P(x)$.

Proof By 11.16 and 6.10. \hfill \Box

11.18 / Lemma

Let $X$ be a type, $P$ a predicate on $X$, and $Q$ a proposition. Then $Q \lor (\forall_{(x:X)} P(x))$ is equivalent to $\forall_{(x:X)} (Q \lor P(x))$.

Proof By 6.19.1, 11.11, 11.12, and 11.13, it is sufficient to prove that $\neg Q \land \exists_{(x:X)} \neg P(x)$ implies $\exists_{(x:X)} (\neg Q \land \neg P(x))$.

Hence the result by 9.20.1. \hfill \Box

Void Types and Complements

Definition Let $X$ be a type. The type $\emptyset_X$ defined for $x$ of type $X$ by the proposition $0 = 1$ is called the void subtype of $X$. 
Remark  In other words, $x$ has type $\emptyset_X$ if and only if $0 = 1$.

11.19 / THEOREM
If $x$ is an object of type $X$, then the proposition $x : \emptyset_X$ is false.

Definition  Let $X$ be a type. If $X = \emptyset_Y$ for some type $Y$, then $X$ is said to be void. Therefore $\emptyset_X$ is void.

11.20 / COROLLARY
Let $X$ be a type and $P$ a predicate on $X$. The proposition $\forall_{(x : \emptyset_X)} P(x)$ is true, and $\exists_{(x : \emptyset_X)} P(x)$ is false.

Proof  By 5.3 and 5.8 respectively, using 11.19 and 11.1.

11.21 / COROLLARY
If $A$ is a subtype of $X$, then $\emptyset_X \subseteq A$.

Proof  By 11.20 and 5.5.

11.22 / COROLLARY
Let $X$ be a type. If $A \subseteq \emptyset_X$, then $A = \emptyset_X$.

Proof  By 11.21 and 0.24.

11.23 / COROLLARY
If $f$ is an operation from $X$ to $Y$, then

$$f[\emptyset_X] = \emptyset_Y \quad \text{and} \quad f^{-1}[\emptyset_Y] = \emptyset_X.$$ 

Proof  Let $x$ be an object of type $X$. Then $f(x)$ has type $\emptyset_Y$ if and only if $x$ has type $\emptyset_X$. Hence the result by 11.22 and 11.19.

11.24 / COROLLARY
If $A$ is a family of subtypes of $X$ indexed by $K$, then

$$\bigcap_{(\kappa : \emptyset_K)} A_\kappa = X \quad \text{and} \quad \bigcup_{(\kappa : \emptyset_K)} A_\kappa = \emptyset_X.$$

Proof  By 11.22 and 11.20.

Definition  Let $A$ be a subtype of $X$. The type

$$X - A := \{ x : X \mid \neg(x : A) \}$$

is called the complement of $A$ in $X$. 
11.25 / THEOREM

Let \( X \) be a type. Then \( X - (X - A) = A \).

*Proof* By 11.10.

11.26 / COROLLARY

Let \( X \) be a type. Then \( X - X = \emptyset_X \) and \( X - \emptyset_X = X \).

*Proof* The first identity holds by 11.3, and the second by 11.25.

11.27 / THEOREM

Let \( A_2 \) be a subtype of \( X \). If \( A_1 \subseteq A_2 \), then \( X - A_2 \subseteq X - A_1 \).

*Proof* By 11.2.

11.28 / THEOREM

Let \( X \) be a type and \( A \) a family of subtypes of \( X \) indexed by \( K \). Then

1. \[ X - \bigcap_{\kappa : K} A_\kappa = \bigcup_{\kappa : K} (X - A_\kappa) \]
2. \[ X - \bigcup_{\kappa : K} A_\kappa = \bigcap_{\kappa : K} (X - A_\kappa) \]

*Proof* By 11.12.

11.29 / THEOREM

Let \( f \) be an operation from \( X \) to \( Y \) and \( B \) a subtype of \( Y \). Then

\[ f^{-1}[Y - B] = X - f^{-1}[B] \]

*Proof* Declare \( x \) as an object of type \( X \). Then \( x \) has type \( X - f^{-1}[B] \) if and only if \( f(x) \) does not have type \( B \).
EQUIVALENCE RELATIONS

DEFINITION  Let $X$ be a type and $R$ a relation on $X$.

- If the proposition $R(x, x)$ is true for all $x$ of type $X$, then $R$ is said to be reflexive.
- If $R(x, y)$ implies $R(y, x)$ for all $x$ and $y$ of type $X$, then $R$ is said to be symmetric.
- If $R(x, y) \land R(y, z)$ implies $R(x, z)$ for all $x$, $y$, and $z$ of type $X$, then $R$ is said to be transitive.

A relation that is reflexive, symmetric, and transitive is called an equivalence relation, or simply an equivalence.

EXAMPLE  The identity relation of a type is an equivalence relation by 0.5. The subtyping relation is reflexive by 3.8 and transitive by 3.9. The standard ordering of the natural numbers is reflexive by 7.14 and transitive by 7.18.

EXAMPLE  The logical relations are defined by the following table:

<table>
<thead>
<tr>
<th>LOGICAL RELATION</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x, y : \mathbb{V} \mapsto x \Rightarrow y$</td>
<td>logical implication</td>
</tr>
<tr>
<td>$x, y : \mathbb{V} \mapsto x \Leftarrow y$</td>
<td>logical equivalence</td>
</tr>
<tr>
<td>$x, y : \mathbb{V} \mapsto x \land y$</td>
<td>logical conjunction</td>
</tr>
<tr>
<td>$x, y : \mathbb{V} \mapsto x \lor y$</td>
<td>logical disjunction</td>
</tr>
</tbody>
</table>

**Table 10.** The logical relations.

Logical equivalence is an equivalence relation by 6.12. Implication is reflexive by 6.4 and transitive by 6.5. Conjunction and disjunction are symmetric by 9.12. Conjunction is transitive by 9.2.
Equivalence relations are used to *equate* mathematical objects.

**Definition**  Let $R$ be a relation on $X$ and $A$ a subtype of $X$. The operation

$$x, y : A \mapsto R(x, y)$$

is denoted by $R \mid A$ and called the *restriction of $R$ to $A$*.

**12.1 Lemma**

*Suppose that $A$ is a subtype of $X$ and $R$ is an equivalence relation on $X$. Then $R \mid A$ is an equivalence relation.*

**Proof**  By 9.23.1, using 8.10 through 8.12. For example, to prove that $R \mid A$ is symmetric, declare $Y$ as a subtype of $X$. Then $R \mid Y$ is symmetric if and only if

$$\forall_{(x : Y)} \left( R(x_1, x_2) \implies R(x_2, x_1) \right),$$

by 8.11 and 8.12. Thus $R \mid A$ is symmetric by 8.10 and 9.23. \qed

**Definition**  Let $F$ be a type family indexed by $X$ and $R$ a relation on $F$. The *product of $R$* is defined as the relation

$$f, g : \prod_{x : X} F_x \mapsto \forall_{(x : X)} R_x(f(x), g(x))$$

**Notation**  The product of $R$ is denoted by $\prod_X (R)$ or $\prod_{(x : X)} R_x$.

**12.2 Theorem**

*Let $F$ be a type family indexed by $X$ and $R$ a relation on $F$. Suppose that $R_x$ is an equivalence relation for all $x$ of type $X$. Then $\prod_{(x : X)} R_x$ is an equivalence relation.*

**Proof of reflexivity**  Declare $f$ as an operation from $X$ to $Y$ and $x$ as an object of type $X$. Then the proposition $R_x(f(x), f(x))$ is true, since $R_x$ is reflexive.

**Proof of symmetry**  Declare $f$ and $g$ as operations from $X$ to $Y$. By 6.14.1,

$$\forall_{(x : X)} R_x(f(x), g(x)) \implies \forall_{(x : X)} R_x(g(x), f(x))$$

since $R_x$ is symmetric. In other words, $S(f, g)$ implies $S(g, f)$.

**Proof of transitivity**  Declare $f$, $g$, and $h$ as operations from $X$ to $Y$. Then the proposition $S(f, g) \land S(g, h)$ is equivalent to

$$\forall_{(x : X)} \left( R_x(f(x), g(x)) \land R_x(g(x), h(x)) \right)$$

by 9.15. This proposition implies $S(f, h)$ by 6.141, since $R_x$ is transitive. \qed
NOTATION  Let $m$ be a number and $X$ a type of order $m$. Define $\mathcal{E}_m(X)$ as

$$\{ R : X \rightarrow \mathcal{V}_m \mid R \text{ is an equivalence relation} \}.$$ 

MATHEMATICAL EQUATIONS

DEFINITION  Let $m$ be a number. A set of order $m$ is an object of type

$$\text{Set}_m := \prod_{(X : \mathcal{U}_m)} \mathcal{E}_m(X).$$

Remark  In other words, a set is an assembly consisting of a type $X$ and an equivalence relation on $X$.

DEFINITION  Let $A$ be a set. The proposition $a : \sigma_1(A)$ is written $a \in A$. If $a \in A$, then $a$ is said to be an element of $A$, or to belong to $A$, or to be in $A$, and the set $A$ is said to include $a$.

Definition  The relation $\sigma_2(A)$ is called equality on $A$, or equality of elements of $A$. Let $a$ and $b$ be elements of $A$. The proposition

$$(\sigma_2(A))(a, b) \text{ is denoted by } a =_A b$$

and called an equation. If $a =_A b$, then $a$ is said to equal $b$ in $A$. If the set $A$ is understood, then $a$ is said to equal $b$, written $a = b$.

EXAMPLE  Let $X$ be a type. The assembly consisting of $X$ and the identity relation on $X$ is a set, called the identity set of $X$.

EXAMPLE  The identity set of the natural number type is denoted by $\mathbb{N}$ and called the set of natural numbers.

Notation  For natural numbers $x$ and $y$, the symbol $x = y$ means that $x \equiv y$. In other words, $x$ and $y$ are equal elements of $\mathbb{N}$.

EXAMPLE  The following are examples of higher-order sets.

1  The identity set of $\mathcal{U}$ is denoted by $\mathcal{U}$ and called the set of types.

2  Let $E$ denote logical equivalence. The assembly $[[\mathcal{V}, E]]$ is denoted by $\mathcal{V}$ and called the set of propositions.

Notation  For types $X$ and $Y$, the symbol $X = Y$ means that $X \equiv Y$. In other words, $X$ and $Y$ are equal in the set of types.
EXAMPLE Let $A$ be a set and $X$ a subtype of $\sigma_1(A)$. Then the assembly

$$\vartheta_A(X) := [X, \sigma_2(A) \mid X]$$

is a set by 12.1. It is called the subset $X$ of $A$ or simply the set $X$, if the set $A$ is understood.

Remark By 4.7 and 0.18, the set $\sigma_1(A)$ is identical to $A$.

12.3 / THEOREM
Let $A$ be a set, $X$ a subtype of $\sigma_1(A)$ and $x_1$ and $x_2$ objects of type $X$. Then $x_1 =_A x_2$ if and only if $x_1$ is equal to $x_2$ in the set $X$.

MATHEMATICAL FUNCTIONS

DEFINITION Suppose that $A$ and $B$ are sets and $f : \sigma_1(A) \rightarrow \sigma_1(B)$. If

$$a_1 = a_2 \implies f(a_1) = f(a_2)$$

for all elements $a_1$ and $a_2$ of $A$, then $f$ is said to be a function from $A$ to $B$, or a family of elements of $B$ indexed by $A$.

NOTATION If $f$ is assumed to be a function from $A$ to $B$, then the terms $A$ and $B$ are understood to be sets.

12.4 / THEOREM
Let $X$ be a type, $B$ a set, and $f$ an operation from $X$ to $\sigma_1(B)$. Then $f$ is a function from the identity set of $X$ to $B$.

12.5 / CONSTANT FUNCTIONS
Let $A$ and $B$ be sets. If $b$ belongs to $B$, then $\kappa_A(b)$ is defined as the constant operation $\kappa_{\sigma_1(A)}(b)$, which is a function from $A$ to $B$.

Definition The function $\kappa_A(b)$ is said to be constant.

12.6 / CANONICAL INCLUSION
Let $A$ be a set If $X$ is a subtype of $\sigma_1(A)$, then the identity operation $1_X$ is a function from the set $\vartheta_A(X)$ to $A$.

Proof By 12.3.
Definition. The function $1_X$ is called the canonical inclusion of $X$ into $A$. The canonical inclusion of $\sigma_1(A)$ into $A$ is denoted by $1_A$ and called the identity function of $A$.

12.7 / Composition of Functions

Let $f$ be a function from $A$ to $B$ and $g$ a function from $B$ to $C$. Then $g \circ f$ is a function from $A$ to $C$.

Proof. By 6.5.

The Canonical Product of a Family of Sets

Definition. Let $A$ be a family of sets indexed by the set $X$. The assembly

$$\prod_{x \in X} A_x := \left[ \prod_{x \in X} \sigma_1(A_x), \prod_{x \in X} \sigma_2(A_x) \right]$$

is a set by 12.2. It is called the canonical product of $A$.

Notation. The canonical product of $A$ may be written as $\prod_X(A)$.

Definition. The following statements are defined to have the same meaning:

- For every element $x$ of $X$, let $f_x$ be an element of $A_x$
- Suppose that $\prod_X(A)$ includes the term $x \in X \mapsto f_x$.

12.8 / Theorem

Let $X$ be a set, $A$ a family of sets indexed by $X$, and $f$ and $g$ elements of the canonical product of $A$. Then $f = g$ if and only if $f(x) = g(x)$ for each element $x$ of $X$.

Definition. Suppose that $A$ and $B$ are sets, $F$ is the type of functions from $A$ to $B$, and $P$ is the set

$$\prod_{x \in A} \zeta_A(B, x).$$

Then $F \subseteq \sigma_1(P)$. The set of functions from $A$ to $B$ is defined as $\vartheta_p(F)$ and denoted by the symbol $A \rightarrow B$.

Notation. If $f$ and $g$ are functions from $A$ to $B$, then the symbol $f = g$ means that $f$ equals $g$ in the set $A \rightarrow B$. Therefore $f$ equals $g$ in $P$ by 12.3.
12.9 / Theorem

Let $f$ and $g$ be functions from $A$ to $B$. Then $f = g$ if and only if $f(a) = g(a)$ for every element $a$ of $A$.

\[ \square \]

Notation: Let $A$ and $B$ be sets. The proposition that $f$ belongs to the set $A \rightarrow B$ is denoted by the symbol $f : A \rightarrow B$.

Definition: Let $X$ be a set and $A$ a family of sets indexed by $X$. If $f$ is an element of $X$, then the evaluation operator

\[ \text{ev}_x : \left( \prod_X (\sigma_1 \circ A) \right) \rightarrow \sigma_1(A_x) \]

constructed in 1.10 is denoted by $\pi_x$. It is called the canonical projection with coordinate $x$.

Remark: If $f$ belongs to $\prod_X (A)$, then $\pi_x(f) = f(x)$. By 6.18.1,

$\pi_x : \prod_X (A) \rightarrow A_x$.

Definition: Let $A$ be a set and $P$ a predicate on $\sigma_1(A)$. The proposition

$\exists!_{(x \in A)} P(x) := \exists_{(x \in A)} \left( P(x) \land \forall_{(y \in A)} \left( P(y) \Rightarrow (x =_A y) \right) \right)$

is interpreted as the statement “There exists a unique element $x$ of $A$ such that $P(x)$ is true.”

12.10 / Theorem

Let $X$ and $Y$ be sets and $A$ a family of sets indexed by $Y$. For every element $y$ of $Y$, let $f_y$ be a function from $X$ to $A_x$. There is a unique function

$\varphi : X \rightarrow \prod_Y (A)$

such that $\pi_y \circ \varphi = f_y$ for every element $y$ of $Y$.

Proof of uniqueness: If the function

$\tilde{\varphi} : X \rightarrow \prod_Y (A)$

satisfies $\pi_y \circ \tilde{\varphi} = f_y$ for every element $y$ of $Y$, then

$\tilde{\varphi}_x(y) = f_y(x)$

for every element $x$ of $X$ and every element $y$ of $Y$. 

Proof of existence Define $\phi$ as the operation

$$x \in X, \ y \in Y \mapsto f_y(x)$$

Let $x_1$ and $x_2$ be elements of $X$ and $y$ an element of $Y$. If $x_1 = x_2$, then

$$f_y(x_1) = f_y(x_2),$$

since $f_y$ is a function. Therefore $\phi_{x_1} = \phi_{x_2}$. 

SUBSETS AND PIECEWISE DEFINITION

Definition Let $f$ be a function from $A$ to $B$. If the proposition

$$f(a_1) = f(a_2) \ \text{implies} \ a_1 = a_2$$

for all elements $a_1$ and $a_2$ of $A$, then $f$ is called an injection of $A$ into $B$, and is said to be injective. If there exists an injection of $A$ into $B$, then $A$ is called a subset of $B$.

Notation If it is assumed that $A$ is a subset of $B$, then the terms $A$ and $B$ are understood to be sets.

12.11 / Theorem

If $f$ is an injection of $A$ into $B$ and $g$ is an injection of $B$ into $C$, then $g \circ f$ is an injection of $A$ into $C$.

12.12 / Corollary

If $A$ is a subset of $B$ and $B$ is a subset of $C$, then $A$ is a subset of $C$.

Definition Let $A$ and $B$ be sets. If $\sigma_1(A) \subseteq \sigma_1(B)$ and the proposition

$$a_1 =_A a_2 \ \text{is equivalent to} \ a_1 =_B a_2$$

for all elements $a_1$ and $a_2$ of $A$, then $A$ is said to be a proper subset of $B$.

Remark If $A$ is a proper subset of $B$, then $A$ is a subset of $B$.

12.13 / Lemma

If $A$ is a set and $X$ a subtype of $\sigma_1(A)$, then $\varphi_A(X)$ is a proper subset of $A$.

12.14 / Corollary

If $A$ is a set, then $A$ is a proper subset of $A$. 


Definition  A function from a set $X$ to the set $V$ of propositions is called a propositional function on $X$. If $A$ is a proper subset of $X$ and the predicate

$$x \in X \mapsto x \in A$$

is a propositional function on $X$, then $A$ is called a saturated subset of $X$, and is said to be saturated in $X$.

12.15 / Theorem
Let $X$ be a set, $A$ a saturated subset of $X$, and $x_1$ and $x_2$ equal elements of $X$. If $x_1$ is an element of $A$, then $x_2$ is an element of $A$.

Definition  Let $K$ and $X$ be sets. Let $A$ be a family of proper subsets of $X$ indexed by $K$. If

$$\sigma_1(X) = \bigcup_{\kappa \in K} \sigma_1(A_\kappa),$$

then $A$ is called a covering of $X$, and is said to cover $X$.

Definition  If the set $A_\kappa$ is saturated in $X$ for all $\kappa$ in $K$, then $A$ is said to be saturated, or a covering of $X$ by saturated subsets.

12.16 / The Principle of Piecewise Definition
Let $K$, $X$, and $Y$ be sets. Let $A$ be a saturated covering of $X$ indexed by $K$. For each element $\kappa$ of $K$, let $f_\kappa$ be a function from $A_\kappa$ to $Y$. Suppose that

$$f_\kappa(x) = f_\lambda(x)$$

for all $\kappa$ and $\lambda$ in $K$ and all $x$ in $A_\kappa \cap A_\lambda$. There is a unique function $g$ from $X$ to $Y$ such that

$$g(x) = f_\kappa(x)$$

for every element $\kappa$ of $K$ and every element $x$ of $A_\kappa$.

Proof of uniqueness  Let $x$ be an element of $X$. Choose an element $\varphi(x)$ of $K$ such that $x$ belongs to $A_{\varphi(x)}$. If the function $g$ has the stated properties, then

$$g(x) = f(\varphi(x), x).$$

Proof of existence  Since $A$ is a covering of $X$, it follows that $\varphi : \sigma_1(X) \to Y$, and therefore $g : \sigma_1(X) \to \sigma_1(Y)$. Assume that $x_1$ and $x_2$ are equal elements of $X$. Then $x_2$ belongs to $A_{\varphi(x_1)}$ by 12.15. Consequently

$$g(x_1) = f(\varphi(x_2), x_1) = g(x_2).$$
Definition  The function $g$ is said to be \textit{piecewise defined on $A$ by $f$.}

definition Let $X$ be a set and $P$ a propositional function on $X$. The set

$$X \mid P := \varnothing_X \left( \sigma_1(X) \mid P \right)$$

is called the \textit{subset of $X$ determined by $P$. It is saturated in $X$.}

notation Let $X$ and $Y$ be sets. Let $P_1$ and $P_2$ be propositional functions
defined on $X$ such that

$$P_1(x) \lor P_2(x)$$

is true for all $x$ in $X$. For every binary digit $i$, let $f_i$ be a function from $X \mid P_i$
to $Y$. Suppose that

$$P_1(x) \land P_2(x) \implies f_1(x) = f_2(x)$$

for every element $x$ of $X$. Then the function $g$ piecewise defined on $X$ by $f$ is
denoted by the symbol

$$x \in X \mapsto \begin{cases} f_1(x) & \text{if } P_1(x), \\ f_2(x) & \text{if } P_2(x). \end{cases}$$

12.17 / corollary

\textit{Let $K$, $X$, and $Y$ be sets and $A$ a saturated covering of $X$ indexed by $K$, where}

$$\sigma_1(A_\kappa) \cap \sigma_1(A_\lambda)$$

\textit{is void for all $\kappa$ and $\lambda$ in $K$. For every element $\kappa$ of $K$, let $f_\kappa : A_\kappa \rightarrow Y$.}

\textit{There is a unique function $g : X \rightarrow Y$ such that}

$$g(x) = f_\kappa(x)$$

\textit{for every element $\kappa$ of $K$ and every element $x$ of $A_\kappa$.}

Proof  By 12.16 and 11.20.
PART III

CONCLUDING REMARKS
The definitions and postulates given in this appendix are useful for the formalization of more advanced mathematics.

In this chapter, let $m$ denote a number. The symbol $W$ denotes the constructor with index fourteen.

The term $W(m)$ is denoted by $\mathcal{W}_m$. It is called the \textit{structural universe of order }$m$.

The structural universe $\mathcal{W}_\nu$ is denoted by $\mathcal{W}$.

\textbf{0.37 / Construction of Structural Universes}

It is postulated that $\mathcal{W}_m$ is a type of order $m + 1$ and a subtype of $U_{m+1}$.

\textit{Definition} An object $S$ of type $\mathcal{W}_m$ is called a \textit{structural type of order }$m$. An object $X$ of type $S$ is called a \textit{mathematical structure of order }$m$, or simply a \textit{structure of order }$m$.

The symbol $B$ denotes the constructor with index fifteen. If $X$ is a term, then $B(X)$ is denoted by $X_b$.

\textbf{0.38 / The Base Type of a Structure}

Suppose that $S$ is a structural type of order $m$ and $X$ is a structure of type $S$. It is postulated that $X_b$ is a type of order $m$.

\textit{Definition} The type $X_b$ is called the \textit{base type of }$X$.

\textbf{0.39 / Types as Structures}

It is postulated that $U_m$ is a structural type of order $m$, and that $X_b = X$ for all types $X$ of order $m$.

\textit{Intuition} Types are the most basic mathematical structures.
0.40 / ADDITION OF STRUCTURE

Let $S$ be a structural type of order $m$, where $m$ is a number, and $\Phi$ a family of types of order $m$ indexed by $S$. It is postulated that

$$\prod_S (\Phi) : \mathbb{W}_m$$

and that $[X, \varphi]_b = X_b$

for all $X$ of type $S$ and all $\varphi$ of type $\Phi(X)$.

**Intuition** A structure $[X, \varphi]$ can be constructed from $X$ by combining it with an object $\varphi$, which is usually an operation.

**Example** Sets are mathematical structures. Indeed, if $X$ is a type and $R$ is an equivalence relation on $X$, then the set $[X, R]$ is a structure by 0.41. Its base type is $X$, by 0.41 and 0.40.

0.41 / ACCUMULATION OF STRUCTURAL TYPES

Let $m$ be a number. It is postulated that $\mathbb{W}_m$ is a structural type of order $m + 1$ and a subtype of $\mathbb{W}_{m+1}$.

**E.1 / COROLLARY**

If $S$ is a structural type of order $m$, then $S_b = S$.

**Proof** By 0.38, since $S$ is a type of order $m + 1$ by 0.36. □
PHILOSOPHICAL IMPLICATIONS

We have developed a type theory from first principles and demonstrated that it provides a natural method of formalizing ordinary mathematics. At this point, continuing to formalize mathematics becomes routine. We conclude the book by speculating about its philosophical implications.

1 / WHAT IS MATHEMATICS?
Mathematics is a system of formal definitions. The definitions provide rules for manipulating symbols, and are chosen for their usefulness in modeling natural phenomena.

2 / IS MATHEMATICS A BRANCH OF LOGIC?
No. In fact, logic is a branch of mathematics.

3 / IS MATHEMATICS INVENTED OR DISCOVERED?
The concepts of mathematics are invented, since stating a definition amounts to inventing a concept. The relationships between the concepts are discovered by deduction, and expressed as theorems.

4 / WHAT IS MATHEMATICAL TRUTH?
There are two concepts of truth in mathematics:

- An assertion is true if and only if it is a theorem.
- A proposition is true if and only if its proof type is inhabited.

These definitions provide a syntactic theory of mathematical truth. The theory explains the concept of truth as it is used in practice.

5 / IS MATHEMATICAL KNOWLEDGE CERTAIN?
Mathematical knowledge is expressed by theorems. This knowledge is acquired by applying the first principles of mathematics, which transcend dispute. Thus a complete proof cannot be doubted. However, complete proofs are too detailed
to be useful for communication or understanding. The optimal solution seems to be machine verification of proofs. This will not bring perfect certainty, since computers are not infallible. However, it will be close enough.

6 / WHAT IS A MATHEMATICAL OBJECT?

A mathematical object is a term \( a \) such that the proposition

\[ \exists (x : U) (a : X) \]

is true. The mathematical object \( a \) is then said to exist. With this definition, mathematical objects are just useful symbols.
BIBLIOGRAPHY

INSPIRATION AND MAIN INFLUENCES

BISHOP, EVERETT

BOURBAKI, NICOLAS

MARTIN-LOF, PER

THE UNIVALENT FOUNDATIONS PROGRAM
2013  *Homotopy Type Theory*. Institute for Advanced Study.

REFERENCES CONSULTED

BORCEUX, FRANCIS
1994  *Basic Category Theory*. Cambridge University Press.

DIACONESCU, RADU

GOODMAN, NOAH and MYHILL, JOHN
HOFMANN, MARTIN

KLEENE, STEPHEN C.

RIEHL, EMILY

OTHER INFLUENCES

DE BRUIJN, NICOLAAS G.

EVERETT, CALEB