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.

Email address `fct18@proton.me`

COPYRIGHT

TABLE OF CONTENTS

FIRST PRINCIPLES

FORMALIZATION OF MATHEMATICS

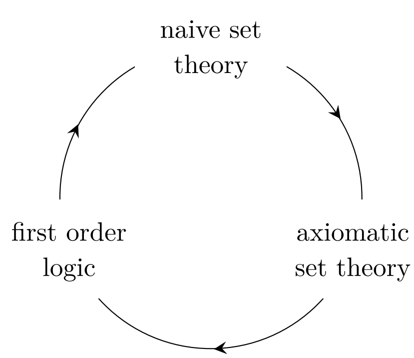
CONCLUDING REMARKS

PREFACE

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:



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

A

THE FOUNDATION OF MATHEMATICS

MATHEMATICAL ASSERTIONS

DEFINITION / *assertion*

- If x is a symbol and δ is a definition, then the statement

  "x satisfies δ"

  is defined to be an *assertion*.

- If the statements α and β are *assertions*, then the statements

  "α and β" and *"If α, then β"*

  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 α, β, and γ be assertions.

SEQUENCE OF STATEMENTS	INTERPRETATION
Suppose that α; Then β	If α, then β
Suppose that α; If β, then γ	If α and β, then γ

TABLE 1. Interpretation of sequences of assertions.

DEFINITION Let α, β, and γ be assertions. The statements defined in table 2 abbreviate sequences of assertions.

SEQUENCE OF ASSERTIONS	ABBREVIATION
If α, then β; If β, then α	*Then α if and only if β*
If α, then γ; If β, then γ	*If α or β, then γ*

TABLE 2. Abbreviations for sequences of assertions.

THE FIRST PRINCIPLES OF MATHEMATICS

INTUITION Let x denote a symbol, δ a definition, and α and β assertions. The *meanings* of the assertions

$$\text{“}x\text{ satisfies }\delta\text{,”} \quad \text{“}\alpha\text{ and }\beta\text{,”} \quad \text{and} \quad \text{“If }\alpha\text{, then }\beta\text{”}$$

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

A.1 / DEFINITION APPLICATION

Let x be a symbol and δ a definition.

1 *From “x is defined to satisfy δ,” we may conclude that “x satisfies δ.”*

2 *From “If α, then x is defined to satisfy δ,” we may conclude that “If α, then x satisfies δ.”*

A.2 / CONJUNCTION INTRODUCTION

Let α, β, and γ be assertions.

1 *From “β” and “γ,” we may conclude “β and γ.”*

2 *From “If α, then β” and “If α, then γ,” we may conclude that “If α, then β and γ.”*

A.3 / CONJUNCTION ELIMINATION

Let α and β be assertions.

1 *We may conclude that “If α and β, then α.”*

2 *We may conclude that “If α and β, then β.”*

A.4 / IMPLICATION INTRODUCTION

Let x be a symbol, δ a definition, and α an assertion. We may conclude that "If α, then α."

A.5 / IMPLICATION ELIMINATION

Let α and β be assertions. From "α" and "If α, then β," we may conclude "β."

A.6 / HYPOTHETICAL SYLLOGISM

Let α, β, and γ be assertions. From "If α, then β" and "If β, then γ," we may conclude that "If α, then γ."

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 □ 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.

B

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 $\mathsf{s}(n)$ is defined to be a *number*. It is called the *successor* of n.

EXAMPLE The symbols $\mathsf{s}(0)$ and $\mathsf{s}(\mathsf{s}(0))$ 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:

n	$\mathsf{s}(n)$	n	$\mathsf{s}(n)$	n	$\mathsf{s}(n)$	n	$\mathsf{s}(n)$
$x0$	$x1$	$x3$	$x4$	$x6$	$x7$	$x9$	$\mathsf{s}(x)0$
$x1$	$x2$	$x4$	$x5$	$x7$	$x8$		
$x2$	$x3$	$x5$	$x6$	$x8$	$x9$		

TABLE 3. Hindu-Arabic notation.

If the left-hand symbol denotes n, then the right-hand symbol denotes $\mathsf{s}(n)$. 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

$$\vee \quad \mathsf{b} \quad \mathsf{c} \quad \mathsf{s} \quad \mathsf{x} \quad \wedge(\mathsf{x}(n))$$

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 α 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 $\wedge_x$ denotes the prefix $\wedge(x)$, 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 β 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 β to $\mathcal{L}$*, or simply *β 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 β 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 ▪ is defined to be a *context* of *length* zero. It is called the *empty context*.
- The symbol $\mathsf{x}_0 : t$ is defined to be a *context* of *length* one.
- If n is a number and Γ is a *context* of *length* $\mathsf{s}(n)$, then the symbol

  $$\Gamma,\ \mathsf{x}_{\mathsf{s}(n)} : t$$

  is defined to be a *context* of *length* $\mathsf{s}(\mathsf{s}(n))$.

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 $\mathsf{s}(n)$. It may have different meanings in different contexts.

EXAMPLE Let t, u, and v be terms. The symbol

$$\mathsf{x}_0 : t,\ \ \mathsf{x}_1 : u(\mathsf{x}_0),\ \ \mathsf{x}_2 : v(\mathsf{x}_0,\mathsf{x}_1) \tag{B.2}$$

is a context of length three.

NOTATION Let Γ 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 Γ. If Γ 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 Γ a context.

- The context $\Gamma,\, x : t$ is defined to *label x with t*.
- If Γ *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 Γ, 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 $u(\mathsf{x}_0)$.

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

JUDGMENTS AND TEXTS

DEFINITION Let t and u be terms and Γ a context. The symbols

$$\Gamma \text{ ctx} \qquad \Gamma \vdash t : u \qquad t \equiv u$$

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

JUDGMENT	INTERPRETATION
Γ ctx	*The context* Γ *is well-formed*
$\Gamma \vdash t : u$	*The label u applies to t in the context* Γ
$t \equiv u$	*The term t is substitutable for u*

TABLE 4. Interpretation of judgments.

DEFINITION / *text*

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

  $$H \qquad K$$

  is defined to be a *text*, called the *conjunction of H and K*. This text is interpreted as the statement *H and K*.

NOTATION The conjunction of H and K is denoted by $H \centerdot K$. 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 Γ a context. The text

$$\Gamma \text{ ctx} \qquad \Gamma \vdash X : \mathsf{c}_0(m) \qquad \Gamma, x : X \vdash Y : \mathsf{c}_0(m)$$

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 $K \centerdot L$.

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 $H \centerdot K$ and $K \centerdot H$. □

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 $K \centerdot L$.

- 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 ▪ ctx is called the *axiom of mathematics*. The text H is said to be *valid* if it is derivable from ▪ 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 $K \mathbin{\scriptstyle\blacksquare} L$ 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 $H \mathbin{\scriptstyle\blacksquare} K$ is valid. □

B.8 / LEMMA

Let H, K, and L be texts. Then $K \mathbin{\scriptstyle\blacksquare} L$ is derivable from H if and only if $L \mathbin{\scriptstyle\blacksquare} K$ 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.

C

THE LANGUAGE OF MATHEMATICS

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 Γ ctx is defined to be the *implicit hypothesis* of S in E, where Γ 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 $H \,\textbf{.}\, K$ is defined to be the *implicit hypothesis* of S in E.
- If $H \,\textbf{.}\, K$ 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.

STATEMENT	INTERPRETATION
Conclude that L	L is derivable from H
K implies L	L is derivable from $H \centerdot K$

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 $H \centerdot K$ and L is derivable from $H \centerdot L$, 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 β 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* $\mathsf{s}(k)$ in $\wedge_x(t)$, and *bound* with *depth* k in $\beta(t)$ and $t(u)$.

DEFINITION The symbol Γ is called the *fixed context*. For the rest of the book, Γ denotes a context.

DEFINITION / *implicit context*

Let t, u, and v be terms and Δ 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 Γ is defined to be the *implicit context* of S in E.
- Otherwise, let S' be the contextual statement directly before S in E.
- If Δ 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 Δ 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.

DEFINITION Let t, u, and v be terms. In table 6, let each statement S in the left-hand column have the implicit context Δ in the entry E.

STATEMENT	INTERPRETATION
The implicit context is well-formed	The context Δ is well-formed
The label u applies to t	The label u applies to t in the context Δ

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 $\mathsf{U}(m)$ is denoted by $\mathbb{U}_m$ and called the *type universe of order m*. The statement that

- *X is a type of order m*

means that the label $\mathbb{U}_m$ 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 $\mathsf{s}(m)$*

2 *If X is a type of order m, then X is a type of order $\mathsf{s}(m)$.*

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

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

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

DEFINITION The symbol ν is called the *fixed number*. For the rest of the book, ν denotes a number.

- The phrase "of order ν" is suppressed. For example, the type universe of order ν is simply called *the type universe*. It is denoted by $\mathbb{U}$.
- The phrase "of order $\mathsf{s}(\nu)$" 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.

STATEMENT	MEANING
X is a type	X is a type of order ν
X is a higher-order type	X is a type of order $\mathsf{s}(\nu)$

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 ν unless otherwise specified.

EXAMPLE Let H and K be texts, Δ a context, and S the statement

"If K, then a has type X."

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

$$\Delta \vdash a : X$$

is derivable from $H \centerdot K$.

INTUITION 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 Δ to $\Delta, x : X$ by *declaring x as an object of type X*.
- from $\Delta, x : X$ to Δ 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*.

D

THE USE OF VARIABLES

DECLARATION OF VARIABLES

DEFINITION Let Δ be a context. If the judgment Δ ctx is true, then Δ is said to be *well-formed*.

D.1 / LEMMA

The empty context is well-formed. $\square$

INTUITION Let Δ 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

$$\frac{\Gamma \text{ ctx} \qquad \Gamma \vdash X : \mathbb{U}}{\Gamma, x : X \text{ ctx}}$$

is defined to be a postulate.

D.2 / COROLLARY

If the context Δ 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. $\square$

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

$$\frac{\Gamma \text{ ctx} \qquad \Gamma \vdash X : \mathbb{U}}{\Gamma, x : X \vdash x : X}$$

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:

$$\frac{\Gamma \text{ ctx} \qquad \Gamma \vdash X : \mathbb{U} \qquad \Gamma \vdash t : u}{\Gamma, x : X \vdash t : u} \; .$$

Intuition Since Γ 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 α 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 $t_1(u_1) \equiv t_2(u_2)$.

0.7 / SUBSTITUTION IN DESCRIPTIONS

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

$$\frac{\Gamma \text{ ctx} \qquad t_1 \equiv t_2 \qquad u_1 \equiv u_2 \qquad \Gamma \vdash t_1 : u_1}{\Gamma \vdash t_2 : u_2}$$

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

$$\Delta \vdash t : v \qquad \text{or} \qquad \Delta \vdash v : t.$$

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

INTUITION Let Δ 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 Δ (intuitively, by 0.2). Therefore u is a type by D.5. It follows from 0.2 that the context Δ, $x : u$ 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:

$$\wedge_x(c) \equiv c, \quad \wedge_x(x) \equiv \mathsf{b}_0, \quad \wedge_x\big(\mathsf{b}_n\big) \equiv \mathsf{b}_{\mathsf{s}(n)}, \quad \wedge_x\big(t(u)\big) \equiv \wedge_x\big(t, \wedge_x(u)\big).$$

Intuition The term $\wedge_x(t)$ 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:

$$\vee_a(c) \equiv c, \quad \vee_a\big(\mathsf{b}_0\big) \equiv a, \quad \vee_a\big(\mathsf{b}_{\mathsf{s}(n)}\big) \equiv \mathsf{b}_n, \quad \vee_a\big(t(u)\big) \equiv \vee_a\big(t, \vee_a(u)\big).$$

Intuition The term $\vee_a(t)$ 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

$$\begin{bmatrix} a \\ x \end{bmatrix}(t) \;:=\; \vee_a\big(\wedge_x(t)\big).$$

INTUITION According to 0.8 and 0.9, the term $\begin{bmatrix} a \\ x \end{bmatrix}(t)$ 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

$$\begin{bmatrix} a \\ x \end{bmatrix}(c) \equiv c, \quad \begin{bmatrix} a \\ x \end{bmatrix}\big(\mathsf{b}_n\big) \equiv \mathsf{b}_n, \quad \begin{bmatrix} a \\ x \end{bmatrix}(x) \equiv a.$$

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

$$\vee_a\Big(\wedge_x\big(\mathsf{b}_n\big)\Big) \equiv \vee_a\big(\mathsf{b}_{\mathsf{s}(n)}\big) \equiv \mathsf{b}_n \quad \text{and} \quad \vee_a\big(\wedge_x(x)\big) \equiv \vee_a\big(\mathsf{b}_0\big) \equiv a. \qquad \square$$

D.7 / THEOREM

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

$$\begin{bmatrix} a \\ x \end{bmatrix}\big(t(u)\big) \equiv \begin{bmatrix} a \\ x \end{bmatrix}(t)\left(\begin{bmatrix} a \\ x \end{bmatrix}(u)\right).$$

Proof It follows from 0.8 and 0.9 that

$$\vee_a\big(\wedge_x(t(u))\big) \equiv \vee_a\big(\wedge_x(t, \wedge_x(u))\big) \equiv \vee_a\big(\wedge_x(t), \vee_a(\wedge_x(u))\big). \qquad \square$$

0.10 / REFLEXIVE SUBSTITUTION

Let t be a term and x a variable. It is postulated that

$$\begin{bmatrix} x \\ x \end{bmatrix}(t) \equiv t.$$

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

$$\begin{bmatrix} a \\ x \end{bmatrix}(t) \equiv t.$$

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

$$\begin{bmatrix} a \\ x \end{bmatrix}\big(b(x)\big) \equiv b(a) \quad \text{and} \quad \begin{bmatrix} a \\ x \end{bmatrix}\big(x(b)\big) \equiv a(b).$$

Proof By D.6, D.7, and 0.11. □

0.12 / 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 $\begin{bmatrix} a \\ x \end{bmatrix}(u)$ applies to $\begin{bmatrix} a \\ x \end{bmatrix}(t)$.

Remark In other words, the symbol

$$\frac{\Gamma \text{ ctx} \qquad \Gamma \vdash X : \mathbb{U} \qquad \Gamma \vdash v : X \qquad \Gamma, x : X \vdash t : u}{\Gamma \vdash \begin{bmatrix} a \\ x \end{bmatrix}(t) : \begin{bmatrix} a \\ x \end{bmatrix}(u)}$$

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 $\begin{bmatrix} a \\ x \end{bmatrix}(Y)$ is a type.

Proof Since $\begin{bmatrix} a \\ x \end{bmatrix}(\mathbb{U}) \equiv \mathbb{U}$ by 0.12, the result follows from D.5. □

PART II

FORMALIZATION OF MATHEMATICS

1

MATHEMATICAL OPERATIONS

PRODUCT TYPES

NOTATION The symbols $\curlywedge$ and Π denote the constructors with indices one and two, respectively.

NOTATION If X is a type, then $\curlywedge_X\big(\wedge_x(y)\big)$ is denoted by

$$\curlywedge_{(x:X)}\, y \qquad \text{or by} \qquad x : X \;\longmapsto\; y.$$

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

$$\prod\nolimits_{(x:X)} Y \;:=\; \Pi\big(X, \curlywedge_{(x:X)} Y\big) : \mathbb{U}.$$

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

$$\frac{\Gamma \text{ ctx} \qquad \Gamma \vdash X : \mathbb{U} \qquad \Gamma, x : X \vdash Y : \mathbb{U}}{\Gamma \vdash \prod\nolimits_{(x:X)} Y : \mathbb{U}}\;.$$

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

$$\mathsf{s}^{\,0}(n) := n \quad \text{and} \quad \mathsf{s}^{\,\mathsf{s}(k)}(n) := \mathsf{s}\big(\mathsf{s}^{\,k}(n)\big).$$

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

- y is an object of type Y

has length n in the entry E, then x is understood to have index $\mathsf{s}^{\,k}(n)$ in E.

EXAMPLE Let X, Y, and Z be terms. If the implicit context of the statement

$$\prod\nolimits_{(x:X)}\Big(\prod\nolimits_{(y:Y)} Z\Big) : \mathbb{U}$$

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

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

1.1 / THEOREM

Let X and Y be types. Then

$$X \longrightarrow Y \;:=\; \prod\nolimits_{(x:X)} Y : \mathbb{U}.$$

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

Definition An object of type $X \longrightarrow Y$ 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 $X \longrightarrow Y$.

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 $\left[\begin{smallmatrix} a \\ x \end{smallmatrix}\right](Y)$.

Remark In other words, the inference

$$\frac{\Gamma \text{ ctx} \qquad \Gamma \vdash X : \mathbb{U} \qquad \Gamma \vdash a : X \qquad \Gamma,\, x : X \vdash Y : \mathbb{U} \qquad \Gamma \vdash f : \prod_{(x:X)} Y}{\Gamma \vdash f(a) : \left[\begin{smallmatrix} a \\ x \end{smallmatrix}\right](Y)}$$

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

$$f(a) : \begin{bmatrix} a \\ x \end{bmatrix}(Y) \equiv Y. \qquad \square$$

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 $X \longrightarrow \mathbb{U}$ is a higher-order type.

Proof The objects X and $\mathbb{U}$ are higher-order types by 0.1. Since the implicit number ν 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 $\mathbb{U}_m$ 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 Γ is an arbitrary context. $\square$

NOTATION Let t, u and v be terms and Λ a constructor. Then

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

1.5 / COROLLARY

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

Proof By 1.4 and 0.13. $\square$

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

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

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

Notation If f is *assumed* to have type $\prod_X(F)$, 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 $\left[\begin{smallmatrix}a\\x\end{smallmatrix}\right]\big(F(x)\big)$ by 0.14. Hence the result by D.5 and D.8. □

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

$$\curlywedge_{(x:X)}\, t : \prod\nolimits_{(x:X)} Y$$

Remark In other words, the inference

$$\frac{\Gamma\ \text{ctx} \qquad \Gamma \vdash X : \mathbb{U} \qquad \Gamma, x : X \vdash Y : \mathbb{U} \qquad \Gamma, x : X \vdash t : Y}{\Gamma \vdash \curlywedge_{(x:X)}\, t : \prod_{(x:X)} Y}$$

is defined to be a postulate.

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

$$x : X \ \longmapsto\ t$$

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 $\curlywedge_X(u)$ has type $\prod_{(x:X)} Y$, it is postulated that $\big(\curlywedge_X(u)\big)(a) \equiv \vee_a(u)$.

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

$$\big(\curlywedge_{(x:X)} t\big)(a) \equiv \begin{bmatrix} a \\ x \end{bmatrix}(t).$$

Proof By 0.16 and 0.17. □

Remark In particular, it follows from 0.10 that $\big(\curlywedge_{(x:X)} t\big)(x) \equiv t$.

1.8 / CONSTANT OPERATION

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

$$\varkappa_X(b) : X \longrightarrow Y$$

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

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

$$x : X \;\longmapsto\; b.$$

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

$$\varkappa_X(b, a) \equiv \begin{bmatrix} a \\ x \end{bmatrix}(b) \equiv b$$

by 1.7 and 0.11. □

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

Remark By definition, $X \longrightarrow Y \equiv \prod_X \big(\varkappa_X(Y)\big)$.

1.9 / IDENTITY OPERATION

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

Solution Define 1_X as the term

$$x : X \;\longmapsto\; x.$$

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

$$1_X(a) \equiv \begin{bmatrix} a \\ x \end{bmatrix}(x) \equiv a. \qquad \square$$

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

$$\mathrm{ev}_a : \prod\nolimits_X (F) \longrightarrow F(a)$$

such that $\mathrm{ev}_a(f) \equiv f(a)$ *if f is a selection of F.*

Proof Define ev_a as the term

$$f : \prod\nolimits_X (F) \;\longmapsto\; f(a).$$

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

$$\mathrm{ev}_a(f) \;\equiv\; \left[\begin{smallmatrix} f \\ x \end{smallmatrix}\right]\big(x(a)\big) \;\equiv\; f(a). \qquad \square$$

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

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

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

1.12 / COMPOSITION OF OPERATIONS

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

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

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

Solution Define $g \circ f$ as the term

$$x : X \;\longmapsto\; g\big(f(x)\big).$$

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

$$g \circ f(a) \;\equiv\; \left[\begin{smallmatrix} a \\ x \end{smallmatrix}\right]\Big(g\big(f(x)\big)\Big) \;\equiv\; g\big(f(a)\big).$$

by 1.7, D.7, 0.11, and D.8. $\square$

Definition The operation $g \circ f$ is called the *composition of g with f*.

1.13 / COROLLARY

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

$$g \circ f : X \longrightarrow Z.$$

Proof It follows from 1.12 and the definitions that $g \circ f$ is a selection of

$$\varkappa_Y(Z) \circ f \;\equiv\; \leftthreetimes_{(x:X)} \big(\varkappa_Y\big(Z, f(x)\big)\big) \;\equiv\; \leftthreetimes_{(x:X)} Z \;\equiv\; \varkappa_X(Z). \qquad \square$$

1.14 / COROLLARY

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

$$g \circ f : \prod\nolimits_Y (G \circ f). \qquad \square$$

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

$$h \circ (g \circ f) : \prod\nolimits_X \big(H \circ (g \circ f)\big) \quad \textit{and} \quad (h \circ g) \circ f : \prod\nolimits_X \big((H \circ g) \circ f\big)$$

Proof By 1.13 and 1.14. $\square$

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 $f \equiv g$.

Remark More precisely, f is a selection of F and g is a selection of G, where F and G are type families.

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

$$x : X \;\longmapsto\; f(x)$$

is a selection of F. Furthermore, it is identical to f.

Proof Let g denote the given formal operator. Then

$$g(x) \equiv \begin{bmatrix} x \\ x \end{bmatrix} \big(f(x)\big) \equiv f(x)$$

by 1.7, 0.10, and 0.11. Hence the result by 1.6, D.5, 0.16, and 0.18. □

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,

$$f\big(1_X(x)\big) \equiv f(x) \equiv 1_Y\big(f(x)\big).$$ □

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,

$$h\big(g \circ f(x)\big) \equiv h\Big(g\big(f(x)\big)\Big) \equiv h \circ g\big(f(x)\big).$$ □

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 *nth term*.

DEFINITION A sequence of type $\mathbb{N} \longrightarrow X$ 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* $\mathsf{s}(n)$ *is a natural number.*

Definition The object $\mathsf{s}(n)$ 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

$$\prod_{(x:\mathbb{N})} \big(C_x \longrightarrow C_{\mathsf{s}(x)}\big).$$

NOTATION The symbol R denotes the constructor with index four.

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 $\mathsf{R}(C, T, c_0)$ *is a selection of C, and that*

$$\mathsf{R}(C, T, c_0, 0) \ \equiv \ c_0.$$

Declare x as a natural number. It is postulated that

$$\mathsf{R}\big(C, T, c_0, \mathsf{s}(x)\big) \ \equiv \ T\big(x, \mathsf{R}(C, T, c_0, x)\big).$$

Intuition The sequence $f := \mathsf{R}(C, T, c_0)$ 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 $f\big(\mathsf{s}(x)\big)$ as the object $T\big(x, f(x)\big)$ of type $C_{\mathsf{s}(x)}$.

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

$$f(0) \equiv c_0 \quad \text{and} \quad f(\mathsf{s}(x)) \equiv T\big(x, f(x)\big)$$

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 $\varkappa_{\mathbb{N}}(X)$ is called a *recursor of* X. If x_0 is an object of type X, then

$$\mathsf{R}_{\varkappa}(X, T, x_0) \ := \ \mathsf{R}\big(\varkappa_{\mathbb{N}}(X), T, x_0\big).$$

This sequence is said to be *defined recursively on* X.

Remark A recursor of X is a sequence in $X \longrightarrow X$.

DEFINITION A recursor of $\mathbb{N}$ is called a *numerical recursor*.

RECURSIVE DEFINITIONS

2.1 / PREDECESSOR OPERATION

Construct a numerical sequence pd *such that*

$$\mathrm{pd}(0) \equiv 0 \quad \textit{and} \quad \mathrm{pd}\big(\mathsf{s}(n)\big) \equiv n$$

if n is a natural number.

Solution Let T denote the term

$$x : \mathbb{N} \longmapsto \varkappa_{\mathbb{N}}(x)$$

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

$$\mathrm{pd} := \mathsf{R}_{\varkappa}(\mathbb{N}, T, 0),$$

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

$$\mathrm{pd}\big(\mathsf{s}(x)\big) \equiv T\big(x, \mathrm{pd}(x)\big) \equiv \varkappa_{\mathbb{N}}\big(x, \mathrm{pd}(x)\big) \equiv x. \qquad \square$$

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

DEFINITION Let x and y be variables and X, Y, and t terms. The term

$$x : X \longmapsto (y : Y \longmapsto t) \quad \text{is denoted by} \quad x : X,\ y : Y \longmapsto t.$$

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* $\mathrm{itr}\, f$ *of* X *such that*

$$\mathrm{itr}\, f(0) \equiv 1_X \quad \textit{and} \quad \mathrm{itr}\, f\big(\mathsf{s}(n)\big) \equiv f \circ \big(\mathrm{itr}\, f(n)\big)$$

if n *is a natural number.*

Solution Let T_f denote the term

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

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

$$\mathrm{itr}\, f := \mathsf{R}_{\varkappa}(X^X, T_f, 1_X).$$

Declare x as a natural number. Then

$$\mathrm{itr}\, f\big(\mathsf{s}(x)\big) \equiv T_f\big(x, \mathrm{itr}\, f(x)\big) \equiv f \circ \big(\mathrm{itr}\, f(x)\big). \qquad \square$$

Definition Let n be a natural number. The operation

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

is called the *nth iterate of* f. With this notation,

$$f^{\,0} \equiv 1_X \quad \text{and} \quad f^{\,\mathsf{s}(n)} \equiv f \circ f^{\,n}.$$

2.3 / COROLLARY

If f is an operation from X to X, then

$$f^1 \equiv f, \qquad f^2 \equiv f \circ f, \qquad f^3 \equiv f \circ f \circ f. \qquad \square$$

NOTATION Let x be a variable, t and X terms, and $\mathcal{L}$ a list of terms. Then

$$\mathcal{L} : X,\ x : X \ \longmapsto\ t \quad \text{is denoted by} \quad \mathcal{L},\, x : X \ \longmapsto\ t.$$

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

$$+_m(0) \equiv m \quad \textit{and} \quad +_m\big(\mathsf{s}(n)\big) \equiv \mathsf{s}\big(+_m(n)\big)$$

if m and n are natural numbers.

Solution Define $+$ as the term

$$x,\, y : \mathbb{N} \ \longmapsto\ \mathsf{s}^x(y),$$

which is a numerical recursor by 0.20 and 2.2. It follows that

$$+_x(0) \equiv \mathsf{s}^0(x) \equiv x$$
$$+_x\big(\mathsf{s}(y)\big) \equiv \mathsf{s}^{\mathsf{s}(y)}(x) \equiv \mathsf{s} \circ \mathsf{s}^y(x) \equiv \mathsf{s}\big(+_x(y)\big). \qquad \square$$

DEFINITION If m and n are natural numbers, then

$$m + n \ := \ +_m(n)$$

is a natural number, called the *sum of n and m.* With this notation,

$$m + 0 \equiv m \quad \text{and} \quad m + \mathsf{s}(n) \equiv \mathsf{s}(m + n).$$

The numerical recursor $+$ is called *addition of natural numbers.*

2.5 / COROLLARY

If m and n are natural numbers, then

$$m + 1 \equiv \mathsf{s}(m) \quad \textit{and} \quad m + (n + 1) \equiv (m + n) + 1. \qquad \square$$

Example The judgment $2 + 2 \equiv 4$ is valid.

3

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

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

Definition The type $\operatorname{im} f$ is called the *image of f*. An object of type $\operatorname{im} f$ 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 $f : X \longrightarrow f(X)$.

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 $$f^* \;:=\; \mathsf{H}(X, f) \;:\; f(X) \longrightarrow X$$

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

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*.

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 $y \equiv f(x)$. Then $y \equiv f\big(f^*(x)\big)$ 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 $\varkappa_{\mathbb{N}}(x)$ 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

$$f \equiv \varkappa_{\{x\}}\big(f(x)\big) \quad \textit{and} \quad f\big(\{x\}\big) \equiv \big\{f(x)\big\}.$$ □

DEFINITION Let X be a type. The *canonical object of type* X is the term

$$\tau(X) \;:=\; \big(\varkappa_X(0)\big)^*(0).$$

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 $f := \varkappa_X(0) : X \longrightarrow \mathbb{N}$. Then $f(x) \equiv 0$, so $\tau(X) \equiv f^*\big(f(x)\big)$ 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 $f^*\big(\tau\big(f(X)\big)\big)$ has type X by 0.24 and 1.2. □

THE SUBTYPING RELATION

DEFINITION Let A and X be types. If $1_A : A \longrightarrow X$, 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 $\mathbb{U}_m$ is a subtype of $\mathbb{U}_{m+1}$ 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 $f : X \longrightarrow Y$.

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 $f \,\big|\, A$ 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*

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

Proof By 0.25. Indeed, if a is an object of type A, then

$$g \circ 1_{f[A]}\big(f \circ 1_A(a)\big) \;\equiv\; \big((g \circ f) \circ 1_A\big)(a). \qquad \square$$

3.14 / COROLLARY

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

$$f[A] \subseteq f[B] \quad \textit{and} \quad (f \mid B)[A] \equiv f[A].$$

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

$$(f \circ 1_B)[A] \;\equiv\; f\big[1_B[A]\big] \;\equiv\; f[A]$$

by the previous two corollaries. □

4

ASSEMBLIES

COPRODUCT TYPES

NOTATION The symbol $\amalg$ 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 $\coprod_X(F)$ is a type.

Definition The type $\coprod_X(F)$ is denoted by $\coprod_{(x\,:\,X)} F_x$ or

$$\coprod_{x\,:\,X} F_x$$

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

$$[x, y] \;:=\; \mathsf{A}(X, F, x, y) : \coprod\nolimits_X(F).$$

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

$$[f, g] : X \longrightarrow \coprod\nolimits_Y(G) \qquad \textit{such that} \qquad [f, g](a) \equiv \big[f(a), g(a)\big]$$

if a is an object of type X.

Proof Define $[f, g]$ as the term

$$x : X \longmapsto \big[f(x), g(x)\big].$$

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 $G\big(f(x)\big)$. Hence the result by 0.27, 0.16, and 1.7. $\square$

4.2 / CANONICAL INCLUSION

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

$$\iota_x : F_x \longrightarrow \coprod\nolimits_X (F) \quad \textit{such that} \quad \iota_x(y) \equiv [x, y]$$

for all objects x of type F_x.

Proof Define ι_x as the operation $\big[\varkappa_{F_x}(x), 1_{F_x}\big]$. $\square$

Definition The object ι_x is called the *canonical inclusion* of F_x into $\coprod_X(F)$.

SEPARATORS AND COMBINATORS

NOTATION The symbol C denotes the constructor with index eight.

NOTATION If f is *assumed* to have type $\coprod_X(F)$, then F is understood to be a type family indexed by the type X. If x, y and h are terms, then

$$h[x, y] := h\big([x, y]\big).$$

DEFINITION Let H be a type family indexed by $\coprod_X(F)$. An object S of type

$$\prod_{x : X} \prod_{y : F_x} H[x, y].$$

is called a *separator of H*. The term $\mathsf{C}(X, F, H, S)$ is denoted by

$$\coprod\nolimits_X (S) \quad \text{or} \quad \coprod_{x : X} S(x)$$

and called the *combinator of S*.

0.28 / CONSTRUCTION OF COMBINATORS

If H is a type family indexed by $\coprod_X(F)$ and S is a separator of H, it is postulated that

$$\coprod\nolimits_X (S) : \prod\nolimits_{\coprod_X (F)} (H).$$

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

$$\Big(\coprod_X(S)\Big)[x,y] \;\equiv\; S(x,y).$$

Definition Given a selection h of H, the operation

$$x : X,\, y : F_x \;\longmapsto\; h[x,y]$$

is denoted by 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

$$\operatorname{sep}\Big(\coprod_X(S)\Big) \;\equiv\; S. \qquad \square$$

4.4 / FIRST COMPONENT OF AN ASSEMBLY

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

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

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

Solution Define σ_1 as the combinator

$$\coprod_{x:X}\big(\varkappa_{F_x}(x)\big). \qquad \square$$

4.5 / SECOND COMPONENT OF AN ASSEMBLY

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

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

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

Solution Define σ_2 as the combinator

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

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 $\coprod_X(F)$. It is postulated that

$$h \equiv \coprod_{x\,:\,X} \big(\operatorname{sep} h\big)(x).$$

4.6 / COROLLARY

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

4.7 / THEOREM

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

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

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 $z \equiv [z_1, z_2]$.

5

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 $\mathsf{V}(m)$ is denoted by $\mathbb{V}_m$.

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

0.31 / CONSTRUCTION OF PROPOSITIONAL UNIVERSES

Let m be a number. It is postulated that $\mathbb{V}_m$ is a type of order $m+1$ and a subtype of $\mathbb{V}_{m+1}$.

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 $\mathsf{S}(X, Y)$ is a proposition and that its proof type is a subtype of $X \longrightarrow Y$. If X is a subtype of Y, it is postulated that 1_X proves $\mathsf{S}(X, Y)$.

Definition The proposition $\mathsf{S}(X,Y)$ is called a *subtyping proposition.*

0.34 / UNIQUENESS OF SUBTYPING PROOFS

Let X and Y be types, and p a proof of $\mathsf{S}(X,Y)$. It is postulated that $p \equiv 1_X$.

Intuition If the proposition $\mathsf{S}(X,Y)$ 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 $\mathsf{S}(X,Y)$ 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 $1_{\{x\}} \equiv \varkappa_{\{x\}}(x)$. 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:

PROPOSITION	INTERPRETATION	NOTATION
$\mathsf{S}(X,Y)$	*X is a subtype of Y*	$X \subseteq Y$
$\mathsf{S}(\{x\},A)$	*x is an object of type A*	$x : A$
$\mathsf{S}(\{x\},\{y\})$	*x is substitutable for y*	$x \equiv y$

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 $\mathbb{V}_m$ 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

$$x : X \longmapsto x : A \quad \text{and} \quad x : X \longmapsto f(x) \equiv g(x)$$

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

$$\square\big(\forall_{(x:X)} P(x)\big) \;=\; \prod\nolimits_{(x:X)} \square\big(P(x)\big).$$

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

$$\prod_{x:X} F_x \subseteq \prod_{x:X} G_x \quad \textit{and} \quad \coprod_{x:X} F_x \subseteq \coprod_{x:X} G_x.$$

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 $\coprod_X(F)$. 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

$$\square\big(\exists_{(x:X)} P(x)\big) = \coprod\nolimits_{(x:X)} \square\big(P(x)\big).$$

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. □

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

$$\sigma_1\big(\diamond\big(\exists_{(x:X)} P(x)\big)\big)$$

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 $P\big(\varepsilon_X(P)\big)$ is true.

Proof The condition is necessary by 4.4 and 4.5. It is sufficient by 5.8 □

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. □

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. □

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

$$P \implies Q := \forall_{\square P}\big(\varkappa_{\square P}(Q)\big)$$

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

$$\square(P) \longrightarrow \square(Q).$$

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 TAUTOLOGY

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. □

6.6 / COROLLARY

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

1 $R \implies P$ *implies* $R \implies Q$

2 $Q \implies R$ *implies* $P \implies R$. □

6.7 / LEMMA

Let X be a type and P a proposition. Then

1 $\exists_X\big(\varkappa_X(P)\big)$ *implies* P

2 P *implies* $\forall_X\big(\varkappa_X(P)\big)$.

Proof By 5.9 and 5.3, respectively. □

6.8 / COROLLARY

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

Proof By 6.7.2. □

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. □

6.10 / COROLLARY

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

1 $\forall_X\big(\varkappa_X(P)\big)$ *is equivalent to* P

2 P *is equivalent to* $\exists_X\big(\varkappa_X(P)\big)$. □

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. □

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. □

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. □

TRANSFER OF IMPLICATION ACROSS QUANTIFIERS

6.14 / THEOREM

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

1 $\forall_{(x:X)} \big(P(x) \implies Q(x)\big)$ *implies* $\forall_{(x:X)} P(x) \implies \forall_{(x:X)} Q(x)$

2 $\forall_{(x:X)} \big(P(x) \implies Q(x)\big)$ *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 $f\big(x, g(x)\big)$ 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 $f\big(\sigma_2(z)\big)$ proves of $Q\big(\sigma_2(z)\big)$. Hence the result by 5.8.And 6.1. □

6.15 / COROLLARY

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

1 $\forall_{(x:X)} \big(Q \implies P(x)\big)$ *implies* $Q \implies \forall_{(x:X)} P(x)$

2 $\forall_{(x:X)} \big(P(x) \implies Q\big)$ *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

$$\exists_{(x:X)} \big(P(x) \implies Q(x)\big) \text{ implies } \forall_{(x:X)} P(x) \implies \exists_{(x:X)} Q(x).$$

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

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

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

6.17 / COROLLARY

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

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

2 $\exists_{(x:X)} \big(Q \implies P(x)\big)$ *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 $\forall_{(x:X)} \big(Q \implies P(x)\big)$ *is equivalent to* $Q \implies \forall_{(x:X)} P(x)$

2 $\forall_{(x:X)} \big(P(x) \implies Q\big)$ *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

$$\prod_{x:X} (F_x \longrightarrow \mathbb{V}_m).$$

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 $\varkappa_X(Y)$ 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:

$$x,\, y : \mathbb{U}_m \;\longmapsto\; x \subseteq y.$$

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 $X \longrightarrow Y$ by the proposition

$$\forall_{(x:X)} \big(R\big(f(x), g(x)\big)\big)$$

is a relation on $X \longrightarrow Y$. It is said to be *induced by R.*

NOTATION Let F be a type family indexed by X and R a relation on F.

PROPOSITION	INTERPRETATION
$\forall_{(x:X)} \big(\forall_{(y:F_x)} R(x,y)\big)$	*For all x of type X and all y of type F_x, the proposition $R(x,y)$ is true*
$\exists_{(x:X)} \big(\exists_{(y:F_x)} R(x,y)\big)$	*There exist x of type X and y of type F_x such that $R(x,y)$ is true*

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 $\forall_{(x:X)} \big(\forall_{(y:X)} R(x,y)\big)$ *implies* $\forall_{(x:X)} R(x,x)$

2 $\exists_{(x:X)} R(x,x)$ *implies* $\exists_{(x:X)} \big(\exists_{(y:X)} R(x,y)\big)$.

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

$$\forall_{(x:X)} \big(\forall_{(y:Y)} R(x,y)\big) \iff \forall_{(y:Y)} \big(\forall_{(x:X)} R(x,y)\big).$$

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

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

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

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

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

$$P \implies (Q \implies R) \quad \textit{is equivalent to} \quad Q \implies (P \implies R). \qquad \square$$

6.22 / INTERCHANGE OF EXISTENTIAL QUANTIFIERS

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

$$\exists_{(x:X)} \big(\exists_{(y:Y)} R(x,y)\big) \iff \exists_{(y:Y)} \big(\exists_{(x:X)} R(x,y)\big).$$

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

$$\exists_{(x:X)} \big(\forall_{(y:Y)} R(x,y)\big) \implies \forall_{(y:Y)} \big(\exists_{(x:X)} R(x,y)\big).$$

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

$$\forall_{(x:X)} \big(\exists_{(y:F_x)} R(x,y)\big) \implies \exists_{\left(f:\prod_X(F)\right)} \Big(\forall_{(x:X)} R\big(x, f(x)\big)\Big)$$

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

$$x : X \longmapsto \sigma_1\big(p(x)\big)$$

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

7

MATHEMATICAL INDUCTION

ITERATION AND ADDITION

7.1 / THE PRINCIPLE OF MATHEMATICAL INDUCTION

Let P be a sequence of propositions such that

$$P_0 \quad \textit{and} \quad \forall_{(x\,:\,\mathbb{N})}\,(P_x \implies P_{x+1})$$

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

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

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

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

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

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 $x = \mathsf{s}^x(0)$.

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

$$x + 1 = \mathsf{s}(x) = \mathsf{s}\big(\mathsf{s}^x(0)\big) = \mathsf{s}^{x+1}(0). \qquad \square$$

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. $\square$

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

$$f \circ f^{n+1} \equiv f \circ f \circ f^n \equiv f \circ f^n \circ f \equiv f^{n+1} \circ f. \qquad \square$$

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,

$$f \circ f^{m+n} \equiv f \circ f^m \circ f^n \equiv f^m \circ f \circ f^n. \qquad \square$$

DEFINITION Let f be a numerical recursor. It is said that f is *commutative* (resp. *associative*) if the identity

$$f(x,y) = f(y,x) \qquad \Big(\text{resp. } \; f\big(f(x,y)\big) = f(x) \circ f(y)\Big)$$

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,

$$x + y = \mathsf{s}^y \circ \mathsf{s}^x(0) = \mathsf{s}^{y+x}(0) = y + x. \qquad \square$$

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. $\square$

7.8 / COROLLARY

Addition of natural numbers is associative.

Proof Declare x, y, and z as natural numbers. By 2.9 and 1.21,

$$x + (y + z) = \mathsf{s}^x \circ \mathsf{s}^y \circ \mathsf{s}^z(0) = (x + y) + z. \qquad \square$$

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 $\mathrm{pd}^m(n)$ is denoted by $m \dot{-} n$ and called the *cutoff difference of* m *and* n. The operation

$$x, y : \mathbb{N} \longmapsto y \dot{-} x$$

is a numerical recursor, called *cutoff subtraction*.

REMARK If n is a natural number, then $n \dot{-} 0 = n$ and $n \dot{-} 1 = \mathrm{pd}(n)$.

7.9 / THEOREM

For all natural numbers x, y, and z,

1 $x \dot{-} (y+z) = (x \dot{-} y) \dot{-} z$

2 $(x+y) \dot{-} y = x$

3 $(x+z) \dot{-} (y+z) \;=\; x \dot{-} y.$

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

$$\big(x + (y+1)\big) \dot{-} (y+1) \;=\; \Big(\big((x+y)+1\big) \dot{-} 1\Big) \dot{-} y \;=\; (x+y) \dot{-} y$$

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

$$(x+z) \dot{-} (y+z) \;=\; \big((x+z) \dot{-} z\big) \dot{-} y \;=\; x \dot{-} y,$$

which completes the proof. □

7.10 / COROLLARY

Let x and y be natural numbers. If $x + z = y$, then $x = y \dot{-} z$. □

7.11 / ADDITIVE CANCELLATION

Let x, y, and z be natural numbers. If $x + z = y + z$, then $x = y$. □

7.12 / THEOREM

If x is a natural number, then $0 \dot{-} x = 0 = x \dot{-} x$.

Proof The first identity is proved by induction, where

$$0 \dot{-} (x+1) \;=\; (0 \dot{-} 1) \dot{-} x \;=\; 0 \dot{-} x$$

by 7.5, 7.1.1, and 2.3. The second follows from 7.7.2, since $x \equiv 0 + x$. □

DEFINITION Let m be a natural number. The image of the operation

$$x : \mathbb{N} \longmapsto m \dot{-} x$$

is denoted by $\mathbb{F}_m$ and called the *finite type generated by* m. It is a subtype of the natural number type. An object of type $\mathbb{F}_m$ 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 $m = n \dot{-} x$ for some natural number x.

THE STANDARD ORDERING OF THE NATURAL NUMBERS

DEFINITION Let x and y be natural numbers. The proposition

$$\exists_{(m:\mathbb{N})}(x + m = y) \quad \text{is denoted by} \quad x \le y \quad \text{or by} \quad y \ge x.$$

If $x \le y$, 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

$$m, n : \mathbb{N} \longmapsto m \le n$$

is called the *standard ordering of natural numbers.*

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

7.13 / THEOREM

If x and y are natural numbers, then $x \le x + y$. □

7.14 / COROLLARY

If x is a natural number, then $0 \le x$ and $x \le x$.

Proof By 7.13. □

7.15 / THEOREM

Let x and y be natural numbers. Then $x \le y$ if and only if $x + (y \dot{-} x) = y$.

Proof By 7.10. □

7.16 / THEOREM

Let x and y be natural numbers. If $x \le y$, then $x = y \dot{-} (y \dot{-} x)$.

Proof By 7.15 and 7.10. □

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

$$(z \dot{-} y) + (y \dot{-} x) \ = \ z \dot{-} x.$$

Proof According to 7.15 and 7.8,

$$z \ = \ y + (z \dot{-} y) \ = \ \big(x + (y \dot{-} x)\big) + (z \dot{-} y)$$

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

$$y \dot{-} x = (x \dot{-} x) \dot{-} (x \dot{-} y) = 0$$

by 7.18, 7.10, and 7.12. Therefore $x = y$ by 7.15. □

7.20 / LEMMA

If x is a natural number, then $x \dot{-} 1 \leq x$.

Proof By induction on x, using 7.12 and 7.13. □

7.21 / COROLLARY

Let x and y be natural numbers. Then $x \dot{-} y \leq x$.

Proof By induction on y. The base step is trivial. To prove the inductive step, declare y as a natural number and suppose that $x \dot{-} y \leq x$. Since

$$x \dot{-} (y + 1) \ = \ (x \dot{-} y) \dot{-} 1 \leq x \dot{-} y$$

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 $y \dot{-} x$ is denoted by $y - x$ and called the *difference of* y *and* x.

DEFINITION *Subtraction of natural numbers* is the operation

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

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. □

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. □

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

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

by 7.23, 7.6, and 7.8. Hence the result by 7.11. □

7.26 / COROLLARY

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

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

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

$$m + n = m + \Big((x - m) + \big(n - (x - m)\big)\Big) = x + \big(n - (x - m)\big). \quad \square$$

8

FINITE SEQUENCES

EXTENSION OF A FINITE SEQUENCE

DEFINITION For natural numbers m and n, the type $+_n[\mathbb{F}_m]$ is denoted by

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

and called the *finite interval from* m *to* $m+n$. It has *left endpoint* m, *right endpoint* $m+n$, and *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. □

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

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

8.2 / COROLLARY

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

Proof By 0.24, using 8.1, 7.14, 7.18, and 3.3. □

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

$$\Big\uparrow_x \big(\mathbb{I}_m(n)\big) \;:=\; +_x \Big| \, \mathbb{I}_m(n) \quad \text{and} \quad \Big\downarrow^x \big(\mathbb{I}_m(n+x)\big) \;:=\; \mathrm{pd}^x \Big| \, \mathbb{I}_m(n+x)$$

The former is called an *upward shift by* n, and the latter a *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

$$\big\uparrow_x \big(\mathbb{I}_m(n)\big) \quad \textit{and} \quad \big\downarrow^x \big(\mathbb{I}_m(n+x)\big)$$

cancel each other.

Proof Declare y as a segment of m. By 7.6, 7.8, 7.25, and 7.9.2,

$$\Big(\big((n+x)+(m-y)\big)-x\Big)+x \;=\; (n+x)+(m-y),$$
$$\Big(\big(n+(m-y)\big)+x\Big)-x \;=\; n+(m-y).$$ □

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

$$x \equiv g\big(f(x)\big).$$ □

8.5 / COROLLARY

Let m, n, and x be natural numbers. Then

$$\operatorname{im}\big\uparrow_x \big(\mathbb{I}_m(n)\big) \;=\; \mathbb{I}_m(n+x)\big) \quad \textit{and} \quad \operatorname{im}\big\downarrow^x \big(\mathbb{I}_m(n+x)\big) \;=\; \mathbb{I}_m(n).$$

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 $C \,\big|\, \mathbb{F}_m$ and c_{m+1} an object of type C_{m+1}. Construct an operation

$$\operatorname{ext}(f, c_{m+1}) : \prod_{i=0}^{m+1} C_i$$

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

$$x : \mathbb{N} \longmapsto C\big((m+1) \dot{-} x\big)$$

and g the selection of D defined recursively by the identities

$$g(0) = c_{m+1} \quad \text{and} \quad g(x+1) = f(m \dot{-} x).$$

Define ext $\big(f, c_{m+1}\big)$ as the finite sequence

$$x : \mathbb{F}_{m+1} \longmapsto g\big((m+1) - x\big).$$

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 $\mathbb{F}_m$. □

8.7 / COROLLARY

Let m be a natural number, C a type family indexed by $\mathbb{F}_{m+1}$, and f a selection of C. Then $f = \operatorname{ext}\big(f \,\big|\, \mathbb{F}_m, f(m+1)\big)$.

Proof Declare x as a natural number. Then

$$f\big((m+1) \dot{-} x\big) \;=\; \operatorname{ext}\big(f \,\big|\, \mathbb{F}_m, f(m+1)\big)\big((m+1) \dot{-} x\big)$$

by induction on x, using 8.6, 7.12, and 7.9.3. □

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 $Z \,\big|\, \mathbb{I}_m(n)$ 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

$$\operatorname{ext}\Big(x \circ {\upharpoonleft}_n\big(\mathbb{F}_m\big), y\Big) \circ {\downharpoonleft}^n\big(\mathbb{I}_{m+1}(n)\big).$$

It has the required properties by 8.6 and 8.3. □

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 $z = \big(z \,\big|\, \mathbb{I}_m(n), z_{n+m+1}\big)$.

Proof By 8.3, it is sufficient to show that

$$z \circ {\upharpoonleft}_n(\mathbb{F}_{m+1}) \;=\; \operatorname{ext}\Big(\big(z \,\big|\, \mathbb{I}_m(n) \circ {\upharpoonleft}_n(\mathbb{F}_m)\big),\, z_{n+m+1}\Big).$$

Hence the result by 8.7 and 8.8. □

DEFINITION Let m be a positive number. An object of type $[1 :: m]$ is called an *m-index*. An operation defined on $[1 :: m]$ is called an *m-tuple*. If X is a type, then the type $[1 :: m] \longrightarrow X$ 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. □

NOTATION Let x be an object of type X. The constant operation $\varkappa_{\{1\}}(x)$ 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 $\forall_{(x:X)}\, P(x) \iff \forall_{(y:X^1)}\, P(y_1)$

2 $\exists_{(x:X)}\, P(x) \iff \exists_{(y:X^1)}\, P(y_1)$.

Proof By 5.3 and 5.8. □

8.12 / LEMMA

Let n be a natural number, X a type, and P a predicate defined on X^{n+1}. Then:

1 $\forall_{(x:X^n)}\big(\forall_{(y:X)}\, P(x,y)\big) \iff \forall_{(z:X^{n+1})}\, P(z)$

2 $\exists_{(x:X^n)}\big(\exists_{(y:X)}\, P(x,y)\big) \iff \exists_{(z:X^{n+1})}\, P(z)$.

Proof By 8.8, 5.3, and 5.8. □

ORDERED PAIRS

DEFINITION A one-tuple (resp. two-tuple, three-tuple) is called an *ordered singleton* (resp. *ordered pair*, *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

$$Z_1 \times Z_2 := \prod_{i=1}^{2} Z_i \quad \text{and} \quad Z_1 + Z_2 := \coprod_{i=1}^{2} Z_i.$$

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

$$F_0 = 0, \quad F_1 = 1, \quad \textit{and} \quad F_{x+2} = F_x + F_{x+1}$$

for all natural numbers x.

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

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

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

$$F_x = G_{x,1}, \qquad F_{x+1} = G_{x,2}$$
$$F_{x+2} = G_{x+1,2} = G_{x,1} + G_{x,2} = F_x + F_{x+1}.$$ □

9

BINARY QUANTIFICATION

CONJUNCTION AND DISJUNCTION

DEFINITION Let P_1 and P_2 be propositions. The proposition

$$\forall_{[1::2]}(P_1, P_2)$$

is denoted by $P_1 \wedge P_2$ 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* $P_1 \wedge P_2$ *is*

$$\Box(P_1) \times \Box(P_2). \qquad \Box$$

Remark A proof of $P_1 \wedge P_2$ 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 $P_1 \wedge P_2$ *is true if and only if* P_1 *and* P_2 *are both true.*

Proof By 8.13 and 8.14. □

9.3 / THEOREM

Let P, Q_1, *and* Q_2 *be propositions. Then* $P \implies (Q_1 \wedge Q_2)$ *is equivalent to*

$$(P \implies Q_1) \wedge (P \implies Q_2).$$

Proof By 6.18.1. □

9.4 / EXPORTATION

Let P_1, P_2, *and* Q *be propositions. Then*

$$P_1 \implies (P_2 \implies Q) \quad \textit{is equivalent to} \quad (P_1 \wedge P_2) \implies Q$$

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

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

$$\big(P \implies Q\big) \wedge \big(Q \implies P\big)$$

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 $P_1 \vee P_2$ and called the *disjunction of P_1 and P_2*.

Notation The proposition $P_1 \vee P_2$ 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 $P_1 \vee P_2$ is

$$\Box(P_1) + \Box(P_2).$$ □

Remark A proof of $P_1 \vee P_2$ 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 $P_1 \vee P_2$ is true.*

2 *If P_2 is true, then $P_1 \vee P_2$ is true.*

Proof By 8.15. □

9.8 / CASE ANALYSIS

Let P_1, P_2, and Q be propositions. Then $(P_1 \vee P_2) \implies Q$ is equivalent to

$$(P_1 \implies Q) \wedge (P_2 \implies Q)$$

Proof By 6.18.2. □

9.9 / LEMMA

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

1. $P \wedge R \implies Q \wedge R$ *and* $R \wedge P \implies R \wedge Q$
2. $P \vee R \implies Q \vee R$ *and* $R \vee P \implies R \vee Q$.

Proof By 6.4 and 6.14. □

9.10 / COROLLARY

Let P, Q, and R be propositions. If P is equivalent to Q, then

1. $P \wedge R \iff Q \wedge R$ *and* $R \wedge P \iff R \wedge Q$
2. $P \vee R \iff Q \vee R$ *and* $R \vee P \iff R \vee Q$.

Proof By 9.5, 9.2, and 9.9. □

THE LOGIC OF BINARY QUANTIFICATION

9.11 / IDEMPOTENT LAWS

Let P be a proposition.

1. $P \wedge P \iff P$
2. $P \vee P \iff P$.

Proof By 6.10. □

9.12 / COMMUTATIVE LAWS

Let P and Q be propositions.

1. $P \wedge Q \iff Q \wedge P$
2. $P \vee Q \iff Q \vee P$.

Proof By 9.2, 9.7, and 9.8. □

Remark The next two theorems are proved in the same way.

9.13 / ASSOCIATIVE LAWS

Let P, Q, and R be propositions.

1. $P \wedge (Q \wedge R) \iff (P \wedge Q) \wedge R$
2. $P \vee (Q \vee R) \iff (P \vee Q) \vee R$. □

9.14 / LEMMA

Let P and Q be propositions.

1 $P \wedge (P \vee Q) \iff P$

2 $P \vee (P \wedge Q) \iff P.$ □

9.15 / THEOREM

Let X be a type and P_1 and P_2 predicates on X.

1 $\forall_{(x:X)} \big(P_1(x) \wedge P_2(x)\big) \iff \big(\forall_{(x:X)} P_1(x)\big) \wedge \big(\forall_{(x:X)} P_2(x)\big)$

2 $\exists_{(x:X)} \big(P_1(x) \vee P_2(x)\big) \iff \big(\exists_{(x:X)} P_1(x)\big) \vee \big(\exists_{(x:X)} P_2(x)\big).$

Proof By 6.20 and 6.22, respectively. □

9.16 / COROLLARY

Assume that P and Q are predicates on a type X.

1 $\forall_{(x:X)} \big(P(x) \iff Q(x)\big)$ *implies* $\forall_{(x:X)} P(x) \iff \forall_{(x:X)} Q(x)$

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

Proof By 9.15.1, 9.9, and 6.14. □

9.17 / COROLLARY

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

1 $Q \wedge \big(\forall_{(x:X)} P(x)\big) \iff \forall_{(x:X)} \big(Q \wedge P(x)\big)$

2 $Q \vee \big(\exists_{(x:X)} P(x)\big) \iff \exists_{(x:X)} \big(Q \vee P(x)\big).$

Proof By 9.15 and 6.10. □

9.18 / DISTRIBUTIVE LAWS

Let P, Q, and R be propositions.

1 $P \wedge \big(Q \wedge R\big) \iff \big(P \wedge Q\big) \wedge \big(P \wedge R\big)$

2 $P \vee \big(Q \vee R\big) \iff \big(P \vee Q\big) \vee \big(P \vee R\big).$ □

9.19 / THEOREM

Let X be a type and P and Q predicates on X. Then

1 $\exists_{(x:X)} \big(P(x) \wedge Q(x)\big)$ *implies* $\big(\exists_{(x:X)} P(x)\big) \wedge \big(\exists_{(x:X)} Q(x)\big).$

2 $\big(\forall_{(x:X)} P(x)\big) \vee \big(\forall_{(x:X)} Q(x)\big)$ *implies* $\forall_{(x:X)} \big(P(x) \vee Q(x)\big).$

Proof By 6.23. □

9.20 / COROLLARY

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

1 $Q \wedge \bigl(\exists_{(x:X)} P(x)\bigr)$ *is equivalent to* $\exists_{(x:X)} \bigl(Q \wedge P(x)\bigr)$

2 $Q \vee \bigl(\forall_{(x:X)} P(x)\bigr)$ *implies* $\forall_{(x:X)} \bigl(Q \vee P(x)\bigr)$.

Proof of 1 Declare p as a proof of the left-hand proposition. Then $\bigl(p_1, \sigma_2(p_2)\bigr)$ is a proof of $Q \wedge P\bigl(\sigma_1(p_2)\bigr)$. Hence the result by 5.8 and 6.23.

Proof of 2 By 6.23, using 6.7, 9.9, and 6.5. □

9.21 / DISTRIBUTIVE LAWS

Let P, Q, and R be propositions. Then

1 $P \wedge (Q \vee R) \iff (P \wedge Q) \vee (P \wedge R)$

2 $P \vee (Q \wedge R) \iff (P \vee Q) \wedge (P \vee R)$.

Proof The first equivalence follows from 9.20.1. The proposition

$$\bigl((P \vee Q) \wedge P\bigr) \vee \bigl((P \vee Q) \wedge R\bigr)$$

is equivalent to the right-hand side of 2 by 1 and 9.10. It is equivalent to

$$\bigl((P \wedge P) \vee (Q \wedge P)\bigr) \vee \bigl((P \wedge R) \vee (Q \wedge R)\bigr)$$

in the same way, using 9.12. Hence the result by 9.11, 9.14, and 9.13. □

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* $\forall_{(x:X)} \bigl((x : A) \implies P(x)\bigr)$

2 $\exists_{(x:A)} P(x)$ *is equivalent to* $\exists_{(x:X)} \bigl((x : A) \wedge P(x)\bigr)$. □

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. □

10

CONSTRUCTING SUBTYPES

DEFINING A SUBTYPE BY A PREDICATE

DEFINITION Let X be a type and P a predicate on X. The type

$$\{x : X \mid P(x)\} := \sigma_1\big(\Box\big(\exists_{(x:X)} P(x)\big)\big)$$

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* $\forall_{(x:X)} \big(P(x) \implies Q(x)\big)$

2 $X \mid P = X \mid Q$ *if and only if* $\forall_{(x:X)} \big(P(x) \iff Q(x)\big)$.

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* $\forall_{(x:X)} \big(x : A \implies x : B\big)$

2 $A = B$ *if and only if* $\forall_{(x:X)} \big(x : A \iff x : B\big)$.

Proof By 10.2 and 10.3. □

10.5 / COROLLARY

If n is a natural number, then $\mathbb{F}_n = \{x : \mathbb{N} \mid x \le n\}$.

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

$$\coprod\nolimits_{(A:\mathbb{U}_m)} \Box(A \subseteq X) : \mathbb{U}_{m+1}$$

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 $A : K \longrightarrow \mathcal{P}(X)$ 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

$$\bigcap_{\kappa : K} A_\kappa := \textstyle\bigcap_{(\kappa : K)} A_\kappa := \{ x : X \mid \forall_{(\kappa : K)} (x : A_\kappa) \}$$

$$\bigcup_{\kappa : K} A_\kappa := \textstyle\bigcup_{(\kappa : K)} A_\kappa := \{ x : X \mid \exists_{(\kappa : K)} (x : A_\kappa) \}$$

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 κ of type K if and only if

$$B \subseteq \bigcap_{\kappa : K} A_\kappa. \qquad \square$$

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 κ of type K if and only if

$$\bigcup_{\kappa : K} A_\kappa \subseteq B. \qquad \square$$

10.10 / LEMMA

Let A be a family of subtypes of X indexed by K. If λ has type K, then

$$\bigcap_{\kappa : K} A_\kappa \subseteq A_\lambda \subseteq \bigcup_{\kappa : K} A_\kappa \quad \textit{and} \quad \bigcap_{\kappa : \{\lambda\}} A_\kappa = A_\lambda = \bigcup_{\kappa : \{\lambda\}} A_\kappa.$$

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 κ of type K, then

$$\bigcap_{\kappa : K} A_\kappa \subseteq \bigcap_{\kappa : K} B_\kappa \quad \textit{and} \quad \bigcup_{\kappa : K} A_\kappa \subseteq \bigcup_{\kappa : K} B_\kappa.$$

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

$$\bigcap_{\kappa : K} A_\kappa \subseteq \bigcap_{\kappa : J} A_\kappa \quad \textit{and} \quad \bigcup_{\kappa : J} A_\kappa \subseteq \bigcup_{\kappa : K} A_\kappa.$$

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 λ 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 κ as an object of type K. Choose λ of type L such that κ has type J_λ. Then by 10.10,

$$x : \bigcap_{\mu : J_\lambda} A_\mu \subseteq A_\kappa.$$

Proof of 2 Declare λ 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 κ of type K and λ of type L such that x has type A_κ and κ has type J_λ. 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

$$f[A] = \bigcup_{a:A} f[a] \quad \textit{and} \quad X = \bigcup_{x:X} \{x\}.$$

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

$$\bigcap_{\lambda : f[K]} A_\lambda = \bigcap_{\kappa : K} A_{f(\kappa)} \quad \textit{and} \quad \bigcup_{\lambda : f[K]} A_\lambda = \bigcup_{\kappa : K} A_{f(\kappa)}.$$

Proof According to 10.14, the operation

$$\kappa : K \longmapsto f[\kappa]$$

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

1 $$f\left[\bigcap_{\kappa : K} A_\kappa\right] \subseteq \bigcap_{\kappa : K} f[A_\kappa]$$

2 $$f\left[\bigcup_{\kappa : K} A_\kappa\right] = \bigcup_{\kappa : K} f[A_\kappa].$$

Proof of 1 Let B denote the intersection of A. If κ has type K, then

$$f[B] \subseteq f[A_\kappa]$$

by 10.10 and 3.14. Hence the result by 10.8.

Proof of 2 By 10.13 and 10.14. □

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

$$\bigcap_{i=m}^{m+n} A_i := \bigcap_{i:\mathbb{I}_n(m)} A_i \quad \text{and} \quad \bigcup_{i=m}^{m+n} A_i := \bigcup_{i:\mathbb{I}_n(m)} A_i.$$

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 $\mathbb{F}_m = \mathbb{F}_{m \dot{-} 1} \cup \{m\}$.

Proof By 7.21, 10.9, and 7.19, it is sufficient to prove that $\mathbb{F}_m \subseteq \mathbb{F}_{m \dot{-} 1} \cup \{m\}$. Suppose that x has type $\mathbb{F}_m$. Choose a natural number y such that

$$x = m \dot{-} y.$$

If $y = 0$, then $x = m$. If $y \geq 1$, then $y = z + 1$ for some natural number z, so

$$x = m \dot{-} (z + 1) = (m \dot{-} 1) \dot{-} z \leq m \dot{-} 1.$$

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 $$f^{-1}\left[\bigcap_{\kappa : K} B_\kappa\right] = \bigcap_{\kappa : K} f^{-1}[B_\kappa]$$

2 $$f^{-1}\left[\bigcup_{\kappa : K} B_\kappa\right] = \bigcup_{\kappa : K} f^{-1}[B_\kappa]. \qquad \square$$

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

$$f\left[f^{-1}[B]\right] \subseteq B.$$

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

$$A \subseteq f^{-1}\left[f[A]\right].$$

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. □

11

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 $\bot \vee Q$, 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

$$x \neq y \quad \text{denotes} \quad \neg (x = y).$$

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 $P \wedge \neg P$ is called a *contradiction.*

11.3 / THEOREM

Let P be a proposition. Then $P \wedge \neg P$ is false.

Proof If x proves $P \wedge \neg P$, 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 $\exists_{(x:\mathbb{V})}(x \wedge \neg x)$ 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

$$i : \mathbb{F}_1 \longmapsto \big\{ j : \mathbb{F}_1 \,\big|\, P \vee (i = j) \big\}.$$

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 $(X = A_0) \vee (X = A_1)$ 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

$$P \vee \Big(\big(f(A_0) = 0\big) \wedge \big(f(A_1) = 1\big) \Big) \tag{11.6}$$

is true by 9.21.2. It follows from 9.7 and 10.3 that P implies $A_0 = \mathbb{F}_1 = A_1$. 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 $\forall_{(x:\mathbb{V})}(x \vee \neg x)$ 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 $P \vee Q$ 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 $P \vee Q$ 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 $\neg (P \wedge Q) \iff \neg P \vee \neg Q$

2 $\neg (P \vee Q) \iff \neg P \wedge \neg Q$. □

11.14 / LEMMA

Let P and Q be propositions. Then $P \implies Q$ is equivalent to $Q \vee \neg P$.

Proof The proposition $Q \vee \neg P$ implies $P \implies Q$ by 9.8, using 6.8 and 11.1. The converse follows from 11.5 and 9.9.2. □

11.15 / COROLLARY

Let P and Q be propositions. Then $\neg(P \implies Q)$ is equivalent to $P \wedge \neg Q$.

Proof By 11.14, 11.13, and 9.12.1. □

11.16 / COROLLARY

Let X be a type and P and Q predicates on X. Then $\exists_{(x:X)} \big(P(x) \implies Q(x)\big)$ 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. □

11.17 / COROLLARY

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

1 $\exists_{(x:X)} \big(P(x) \implies Q\big)$ *is equivalent to* $\forall_{(x:X)} P(x) \implies Q$.

2 $\exists_{(x:X)} \big(Q \implies P(x)\big)$ *is equivalent to* $Q \implies \exists_{(x:X)} P(x)$.

Proof By 11.16 and 6.10. □

11.18 / LEMMA

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

$$Q \vee \big(\forall_{(x:X)} P(x)\big) \text{ is equivalent to } \forall_{(x:X)} \big(Q \vee P(x)\big).$$

Proof By 6.19.1, 11.11, 11.12, and 11.13, it is sufficient to prove that

$$\neg Q \wedge \exists_{(x:X)} \neg P(x) \text{ implies } \exists_{(x:X)} \big(\neg Q \wedge \neg P(x)\big).$$

Hence the result by 9.20.1. □

VOID TYPES AND COMPLEMENTS

DEFINITION Let X be a type. The type $\varnothing_X$ 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 $\varnothing_X$ if and only if $0 = 1$.

11.19 / THEOREM

If x is an object of type X, then the proposition $x : \varnothing_X$ is false. □

DEFINITION Let X be a type. If $X = \varnothing_Y$ for some type Y, then X is said to be *void*. Therefore $\varnothing_X$ is void.

11.20 / COROLLARY

Let X be a type and P a predicate on X. The proposition $\forall_{(x : \varnothing_X)} P(x)$ is true, and $\exists_{(x : \varnothing_X)} P(x)$ 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 $\varnothing_X \subseteq A$.

Proof By 11.20 and 5.5. □

11.22 / COROLLARY

Let X be a type. If $A \subseteq \varnothing_X$, then $A = \varnothing_X$. □

Proof By 11.21 and 0.24. □

11.23 / COROLLARY

If f is an operation from X to Y, then

$$f[\varnothing_X] = \varnothing_Y \quad \textit{and} \quad f^{-1}[\varnothing_Y] = \varnothing_X.$$

Proof Let x be an object of type X. Then $f(x)$ has type $\varnothing_Y$ if and only if x has type $\varnothing_X$. 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

$$\bigcap\nolimits_{(\kappa : \varnothing_K)} A_\kappa = X \quad \textit{and} \quad \bigcup\nolimits_{(\kappa : \varnothing_K)} A_\kappa = \varnothing_X.$$

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 $X - X = \varnothing_X$ and $X - \varnothing_X = X$.

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. □

12

EQUATING MATHEMATICAL OBJECTS

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 $R(x, y) \wedge R(y, z)$ 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:

LOGICAL RELATION	NAME
$x,\, y : \mathbb{V} \longmapsto x \implies y$	*logical implication*
$x,\, y : \mathbb{V} \longmapsto x \iff y$	*logical equivalence*
$x,\, y : \mathbb{V} \longmapsto x \wedge y$	*logical conjunction*
$x,\, y : \mathbb{V} \longmapsto x \vee y$	*logical disjunction*

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.

INTUITION Equivalence relations are used to *equate* mathematical objects.

DEFINITION Let R be a relation on X and A a subtype of X. The operation

$$x, y : A \longmapsto R(x, y)$$

is denoted by $R \,\big|\, A$ 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 $R \,\big|\, A$ is an equivalence relation.

Proof By 9.23.1, using 8.10 through 8.12. For example, to prove that $R \,\big|\, A$ is symmetric, declare Y as a subtype of X. Then $R \,\big|\, Y$ is symmetric if and only if

$$\forall_{(x:Y^2)} \big(R(x_1, x_2) \implies R(x_2, x_1)\big),$$

by 8.11 and 8.12. Thus $R \,\big|\, A$ is symmetric by 8.10 and 9.23. □

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

$$f, g : \prod_{x:X} F_x \longmapsto \forall_{(x:X)} R_x\big(f(x), g(x)\big)$$

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 $R_x\big(f(x), f(x)\big)$ is true, since R_x is reflexive.

Proof of symmetry Declare f and g as operations from X to Y. By 6.14.1,

$$\forall_{(x:X)} R_x\big(f(x), g(x)\big) \implies \forall_{(x:X)} R_x\big(g(x), f(x)\big)$$

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 $S(f, g) \wedge S(g, h)$ is equivalent to

$$\forall_{(x:X)} \Big(R_x\big(f(x), g(x)\big) \wedge R_x\big(g(x), h(x)\big)\Big)$$

by 9.15. This proposition implies $S(f, h)$ by 6.141, since R_x is transitive. □

NOTATION Let m be a number and X a type of order m. Define $\mathcal{E}_m(X)$ as

$$\Big\{ R : X \longrightarrow \mathbb{V}_m \;\Big|\; R \text{ is an equivalence relation} \Big\}.$$

MATHEMATICAL EQUATIONS

DEFINITION Let m be a number. A *set of order m* is an object of type

$$\mathrm{SET}_m := \coprod_{(X : \mathbb{U}_m)} \mathcal{E}_m(X).$$

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

$$\big(\sigma_2(A)\big)(a, b) \quad \text{is denoted by} \quad a =_A b$$

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 $\mathbf{N}$ 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 $\mathbf{N}$.

EXAMPLE The following are examples of higher-order sets.

1. The identity set of $\mathbb{U}$ is denoted by $\mathbf{U}$ and called the *set of types*.
2. Let E denote logical equivalence. The assembly $[\,\mathbb{V}, E\,]$ is denoted by $\mathbf{V}$ 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

$$\vartheta_A(X) := \bigl[X, \sigma_2(A) \bigm| X\bigr]$$

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 $f : \sigma_1(A) \longrightarrow \sigma_1(B)$. If

$$a_1 = a_2 \quad \text{implies} \quad f(a_1) = f(a_2)$$

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 $\varkappa_A(b)$ is defined as the constant operation $\varkappa_{\sigma_1(A)}(b)$, which is a function from A to B. □

Definition The function $\varkappa_A(b)$ 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 $x \in X \longmapsto f_x$.

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

$$\prod_{x \in A} \varkappa_A(B, x).$$

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 $A \longrightarrow B$.

Notation If f and g are functions from A to B, then the symbol $f = g$ means that f equals g in the set $A \longrightarrow B$. 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. □

NOTATION Let A and B be sets. The proposition that f belongs to the set $A \longrightarrow B$ is denoted by the symbol $f : A \longrightarrow B$.

DEFINITION Let X be a set and A a family of sets indexed by X. If x is an element of X, then the evaluation operator

$$\mathrm{ev}_x : \left(\prod\nolimits_X (\sigma_1 \circ A)\right) \longrightarrow \sigma_1(A_x)$$

constructed in 1.10 is denoted by π_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,

$$\pi_x : \prod\nolimits_X (A) \longrightarrow A_x.$$

DEFINITION Let A be a set and P a predicate on $\sigma_1(A)$. The proposition

$$\exists!_{(x \in A)}\, P(x) := \exists_{(x \in A)} \Big(P(x) \wedge \forall_{(y \in A)} \big(P(y) \implies (x =_A y)\big)\Big)$$

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

$$\varphi : X \longrightarrow \prod\nolimits_Y (A)$$

such that $\pi_y \circ \varphi = f_y$ for every element y of Y.

Proof of uniqueness If the function

$$\tilde{\varphi} : X \longrightarrow \prod\nolimits_Y (A)$$

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 φ as the operation

$$x \in X,\, y \in Y \;\longmapsto\; f_y(x)$$

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 $\varphi_{x_1} = \varphi_{x_2}$. □

SUBSETS AND PIECEWISE DEFINITION

DEFINITION Let f be a function from A to B. If the proposition

$$f(a_1) = f(a_2) \quad \text{implies} \quad a_1 = a_2$$

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

$$a_1 =_A a_2 \quad \text{is equivalent to} \quad a_1 =_B a_2$$

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 $\vartheta_A(X)$ 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 $\mathbf{V}$ of propositions is called a *propositional function on* X. If A is a proper subset of X and the predicate

$$x \in X \longmapsto x \in A$$

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_κ is saturated in X for all κ 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 κ of K, let f_κ be a function from A_κ to Y. Suppose that

$$f_\kappa(x) = f_\lambda(x)$$

for all κ and λ 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 κ of K and every element x of A_κ.

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

$$g(x) = f\big(\varphi(x), x\big).$$

Proof of existence Since A is a covering of X, it follows that $\varphi : \sigma_1(X) \longrightarrow Y$, and therefore $g : \sigma_1(X) \longrightarrow \sigma_1(Y)$. 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

$$g(x_1) = f\big(\varphi(x_2), x_1\big) = g(x_2). \qquad \square$$

Definition The function g is said to be *piecewise defined on A by f*.

DEFINITION Let X be a set and P a propositional function on X. The set

$$X \mid P := \vartheta_X \big(\sigma_1(X) \mid P\big)$$

is called the *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

$$P_1(x) \vee P_2(x)$$

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

$$P_1(x) \wedge P_2(x) \quad \text{implies} \quad f_1(x) = f_2(x)$$

for every element x of X. Then the function g piecewise defined on X by f is denoted by the symbol

$$x \in X \longmapsto \begin{cases} f_1(x) & \textit{if } P_1(x), \\ f_2(x) & \textit{if } P_2(x). \end{cases}$$

12.17 / COROLLARY

Let K, X, and Y be sets and A a saturated covering of X indexed by K, where

$$\sigma_1(A_\kappa) \cap \sigma_1(A_\lambda)$$

is void for all κ and λ in K. For every element κ of K, let $f_\kappa : A_\kappa \longrightarrow Y$. There is a unique function $g : X \longrightarrow Y$ such that

$$g(x) = f_\kappa(x)$$

for every element κ of K and every element x of A_κ.

Proof By 12.16 and 11.20. □

PART III

CONCLUDING REMARKS

E

MATHEMATICAL STRUCTURES

REMARK The definitions and postulates given in this appendix are useful for the formalization of more advanced mathematics.

NOTATION In this chapter, let m denote a number. The symbol W denotes the constructor with index fourteen.

DEFINITION The term $\mathsf{W}(m)$ is denoted by $\mathbb{W}_m$. It is called the *structural universe of order m*.

NOTATION The structural universe $\mathbb{W}_\nu$ is denoted by $\mathbb{W}$.

0.37 / CONSTRUCTION OF STRUCTURAL UNIVERSES

It is postulated that $\mathbb{W}_m$ is a type of order $m+1$ and a subtype of $\mathbb{U}_{m+1}$.

Definition An object S of type $\mathbb{W}_m$ is called a *structural type of order m*. An object X of type S is called a *mathematical structure of order m*, or simply a *structure of order m*.

NOTATION The symbol B denotes the constructor with index fifteen. If X is a term, then $\mathsf{B}(X)$ is denoted by X_b.

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.

Definition The type X_b is called the *base type of X*.

0.39 / TYPES AS STRUCTURES

It is postulated that $\mathbb{U}_m$ is a structural type of order m, and that $X_b = X$ for all types X of order m.

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 Φ a family of types of order m indexed by S. It is postulated that

$$\coprod_S (\Phi) : \mathbb{W}_m \quad \textit{and that} \quad [X, \varphi]_b = X_b$$

for all X of type S and all φ of type $\Phi(X)$.

Intuition A structure $[X, \varphi]$ can be constructed from X by combining it with an object φ, 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. □

F

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 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

$$\exists_{(X : \mathbb{U})} (a : X)$$

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

1967 *Foundations of Constructive Analysis*. Ishi Press, 2012 edition.

1975 The crisis in contemporary mathematics. *Historia Mathematica*, vol. **2**,pp. 507-517.

BOURBAKI, NICOLAS

1970 *Theory of Sets*. Springer, 2004 edition.

MARTIN-LÖF, PER

1980 Intuitionistic type theory. Notes by Giovanni Sambin [on] a series of lectures given in Padua, June 1980.

1996 On the meanings of the logical constants and the justifications of the logical laws. *Nordic Journal of Philosophical Logic*, vol. **1**, pp. 11-60.

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

1975 Axiom of choice and complementation. *Proceedings of the American Mathematical Society*, vol. **51**, pp. 176-178.

GOODMAN, NOAH and MYHILL, JOHN

1978 Choice implies excluded middle. *Mathematical Logic Quarterly*, vol. **24**, p. 461.

Hofmann, Martin

1997 Syntax and semantics of dependent types. In *Semantics and Logics of Computation*, Cambridge University Press, pp. 79-130.

Kleene, Stephen C.

1952 *Introduction to Metamathematics.* North Holland, 1971 edition.

Riehl, Emily

2016 *Category Theory in Context.* Dover Publications.

OTHER INFLUENCES

de Bruijn, Nicolaas G.

1972 Lambda calculus notation with nameless dummies, a tool for automatic formula manipulation, with application to the Church-Rosser theorem. *Indagationes Mathematicae*, vol. **75**, pp. 381-392.

Everett, Caleb

2017 *Numbers and the Making of Us: Counting and the Course of Human Cultures.* Harvard University Press.