Elements of Mathematical Logic

Michael Meyling

February 14, 2014

The source for this document can be found here:

`http://www.qedeq.org/0_04_08/doc/math/qedeq_logic_v1.xml`

If you have any questions, suggestions or want to add something to the list of modules that use this one, please send an email to the address mime@qedeq.org

The authors of this document are: Michael Meyling michael@meyling.com

Contents

Summary

The project **Hilbert II** deals with the formal presentation and documentation of mathematical knowledge. For this reason **Hilbert II** provides a program suite to accomplish that tasks. The concrete documentation of mathematical basics is also a purpose of this project. For further informations about the **Hilbert II** project see under `http://www.qedeq.org`.

This document describes the logical axioms and the rules and meta rules that are used to derive new propositions.

The presentation is axiomatic and in a formal form. A formal calculus is given that enables us to derive all true formulas. Additional derived rules, theorems, definitions, abbreviations and syntax extensions basically correspond with the mathematical practice.

This document is also written in a formal language, the original text is a XML file with a syntax defined by the XSD `http://www.qedeq.org/current/xml/qedeq.xsd`.

This document is work in progress and is updated from time to time. Especially at the locations marked by „+++“ additions or changes will take place.

Foreword

The whole mathematical universe can be unfolded by set–theoretic means. Beside the set–theoretic axioms only logical axioms and rules are required. These elementary basics are sufficient to define the most complex mathematical structures and enable us to prove propositions for those structures. This approach can be fully formalized and can be reduced to simple manipulations of character strings. The semantical interpretation of these character strings represent the mathematical universum.

It is more than convenient to introduce abbreviations and use further derivation rules. But these comforts could be eliminated and replaced by the basic terms at any time[1].

This project has its source in a childhood dream to undertake a formalization of mathematics. In the meantime the technical possibilities are highly developed so that a realization seems within reach.

Special thanks go to the professors *W. Kerby* and *V. Günther* of the university of Hamburg for their inspiring lectures about logic and axiomatic set theory. Without these important impulses this project would not exist.

I am deeply grateful to my wife *Gesine Dräger* and our son *Lennart* for their support and patience.

Hamburg, January, 2013

Michael Meyling

---

[1] At least this is theoretically possible. This transformation is not in each case practically realizable due to restrictions in time and space. For example it is not possible to write down the natural number $1,000,000,000$ completely in set notation.

Introduction

At the beginning we quote *D. Hilbert* from the lecture „The Logical Basis of Mathematics“, September 1922[2].

> „The fundamental idea of my proof theory is the following:
>
> All the propositions that constitute in mathematics are converted into formulas, so that mathematics proper becomes all inventory of formulas. These differ from the ordinary formulas of mathematics only in that, besides the ordinary signs, the logical signs especially „implies“ ($\rightarrow$) and for „not“ ($\bar{\ }$) occur in them. Certain formulas, which serve as building blocks for the formal edifice of mathematics, are called axioms. A proof is an array that must be given as such to our perceptual intuition of it of inferences according to the schema
>
> $$\begin{array}{c} A \\ A \rightarrow B \\ \hline B \end{array}$$
>
> where each of the premises, that is, the formulae, A and $A \rightarrow B$ in the array either is an axiom or directly from an axiom by substitution, or else coincides with the end formula B of an inference occurring earlier in the proof or results from it by substitution. A formula is said to be provable if it is either an axiom or the end formula of a proof.“

At the beginning there is logic. Logic is the analysis of methods of reasoning. It helps to derive new propositions from already given ones. Logic is universally applicable.

In the 1928 published book *Grundzüge der theoretischen Logik* (Principles of Theoretical Logic) *D. Hilbert* and *W. Ackermann* formalized propositional calculus in a way that build the basis for the logical system used here. 1959 *P. S. Novikov* specified a refined axiom and rule system for predicate calculus.

In this text we present a first order predicate calculus with identity and functors that is the starting point for the development of the mathematical theory. Only the results without any proofs and in short form are given in the following.[3]

---

[2] Lecture given at the Deutsche Naturforscher-Gesellschaft, September 1922.

[3] If there is time proofs will be added.

Chapter 1

Language

In this chapter we define a formal language to express mathematical propositions in a very precise way. Although this document describes a very formal approach to express mathematical content it is not sufficent to serve as a definition for an computer readable document format. Therefore such an extensive specification has to be done elsewhere. The choosen format is the *Extensible Markup Language* abbreviated *XML*. XML is a set of rules for encoding documents electronically.[1] The according formal syntax specification can be found at `http://www.qedeq.org/current/xml/qedeq.xsd`. It specifies a complete mathematical document format that enables the generation of LaTeXbooks and makes automatic proof checking possible. Further syntax restrictions and some explanations can be found at `http://www.qedeq.org/current/doc/project/qedeq_logic_language_en.pdf`.

Even this document is (or was generated) from an XML file that can be found here: `http://wwww.qedeq.org/0_04_08/doc/math/qedeq_logic_v1.xml`. But now we just follow the traditional mathematical way to present the elements of mathematical logic.

1.1 Terms and Formulas

We use the *logical symbols* $L = \{$ ‘$\neg$’, ‘$\vee$’, ‘$\wedge$’, ‘$\leftrightarrow$’, ‘$\rightarrow$’, ‘$\forall$’, ‘$\exists$’ $\}$, the *predicate constants* $C = \{c_i^k \mid i,k \in \omega\}$, the *function variables*[2] $F = \{f_i^k \mid i,k \in \omega \wedge k > 0\}$, the *function constants*[3] $H = \{h_i^k \mid i,k \in \omega\}$, the *subject variables* $V = \{v_i \mid i \in \omega\}$, as well as *predicate variables* $P = \{p_i^k \mid i,k \in \omega\}$.[4] For the *arity* or *rank* of an operator we take the upper index. The set of predicate variables with zero arity is also called set of *proposition variables* or *sentence letters*: $A := \{p_i^0 \mid i \in \omega\}$. For subject variables we write short hand certain lower letters: $v_1 =$ ‘u’, $v_2 =$ ‘v’, $v_3 =$ ‘w’, $v_4 =$ ‘x’, $v_5 =$ ‘y’, $v_5 =$ ‘z’. Furthermore we use the following short notations: for the predicate variables $p_1^n =$ ‘ϕ’ und $p_2^n =$ ‘ψ’, where the appropriate arity n is calculated by counting the subsequent parameters, for the proposition variables $a_1 =$ ‘A’, $a_2 =$ ‘B’ and $a_3 =$ ‘C’,

---

[1] See `http://www.w3.org/XML/` for more information.

[2] Function variables are used for a shorter notation. For example writing an identity proposition $x = y \rightarrow f(x) = f(y)$. Also this introduction prepares for the syntax extension for functional classes.

[3] Function constants are also introduced for convenience and are used for direct defined class functions. For example to define building of the power class operator, the union and intersection operator and the successor function. All these function constants can be interpreted as abbreviations.

[4] By ω we understand the natural numbers including zero. All involved symbols are pairwise disjoint. Therefore we can conclude for example: $f_i^k = f_{i'}^{k'} \rightarrow (k = k' \wedge i = i')$ and $h_i^k \neq v_j$.

for the function variables: $f_1^n =$ 'f' und $f_2^n =$ 'g', where again the appropriate arity n is calculated by counting the subsequent parameters. All binary propositional operators are written in infix notation. Parentheses surrounding groups of operands and operators are necessary to indicate the intended order in which operations are to be performed. E. g. for the operator $\land$ with the parameters A and B we write $(A \land B)$.

In the absence of parentheses the usual precedence rules determine the order of operations. Especially outermost parentheses are omitted. Also empty parentheses are stripped.

The operators have the order of precedence described below (starting with the highest).

$$\begin{array}{c} \neg, \forall, \exists \\ \land \\ \lor \\ \rightarrow, \leftrightarrow \end{array}$$

The term *term* is defined recursively as follows:

1. Every subject variable is a term.
2. Let $i, k \in \omega$ and let t_1, …, t_k be terms. Then $h_i^k(t_1, \ldots, t_k)$ is a term and if $k > 0$, so $f_i^k(t_1, \ldots, t_k)$ is a term too.

Therefore all zero arity function constants $\{h_i^0 \mid i \in \omega\}$ are terms. They are called *individual constants*.[5]

We define a *formula* and the relations *free* and *bound* subject variable recursivly as follows:

1. Every proposition variable is a formula. Such formulas contain no free or bound subject variables.
2. If p^k is a predicate variable with arity k and c^k is a predicate constant with arity k and $t_1, t_2, \ldots, t_k$ are terms, then $p^k(t_1, t_2, \ldots t_k)$ and $c^k(t_1, t_2, \ldots, t_k)$ are formulas. All subject variables that occur at least in one of $t_1, t_2, \ldots, t_k$ are free subject variables. Bound subject variables does not occur.[6]
3. Let α, β be formulas in which no subject variables occur bound in one formula and free in the other. Then $\neg\alpha$, $(\alpha \land \beta)$, $(\alpha \lor \beta)$, $(\alpha \rightarrow \beta)$ and $(\alpha \leftrightarrow \beta)$ are also formulas. Subject variables which occur free (respectively bound) in α or β stay free (respectively bound).
4. If in the formula α the subject variable x_1 occurs not bound[7], then also $\forall x_1\ \alpha$ and $\exists x_1\ \alpha$ are formulas. The symbol $\forall$ is called *universal quantifier* and $\exists$ as *existential quantifier*.

   Except for x_1 all free subject variables of α stay free. All bound subject variables are still bound and additionally x_1 is bound too.

All formulas that are only built by usage of 1. and 3. are called formulas of the *propositional calculus*.

---

[5] In an analogous manner subject variables might be defined as function variables of zero arity. Because subject variables play an important role they have their own notation.

[6] This second item includes the first one, which is only listed for clarity.

[7] This means that x_1 is free in the formula or does not occur at all.

For each formula α the following proposition holds: the set of free subject variables is disjoint with the set of bound subject variables..[8]

If a formula has the form $\forall x_1\ \alpha$ respectively $\exists x_1\ \alpha$ then the formula α is called the *scope* of the quantifier $\forall$ respectively $\exists$.

All formulas that are used to build up a formula by 1. to 4. are called *part formulas*.

Despite the fact we have predicate and function variables we still have a first order language. If predicates could have other predicates or functions as arguments or if we permited predicate or function quantifiers we would have left the area of first order theories. See [5], p. 55.

---

[8] Other formalizations allow for example $\forall x_1\ \alpha$ also if x_1 occurs already bound within α. Also propositions like $\alpha(x_1) \ \wedge\ (\forall\ x_1\ \beta)$ are allowed. In this formalizations free and bound are defined for a single occurrence of a variable.

Chapter 2

Axioms and Rules of Inference

We now state the system of axioms for the predicate calculus and present the rules for obtaining new formulas from them.

2.1 Axioms

The language of our calculus bases on the formalizations of *D. Hilbert*, *W. Ackermann*[3], *P. Bernays* and *P. S. Novikov*[4]. New rules can be derived from the herein presented. Only these meta rules lead to a smooth flowing logical argumentation.

We want to present the axioms, definitions and rules in an syntactical manner but to motivate the choosen form we already give some semantical *interpretations*.

The logical operators of propositional calculus '$\neg$', '$\vee$', '$\wedge$', '$\rightarrow$' and '$\leftrightarrow$' combine arbitrary *propositions* to new propositions. A proposition is a statement that affirms or denies something and is either „true“ or „false“ (but not both).[1]

The binary operator '$\vee$' (logical disjunction) combines the two propositions α and β into the new proposition $\alpha \vee \beta$. It results in true if at least one of its operands is true.

The unary operator '$\neg$' (logical negation) changes the truth value of a proposition α. $\neg\alpha$ has a value of true when its operand is false and a value of false when its operand is true.

The *logical implication* (*if . . . then*) the, *logical conjunction* (*and*) and the *logical equivalence* (*logical biconditional*) are defined as abbreviations.[2]

The logical implication ('if . . . then') could be defined as follows.

$$\alpha \rightarrow \beta \;:\leftrightarrow\; \neg\alpha \;\vee\; \beta$$

The logical conjunction ('and') could be defined with de Morgan.

$$\alpha \wedge \beta \;:\leftrightarrow\; \neg(\neg\alpha \;\vee\; \neg\beta)$$

---

[1] Later on we will define the symbols $\top$ and $\bot$ as truth values.

[2] Actually the symbols $\wedge, \rightarrow, \leftrightarrow$ are defined later on and are a syntax extension. But for convenience these symbols are already part of the logical language.

The logical equivalence (‘iff’) is defined as usual.

$$\alpha \leftrightarrow \beta \ :\leftrightarrow \ (\alpha \ \rightarrow \ \beta) \ \land \ (\beta \ \rightarrow \ \alpha)$$

Now we come to the first axiom of propositional calculus. This axiom enables us to get rid of an unnecessary disjunction.

Axiom 1 (Disjunction Idempotence). `[axiom:disjunction_idempotence]`

$$(A \ \lor \ A) \ \rightarrow \ A$$

If a proposition is true, any alternative may be added without making it false.

Axiom 2 (Axiom of Weakening). `[axiom:disjunction_weakening]`

$$A \ \rightarrow \ (A \ \lor \ B)$$

The disjunction should be commutative.

Axiom 3 (Commutativity of the Disjunction). `[axiom:disjunction_commutative]`

$$(A \ \lor \ B) \ \rightarrow \ (B \ \lor \ A)$$

An disjunction could be added at both side of an implication.

Axiom 4 (Disjunctive Addition). `[axiom:disjunction_addition]`

$$(A \ \rightarrow \ B) \ \rightarrow \ ((C \ \lor \ A) \ \rightarrow \ (C \ \lor \ B))$$

If something is true for all x, it is true for any specific y.

Axiom 5 (Universal Instantiation). `[axiom:universalInstantiation]`

$$\forall x \ \phi(x) \ \rightarrow \ \phi(y)$$

If a predicate holds for some particular y, then there is an x for which the predicate holds.

Axiom 6 (Existential Generalization). `[axiom:existencialGeneralization]`

$$\phi(y) \ \rightarrow \ \exists x \ \phi(x)$$

2.2 Rules of Inference

The following rules of inference enable us to obtain new true formulas from the axioms that are assumed to be true. From these new formulas we derive further formulas. So we can successively extend the set of true formulas.

Rule 1 (Modus Ponens). `[rule:modusPonens]`

Name: `MP` - Version: `0.01.00`

If both formulas α and $\alpha \rightarrow \beta$ are true, then we can conclude that β is true as well.

Rule 2 (Replace Free Subject Variable). `[rule:replaceFree]`

Name: `replaceFree` - Version: `0.01.00`

We start with a true formula. A free subject variable may be replaced by an arbitrary term, provided that the substituted term contains no subject variable that have a bound occurrence in the original formula. All occurrences of the free variable must be simultaneously replaced.

The prohibition to use subject variables within the term that occur bound in the original formula has two reasons. First it ensures that the resulting formula is well-formed. Secondly it preserves the validity of the formula. Let us look at the following derivation.

$$\begin{array}{rcll}
\forall x\ \exists y\ \phi(x,y) & \rightarrow & \exists y\ \phi(z,y) & \text{with axiom 5} \\
\forall x\ \exists y\ \phi(x,y) & \rightarrow & \exists y\ \phi(y,y) & \text{forbidden replacement: } z \text{ in } y\text{, despite } y \text{ is} \\
 & & & \text{already bound} \\
\forall x\ \exists y\ x \neq y & \rightarrow & \exists y\ \neq y & \text{replace } \neq \text{ for } \phi
\end{array}$$

This last proposition is not valid in many models.

Rule 3 (Rename Bound Subject Variable). `[rule:renameBound]`

Name: `renameBound` - Version: `0.01.00`

We may replace a bound subject variable occurring in a formula by any other subject variable, provided that the new variable occurs not free in the original formula. If the variable to be replaced occurs in more than one scope, then the replacement needs to be made in one scope only.

Rule 4 (Replace Predicate Variable). `[rule:replacePred]`

Name: `replacePred` - Version: `0.01.00`

Let α be a true formula that contains a predicate variable p of arity n, let x_1, …, x_n be pairwise different subject variables and let $\beta(x_1,\ldots,x_n)$ be a formula where x_1, …, x_n are not bound. The formula $\beta(x_1,\ldots,x_n)$ must not contain all x_1, …, x_n as free subject variables. Furthermore it can also have other subject variables either free or bound.

If the following conditions are fulfilled, then a replacement of all occurrences of $p(t_1,\ldots,t_n)$ each with appropriate terms t_1, …, t_n in α by $\beta(t_1,\ldots,t_n)$ results in another true formula.

- *the free variables of $\beta(x_1,\ldots,x_n)$ without x_1, …, x_n do not occur as bound variables in α*
- *each occurrence of $p(t_1,\ldots,t_n)$ in α contains no bound variable of $\beta(x_1,\ldots,x_n)$*
- *the result of the substitution is a well-formed formula*

See III §5 in [3].

The prohibition to use additional subject variables within the replacement formula that occur bound in the original formula assurs that the resulting formula is well-formed. Furthermore it preserves the validity of the formla. Take a look at the following derivation.

$$\begin{array}{rcll}
\phi(x) & \rightarrow & \exists y\ \phi(y) & \text{with axiom 6} \\
(\exists y\ y = y) \land \phi(x) & \rightarrow & \exists y\ \phi(y) & \\
\exists y\ (y = y \land \phi(x)) & \rightarrow & \exists y\ \phi(y) & \\
\exists y\ (y = y \land x \neq y) & \rightarrow & \exists y\ y \neq y & \text{forbidden replacment: } \phi(x) \text{ by } x \neq y\text{,} \\
 & & & \text{despite } y \text{ is already bound} \\
\exists y\ x \neq y & \rightarrow & \exists y\ y \neq y &
\end{array}$$

The last proposition is not valid in many models.

Analogous to rule 4 we can replace function variables too.

Rule 5 (Replace Function Variable). `[rule:replaceFunct]`

Name: `replaceFunct` - Version: `0.01.00`

Let α *be an already proved formula that contains a function variable* σ *of arity* n*, let* $x_1, \ldots, x_n$ *be pairwise different subject variables and let* $\tau(x_1, \ldots, x_n)$ *be an arbitrary term where* $x_1, \ldots, x_n$ *are not bound. The term* $\tau(x_1, \ldots, x_n)$ *must not contain all* $x_1, \ldots, x_n$ *as free subject variables. Furthermore it can also have other subject variables either free or bound.*

If the following conditions are fulfilled we can obtain a new true formula by replacing each occurrence of $\sigma(t_1, \ldots, t_n)$ *with appropriate terms* $t_1, \ldots, t_n$ *in* α *by* $\tau(t_1, \ldots, t_n)$*.*

- *the free variables of* $\tau(x_1, \ldots, x_n)$ *without* $x_1, \ldots, x_n$ *do not occur as bound variables in* α
- *each occurrence of* $\sigma(t_1, \ldots, t_n)$ *in* α *contains no bound variable of* $\tau(x_1, \ldots, x_n)$
- *the result of the substitution is a well-formed formula*

Rule 6 (Universal Generalization). `[rule:universalGeneralization]`

Name: `universalGeneralization` - Version: `0.01.00`

If $\alpha \rightarrow \beta(x_1)$ *is a true formula and* α *does not contain the subject variable* x_1*, then* $\alpha \rightarrow (\forall x_1\ (\beta(x_1)))$ *is a true formula too.*

Rule 7 (Existential Generalization). `[rule:existentialGeneralization]`

Name: `existentialGeneralization` - Version: `0.01.00`

If $\alpha(x_1) \rightarrow \beta$ *is already proved to be true and* β *does not contain the subject variable* x_1*, then* $(\exists x_1\ \alpha(x_1)) \rightarrow \beta$ *is also a true formula.*

The usage and elimination of abbreviations and constants is also an inference rule. In many texts about mathematical logic these rules are not explicitly stated and this text is no exception. But in the exact QEDEQ format corresponding rules exist.

Chapter 3

Derived Propositions

Now we derive elementary propositions with the axioms and rules of inference of chapter 1.

3.1 Propositional Calculus

At first we look at the propositional calculus.

To define the predicate *true* we just combine a predicate and its negation.

Definition 1 (True). `[definition:True]`

$$\top \;\leftrightarrow\; (A \;\vee\; \neg A)$$

For a precise definition we should have written something like $p_0^0 = \top$ and $\top :\leftrightarrow (A \wedge \not A)$.[1] In the future we only write the symbol itself. It's arity should be evident from the formula.

For the predicate *false* we just negate *true*.[2]

Definition 2 (False). `[definition:False]`

$$\bot \;\leftrightarrow\; \neg\top$$

We have the following basic propositions.

Proposition 1 (Basic Propositions). `[theorem:propositionalCalculus]`

$$\top \tag{aa}$$

$$\neg\bot \tag{ab}$$

$$A \;\rightarrow\; A \tag{ac}$$

$$A \;\leftrightarrow\; A \tag{ad}$$

$$(A \;\vee\; B) \;\leftrightarrow\; (B \;\vee\; A) \tag{ae}$$

$$(A \;\wedge\; B) \;\leftrightarrow\; (B \;\wedge\; A) \tag{af}$$

$$(A \;\wedge\; B) \;\rightarrow\; A \tag{ag}$$

$$(A \;\leftrightarrow\; B) \;\leftrightarrow\; (B \;\leftrightarrow\; A) \tag{ah}$$

$$(A \;\vee\; (B \;\vee\; C)) \;\leftrightarrow\; ((A \;\vee\; B) \;\vee\; C) \tag{ai}$$

$$(A \;\wedge\; (B \;\wedge\; C)) \;\leftrightarrow\; ((A \;\wedge\; B) \;\wedge\; C) \tag{aj}$$

$$A \;\leftrightarrow\; (A \;\vee\; A) \tag{ak}$$

---

[1] In the deeper laying (see `http://www.qedeq.org/current/doc/project/qedeq_logic_language_en.pdf`) formal language this predicate has the name *TRUE* and zero arguments. So we just have to map names to natural numbers to fulfill the exact definition.

[2] Analogous to the preceeding definition we can set $p_1^0 = \bot$

$$A \leftrightarrow (A \land A) \tag{al}$$
$$A \leftrightarrow \neg\neg A \tag{am}$$
$$(A \rightarrow B) \leftrightarrow (\neg B \rightarrow \neg A) \tag{an}$$
$$(A \leftrightarrow B) \leftrightarrow (\neg A \leftrightarrow \neg B) \tag{ao}$$
$$(A \rightarrow (B \rightarrow C)) \leftrightarrow (B \rightarrow (A \rightarrow C)) \tag{ap}$$
$$\neg(A \lor B) \leftrightarrow (\neg A \land \neg B) \tag{aq}$$
$$\neg(A \land B) \leftrightarrow (\neg A \lor \neg B) \tag{ar}$$
$$(A \lor (B \land C)) \leftrightarrow ((A \lor B) \land (A \lor C)) \tag{as}$$
$$(A \land (B \lor C)) \leftrightarrow ((A \land B) \lor (A \land C)) \tag{at}$$
$$(A \land \top) \leftrightarrow A \tag{au}$$
$$(A \land \bot) \leftrightarrow \bot \tag{av}$$
$$(A \lor \top) \leftrightarrow \top \tag{aw}$$
$$(A \lor \bot) \leftrightarrow A \tag{ax}$$
$$(A \lor \neg A) \leftrightarrow \top \tag{ay}$$
$$(A \land \neg A) \leftrightarrow \bot \tag{az}$$
$$(\top \rightarrow A) \leftrightarrow A \tag{ba}$$
$$(\bot \rightarrow A) \leftrightarrow \top \tag{bb}$$
$$(A \rightarrow \bot) \leftrightarrow \neg A \tag{bc}$$
$$(A \rightarrow \top) \leftrightarrow \top \tag{bd}$$
$$(A \leftrightarrow \top) \leftrightarrow A \tag{be}$$
$$((A \rightarrow B) \land (B \rightarrow C)) \rightarrow (A \rightarrow C) \tag{bf}$$
$$((A \leftrightarrow B) \land (C \leftrightarrow B)) \rightarrow (A \leftrightarrow C) \tag{bg}$$
$$((A \land B) \leftrightarrow (A \land C)) \leftrightarrow (A \rightarrow (B \leftrightarrow C)) \tag{bh}$$
$$((A \land B) \leftrightarrow (A \land \neg B)) \leftrightarrow \neg A \tag{bi}$$
$$(A \leftrightarrow (A \land B)) \leftrightarrow (A \rightarrow B) \tag{bj}$$
$$(A \rightarrow B) \rightarrow ((A \land C) \rightarrow (B \land C)) \tag{bk}$$
$$(A \leftrightarrow B) \rightarrow ((A \land C) \leftrightarrow (B \land C)) \tag{bl}$$
$$(A \land (A \rightarrow B)) \rightarrow B \tag{bm}$$
$$(A \land (A \rightarrow B)) \leftrightarrow (A \land B) \tag{bn}$$

3.2 Predicate Calculus

For predicate calculus we achieve the following propositions.

We have the following basic propositions.

Proposition 2 (Basic Propositions). `[theorem:predicateCalculus]`

$$\forall x\ (\phi(x) \rightarrow \psi(x)) \rightarrow (\forall x\ \phi(x) \rightarrow \forall x\ \psi(x)) \tag{a}$$
$$\forall x\ (\phi(x) \rightarrow \psi(x)) \rightarrow (\exists x\ \phi(x) \rightarrow \exists x\ \psi(x)) \tag{b}$$
$$\forall x\ (\phi(x) \leftrightarrow \psi(x)) \rightarrow (\forall x\ \phi(x) \leftrightarrow \forall x\ \psi(x)) \tag{c}$$
$$\exists x\ (\phi(x) \land \psi(x)) \rightarrow (\exists x\ \phi(x) \land \exists x\ \psi(x)) \tag{d}$$
$$(\forall x\ \psi(x) \lor \forall x\ \psi(x)) \rightarrow \forall x\ (\phi(x) \lor \psi(x)) \tag{e}$$
$$\exists x\ (\phi(x) \lor \psi(x)) \leftrightarrow (\exists x\ \phi(x) \lor \exists x\ \psi(x)) \tag{f}$$
$$\forall x\ (\phi(x) \land \psi(x)) \leftrightarrow (\forall x\ \phi(x) \land \forall x\ \psi(x)) \tag{g}$$
$$\forall x\ \forall y\ \phi(x,y) \leftrightarrow \forall y\ \forall x\ \phi(x,y) \tag{h}$$
$$\exists x\ \exists y\ \phi(x,y) \leftrightarrow \exists y\ \exists x\ \phi(x,y) \tag{i}$$
$$\forall x\ (\phi(x) \rightarrow A) \rightarrow (\forall x\ \phi(x) \rightarrow A) \tag{j}$$
$$\forall x\ (A \rightarrow \phi(x)) \leftrightarrow (A \rightarrow \forall x\ \phi(x)) \tag{k}$$
$$\forall x\ (\phi(x) \land A) \leftrightarrow (\forall x\ \phi(x) \land A) \tag{l}$$
$$\forall x\ (\phi(x) \lor A) \leftrightarrow (\forall x\ \phi(x) \lor A) \tag{m}$$
$$\forall x\ (\phi(x) \leftrightarrow A) \rightarrow (\forall x\ \phi(x) \leftrightarrow A) \tag{n}$$
$$\forall x\ (\phi(x) \leftrightarrow \psi(x)) \rightarrow (\forall x\ \phi(x) \leftrightarrow \forall x\ \psi(x)) \tag{o}$$

3.3 Derived Rules

Beginning with the logical basis logical propositions and metarules can be derived an enable a convenient argumentation. Only with these metarules and additional definitions and abbreviations the mathematical world is unfolded. Every additional syntax is *conservative*. That means: within extended system no formulas can be derived, that are written in the old syntax but can not be derived in the old system. In the following such conservative extensions are introduced.

Rule 8 (Replace by Logical Equivalent Formula). `[rule:replaceEquiFormula]`

Name: `replaceEquiFormula` - Version: `1.00.00`

Let the formula $\alpha \leftrightarrow \beta$ *be true. If in a formula* δ *we replace an arbitrary occurence of* α *by* β *and the result* γ *is also a formula*[3] *and contains all the free subject variables of* δ*, then* $\delta \leftrightarrow \gamma$ *is a true formula.*

Rule 9 (Replacement of $\top$ by already derived formula). `[rule:replaceTrueByTrueFormula]`

Name: `replaceTrueByTrueFormula` - Version: `1.00.00`

Let α *be an already derived true formula and* β *a formula that contains* $\top$*. If we get a well formed formula* γ *by replacing an arbitray occurence of* $\top$ *in* β *with* α *then the following formula is also true:* $\beta \leftrightarrow \gamma$

Rule 10 (Replacement of already derived formula by $\top$). `[rule:replaceTrueFormulaByTrue]`

Name: `replaceTrueFormulaByTrue` - Version: `1.00.00`

Let α *be an already derived true formula and* β *a formula that contains* α*. If we get a well formed formula* γ *by replacing an arbitray occurence of* α *in* β *by* $\top$ *then the following formula is also true:* $\beta \leftrightarrow \gamma$

Rule 11 (Derived Quantification). `[rule:derivedQuantification]`

Name: `derivedQuantification` - Version: `1.00.00`

If we have already derived the true formula $\alpha(x)$ *and* x *is not bound in* $\alpha(x)$ *then the formula* $\forall x\ \alpha(x)$ *is also true.*

Rule 12 (General Associativity). `[rule:generalAssociativity]`

Name: `generalAssociativity` - Version: `1.00.00`

If an operator of arity two fulfills the associative law it also fulfills the general associative law. The operator can be extended to an operator of arbitrary arity greater one. For example: instead of $(a+b)+(c+d)$ *we simply write* $a+b+c+d$*.*[4]

Rule 13 (General Commutativity). `[rule:generalCommutativity]`

Name: `generalCommutativity` - Version: `1.00.00`

If an operator fulfills the general associative law and is commutative then all permutations of parameters are equal or equivalent.[5] *For example we have:* $a+b+c+d = c+a+d+b$*.*

Rule 14 (Deducible from Formula). `[rule:definitionDeductionFromFormula]`

Name: `definitionDeducibleFromFormula` - Version: `1.00.00`

We shall say that the formula β *is* deducible from the formula α *if the formula* β *from the totality of all true formulas of the predicate calculus and the formula*

---

[3] During that substitution it might be necessary to rename bound variables of β.

[4] The operator of arity n is defined with a certain bracketing, but every other bracketing gives the same result.

[5] That depends on the operator type: term or formula operator.

α by means of application of all the rules of the predicate calculus, in which connection both rules for binding by a quantifier, the rules for substitution in place of predicate variables and in place of free subject variables must be applied only to predicate variables or subject variables which do not occur in the formula α and $\alpha \rightarrow \beta$ is a formula.

Notation: $\alpha \vdash \beta$.

That a formula β is deducible from the formula α must be strictly distinguished from the deduction of a true formula from the axioms of the predicate calculus. In the second case more derivation rules are available. For example if A is added to the axioms then the formula B can be derived. But B is not deducible from A.[6]

Rule 15 (Deduction Theorem). `[rule:deductionTheorem]`

Name: `deductionTheorem` - Version: `1.00.00`

If the formula β is deducible from the formula α, then the formula $\alpha \rightarrow \beta$ can be derived from the predicate calculus.

---

[6] We must make this distinction due to possible substitution rule usages.

Chapter 4

Identity

Everything that exists has a specific nature. Each entity exists as something in particular and it has characteristics that are a part of what it is. Identity is whatever makes an entity definable and recognizable, in terms of possessing a set of qualities or characteristics that distinguish it from entities of a different type. An entity can have more than one characteristic, but any characteristic it has is a part of its identity.

4.1 Identity Axioms

We start with the identy axioms.

We define a predicate constant of arity two that shall stand for the identity of subjects.

Initial Definition 3 (Identity). `[definition:identity]`

$$x \; = \; y$$

For convenience we also define the negation of the identity a predicate constant.

Definition 4 (Not Identical). `[definition:notEqual]`

$$x \; \neq \; y \; \leftrightarrow \; \neg x \; = \; y$$

Axiom 7 (Reflexivity of Identity). `[axiom:identityIsReflexive]`

$$x \; = \; x$$

Axiom 8 (Leibniz' replacement). `[axiom:leibnizReplacement]`

$$x \; = \; y \; \rightarrow \; (\phi(x) \; \rightarrow \; \phi(y))$$

Axiom 9 (Symmetrie of identity). `[axiom:symmetryOfIdentity]`

$$x \; = \; y \; \rightarrow \; y \; = \; x$$

Axiom 10 (Transitivity of identity). `[axiom:transitivityOfIdentity]`

$$(x \; = \; y \; \wedge \; y \; = \; z) \; \rightarrow \; x \; = \; z$$

We can reverse the second implication in the Leibniz replacement.

Proposition 3. [theorem:leibnizEquivalence]

$$x \; = \; y \; \rightarrow \; (\phi(x) \; \leftrightarrow \; \phi(y))$$

Proposition 4. [theorem:identyImpliesFunctionalEquality]

$$x \; = \; y \; \rightarrow \; f(x) \; = \; f(y)$$

4.2 Restricted Quantifiers

Every quantification involves one specific subject variable and a domain of discourse or range of quantification of that variable. Until now we assumed a fixed domain of discourse for every quantification. Specification of the range of quantification allows us to express that a predicate holds only for a restricted domain.

At the following definition the replacement formula for $\alpha(x)$ must „reveal“ its quantification subject variable. This is usually the first following subject variable.[1] In the exact syntax of the QEDEQ format[2] the quantification subject variable is always given.

Axiom 11 (Restricted Universal Quantifier)**.** [axiom:restrictedUniversalQuantifier]

$$\forall \; \alpha(x) \;\; \beta(x) \; \leftrightarrow \; \forall x \; (\alpha(x) \; \rightarrow \; \beta(x))$$

A matching definiton for the restricted existential quantifier is the following.[3]

Axiom 12 (Restricted Existential Quantifier)**.** [axiom:restrictedExistentialQuantifier]

$$\exists \; \alpha(x) \;\; \beta(x) \; \leftrightarrow \; \exists x \; (\alpha(x) \; \land \; \beta(x))$$

For restricted quantifiers we find formulas according to Proposition proposition 2.
+++

To express the existence of only one individuum with a certain property we introduce a new quantifier.

Axiom 13 (Restricted Uniqueness Quantifier)**.** [axiom:restrictedUniquenessQuantifier]

$$\exists! \; \alpha(x) \;\; \beta(x) \; \leftrightarrow \; \exists \; \alpha(x) \;\; (\beta(x) \; \land \; \forall \; \alpha(y) \;\; (\beta(y) \; \rightarrow \; x \; = \; y))$$

Rule 16 (Term Definition by Formula)**.** *If the formula $\exists!x \; \alpha(x)$ holds, we can expand the term syntax by $D(x, \alpha(x))$. May the formula $\alpha(x)$ not contain the variable y and let $\beta(y)$ be a formula that doesn't contain the variable x. Then we define a new formula $\beta(D(x, \alpha(x)))$ by $\beta(y) \land \exists!x \; (\alpha(x) \land x \; = \; y)$. Also in this abbreviate notation the subject variable x counts as bound, the subject variable y is arbitrary (if it fulfills the given conditions) and will be ignored in the abbreviation. Changes in α that lead to another formula α' because of variable collision with β must also be done in the abbreviation. All term building rules are extended accordingly. The expression is also replaceble by $\exists!y \; (\beta(y) \land \alpha(y)$ or by $\beta(y) \land \alpha(y)$.*

---

[1] For example: in the following formula we identify the subject variable m for the second quantification: $\forall \; n \in \mathbb{N} \; \forall \; m \in n \; m < n$.

[2] Again see `http://www.qedeq.org/current/xml/qedeq/`.

[3] Matching because of $\neg\forall \; \psi(x) \; (\phi(x)) \leftrightarrow \exists \; x \; \neg(\psi(x) \rightarrow \phi(x)) \leftrightarrow \exists \; x \; (\psi(x) \land \neg\phi(x)) \leftrightarrow \exists \; \psi(x) \; (\neg\phi(x))$.

Bibliography

[1] *A.N. Whitehead, B. Russell*, Principia Mathematica, Cambridge University Press, London 1910

[2] *P. Bernays*, Axiomatische Untersuchung des Aussagen-Kalkuls der „Principia Mathematica“, Math. Zeitschr. 25 (1926), 305-320

[3] *D. Hilbert, W. Ackermann*, Grundzüge der theoretischen Logik, 2nd ed., Berlin: Springer, 1938. English version: Principles of Mathematical Logic, Chelsea, New York 1950, ed. by R. E. Luce. See also `http://www.math.uwaterloo.ca/~snburris/htdocs/scav/hilbert/hilbert.html` 15, 17

[4] *P.S. Novikov*, Elements of Mathematical Logic, Edinburgh: Oliver and Boyd, 1964. 15

[5] *E. Mendelson*, Introduction to Mathematical Logic, 3rd. ed., Belmont, CA: Wadsworth, 1987. 13

[6] *V. Günther*, Lecture „Mathematik und Logik“, given at the University of Hamburg, 1994/1995.

[7] *M. Meyling*, Hilbert II, Presentation of Formal Correct Mathematical Knowledge, Basic Concept, `http://www.qedeq.org/current/doc/project/qedeq_basic_concept_en.pdf`.

[8] qedeq_set_theory_v1 `http://www.qedeq.org/0_04_08/doc/math/qedeq_set_theory_v1.xml`

Index