

Hilbert II

Darstellung von formal korrektem
mathematischen Wissen

Grobkonzept

Version 0.62

Michael Meyling

6. August 2004

Homepage von **Hilbert II**: http://www.qedeq.org/index_de.html

Dieses Dokument ist in elektronischer Form an dem folgenden Ort zu finden:
http://www.qedeq.org/0_01_05/projektbeschreibung.pdf

Die vorliegende Publikation ist urheberrechtlich geschützt:

Copyright © 2004 Michael Meyling. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in appendix [A](#) entitled *GNU Free Documentation License*.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären. Die in diesem Text erwähnten Software- und Hardwarebezeichnungen sind in den meisten Fällen auch eingetragene Warenzeichen und unterliegen als solche den gesetzlichen Bestimmungen.

Inhaltsverzeichnis

Zusammenfassung und Management Summary	5
Vorwort	7
Einleitung	9
1 Ziele der Darstellung	11
1.1 Freie Verfügbarkeit	11
1.2 Formale Form und andere Ansichten	11
1.3 Textbuch und Referenzen	12
1.4 Formale Korrektheit	12
1.5 Vernetzung	12
1.6 Sprachen	12
1.7 Detailgrad	13
1.8 Metaregeln und Versionen	13
1.9 Erzeugung und Bearbeitung	13
2 Mathematische Ausgangspunkte	15
2.1 Logische Axiome	15
2.2 Identität	18
2.3 Eingeschränkte Quantoren	19
2.4 Axiome der Mengenlehre	19
2.5 Literaturverzeichnis	22
3 Technischer Rahmen	25
3.1 Kernel	25
3.2 Organisation von Anwendungen	25
3.3 Qedeq-Format	26
3.4 L ^A T _E X	28
3.5 XML	28
3.6 Formaterweiterungen	28
3.7 Programmiersprache	29
4 Prototyp	31
4.1 Qedeq-Module	31
4.2 Funktionalität	31
5 Arbeitsbereiche	33
5.1 Erstellung von mathematischen Texten	33
5.1.1 Mathematik	33
5.1.2 Formalisierung	33
5.1.3 Weitere Ergänzungen	33
5.2 Qedeq-Syntaxdefinition	34
5.3 Kernelentwicklung	34
5.4 Anwendungsentwicklung	34
5.5 Test und Qualitätssicherung	34

5.6 Übersetzung	34
5.7 Releasemanagement und Distribution	34
5.8 Support und Fehlerverfolgung	35
5.9 Organisation des Internet-Auftritts	35
5.10 Dokumentation und Redaktion	35
5.11 Gestaltung und Design	35
5.12 Öffentlichkeitsarbeit und Kontaktpflege	35
5.13 Gesamtkoordination	35
6 Planung	37
6.1 Umfang der Version 1.00	37
6.2 Umfang der Version 1.01	37
6.3 Umfang der Version 1.02	37
6.4 Umfang der Version 1.03	38
6.5 Weitere Aufgaben	38
7 Technische Realisierung	39
8 Einsatzszenarien	41
8.1 Ausstattungen	41
8.2 Mengengerüst	41
8.3 Verwendung	42
A GNU Free Documentation License	43
B Qedeq-Format	49
Literaturverzeichnis	59
Index	59

Zusammenfassung und Management Summary

Das Projekt **Hilbert II**, beschäftigt sich mit der Darstellung und Dokumentation von mathematischem Wissen. Dazu stellt **Hilbert II** eine Programmsuite zur Verwirklichung unterschiedlicher Aufgaben bereit. Auch die konkrete Dokumentation mathematischen Grundlagenwissens mit diesen Hilfsmitteln gehört zum Ziel dieses Projekts.

Dieses Dokument ist eine grobe, fachliche, DV-unabhängige Leistungsbeschreibung des zu realisierenden Softwaresystems und der zugehörigen mathematischen Grundlagen. Es werden die grobe Funktionalität und die wesentlichen Daten der Anwendung beschrieben. Dieses Grobkonzept soll es der Mathematikerin erlauben, Inhalt und Aufbau des Softwaresystems zu verstehen und zu beurteilen. Auch einige technische Hinweise und Vorgaben werden angegeben. Dieses Konzept dient auch als Referenz und Entscheidungsgrundlage für das weitere Vorgehen.

Das mathematische Wissen soll in *formal korrekter* Form *dezentral* im Internet *frei verfügbar* gemacht werden. Dazu wird das Wissen in sogenannten *Qedeq-Modulen* organisiert, die analog zu mathematischen Textbüchern aufgebaut sind. Wesentlich ist dabei die Möglichkeit zur formal korrekten Darstellung in einem Prädikatenkalkül erster Stufe, die eine maschinelle Auswertung ermöglicht. Dazu ist ein *Proofchecker* integriert, der die logische Folgerichtigkeit der formalen Beweise überprüft.¹ Ein *Qedeq-Viewer* ermöglicht eine komfortable Betrachtung der Hypertexte. Weitere Analysen, z. B. zur Satz- und Axiomabhängigkeit, sind ebenfalls vorgesehen. Aus den Qedeq-Modulen können mathematische Textbücher im L^AT_EX-, PDF-, HTML- oder XML-Format generiert werden. Durch die Verweismöglichkeit auf andere Qedeq-Module kann im Internet ein riesiges mathematisches Wissensnetz aufgebaut werden.

Dabei ist die formale Korrektheit der dargestellten Sätze nicht erforderlich, um mathematisches Wissen in Qedeq-Modulen abzubilden. Schon in den ersten Phasen des Projekts kann **Hilbert II** mathematisches Wissen aufnehmen. Mathematische Erkenntnisse können bereits frühzeitig in Qedeq-Form gebracht und erst später in formal korrekter Form erfasst werden. Das Qedeq-Format ermöglicht die Darstellung verschiedener Detaillierungsstufen von mathematischer Theorie innerhalb eines Moduls. So können in der niedrigsten Detaillierung die Hauptsätze betrachtet werden, in weiteren Stufen sind informale Beweise und weitere Informationen zu sehen und im letzten Schritt sollten dann formale Beweise stehen, deren Korrektheit mit dem Proofchecker überprüft wurde. In dem Qedeq-Viewer kann zwischen den Detaillierungsstufen und gegebenenfalls

¹Dabei ist **Hilbert II** kein Theorembeweiser oder Theoremfinder im herkömmlichen Sinne. Lediglich Zwischenstritte innerhalb von formalen Beweisen werden ermittelt. Dennoch wird es für die Bequemlichkeit entscheidend sein, ob ein dem „normalen mathematischen Gebrauch“ entsprechender Übergang von einer Beweiszeile zur nächsten von **Hilbert II** ermittelt werden kann.

zwischen verschiedenen Sprachen gewechselt werden, außerdem ist zu erkennen, welchen Korrektheitsgrad ein Satz besitzt.²

Weitere Einzelheiten zum aktuellen Stand des dieses internationalen Open-Source-Projekts sind auf der Homepage dieses Projekts zu finden: http://www.qedeq.org/index_de.html.

Vorwort und Einleitung beschreiben die Ausgangssituation und Problemstellung. Im Kapitel 1 werden die Anforderungen formuliert, die an die Darstellung mathematischen Wissens gestellt werden. Die mathematischen Grundlagen, die der Konzeption und den Textbüchern von **Hilbert II** zugrundeliegen, werden in Kapitel 2 formuliert. Auf technische Einzelheiten wird in Kapitel 3 eingegangen. Es gibt bereits einen funktionierenden Prototypen, welcher in Kapitel 4 vorgestellt wird. Die Arbeitsbereiche dieses Projekts werden in Kapitel 5 beschrieben. Das Kapitel 6 beschäftigt sich mit der initialen Planung des Projektverlaufs. Entwicklungsleitlinien für die Programmentwicklung sind in Kapitel 7 zu finden. Um die Einsatzszenarien geht es in Kapitel 8. Die Lizenzbedingungen für dieses Dokument sind in Anhang A angegeben.

Dieses Dokument ist noch in Arbeit und wird ständig aktualisiert. Insbesondere sind an den durch „+++“ gekennzeichneten Stellen noch Ergänzungen vorzunehmen.

²D. h. es ist erkennbar, ob ein formal korrekter Beweis vorliegt, und die formale Korrektheit der beim Beweis verwendeten anderen mathematischen Sätze gegeben ist. Der Korrektheitsgrad steigt je nach Anzahl der erfüllten Bedingungen.

Vorwort

Mathematik ist eine Wissenschaft, deren Wissensgebäude im Laufe der Zeit gigantische Ausmaße angenommen hat. Gebaut auf ein Fundament aus wenigen mengentheoretischen Axiomen, lässt sich dieses Gebäude mit prädikatenlogischem Mörtel in schwindelnde Höhen errichten. Dieser Aufbau kann im Prinzip von jeder Mathematikerin nachvollzogen werden. Auch von dem letzten Türmchen neuester mathematische Erkenntnis kann der Weg ganz nach unten bis in das Fundament verfolgt werden.

Praktisch durchführbar ist dieser Weg jedoch kaum. Zumal für Mathematikerinnen mit anderer Spezialdisziplin ist es nicht möglich, alle Detailschritte bis in die Niederungen des elementar Einfachen zu überblicken oder zu verfolgen. Es fehlt einfach die Zeit diesen langen Weg zurückzulegen. Mathematik ist auch die Wissenschaft der Abstraktion, der Untersuchung von Strukturen. Und so thronen „schwere“³ Aussagen und ihre Beweise häufig auf einem Stapel neuer abstrakter Strukturen, die sich die Leserin erst einmal aneignen muss. Daher ist die Voraussetzung für das Lesen von mathematischen Artikeln in der Regel die Kenntnis von mathematischen Standardwerken für die jeweilige Disziplin. Die darin benutzten Nomenklaturen, Definitionen und Konventionen unterscheiden sich nicht nur von Disziplin zu Disziplin, sondern auch schon innerhalb eines Gebietes.

Dazu sind bestimmte mathematische Werke nicht einfach zu bekommen. So sie denn noch käuflich zu erwerben sind, ist ihr Preis für viele unerschwinglich und zu Bibliotheken haben leider die meisten Menschen keinen Zugang.⁴ Dabei sind mathematische Erkenntnisse öffentliches Kulturgut und sollten der Öffentlichkeit frei zur Verfügung stehen. Aber selbst dann, wenn eine Bibliothek in der Nähe und das entsprechende Werk im Bestand vorhanden und nicht verliehen ist, sind auch in Standardwerken nicht alle mathematischen Aussagen richtig, sie enthalten kleinere und größere Fehler.

Gerade in der mathematischen Spitzforschung kommt es gelegentlich vor, dass nur wenige Menschen die Beweisschritte für einen mathematischen Satz nachvollziehen können und deshalb die Verantwortung für die Korrektheit der mathematischen Argumentation auf wenige Schultern verteilt ist.

Auch die Verwendung von Erkenntnissen aus anderen mathematischen Disziplinen in der eigenen gestaltet sich häufig als schwierig. Die benötigten Voraussetzungen, Vorbedingungen und notwendigen Definitionen können für eine in der Disziplin nicht bewanderte Mathematikerin nicht einfach ermittelt und dann herausgelöst werden.⁵ Wünschenswert ist ein Hilfsmittel zur automatischen Auflistung dieser Bedingungen und ihre Überprüfung bei der Anwendung.

³ „Schwer“ bedeutet nicht unbedingt lang und kompliziert, sondern dass für den Beweis Ideen benötigt werden, die nicht naheliegend, also *schwer zu finden* sind.

⁴ Die Lebensumstände der meisten Menschen dieser Erde sind mit denen in Deutschland vorherrschenden nicht vergleichbar.

⁵ Das zeigt sich schon an ganz einfachen Beispielen. Beim Nachschlagen in Büchern können manche Voraussetzungen leicht übersehen werden. Z. B.: „In diesem Kapitel bedeutet *Körper* stets einen *kommutativen Körper*.“ „Das Prädikat \subset beinhaltet auch die Gleichheit.“

Welche Anforderungen können aus den zuvor formulierten Problemen nun abgeleitet werden?

Mathematische Standardwerke sollten frei verfügbar sein. Die darin getroffenen mathematischen Aussagen sollten in einer normierten Sprache verfasst und auch formal nachvollziehbar sein.

Gerade in unserem elektronischen Zeitalter bieten sich Lösungsmöglichkeiten für diese Anforderungen an. Die Erarbeitung und Umsetzung solcher Lösungen ist das Ziel von **Hilbert II**.

Dank gebührt den Professoren *W. Kerby* und *V. Günther* der Hamburger Universität für ihre inspirierenden Vorlesungen zu den Themen Logik und Axiomatische Mengenlehre. Ohne diese entscheidenden Impulse hätte es dieses Projekt nie gegeben.

Dank auch an *F. Fritzsche* für zahllose Korrekturvorschläge und inhaltliche Anregungen.

Einleitung

Mathematik beschäftigt sich mit abstrakten Strukturen, welche keinen direkten Bezug zur „realen“ Welt haben. Diese abstrakten Strukturen sind sehr gut formalisierbar.⁶ Die Exaktheit der Mathematik erfordert diesen hohen Grad der Formalisierbarkeit, es kann sogar behauptet werden, dass die meisten Gebiete der Mathematik in einer einfachen formalen Sprache vollständig beschrieben werden können.⁷

Grundlage der mathematischen Argumentation ist die sogenannte *Prädikatenlogik*. Diese wiederum fußt auf der *Aussagenlogik*. Jeder *mathematischen Aussage* kann genau ein Wahrheitswert *wahr*, beziehungsweise *falsch* zugeordnet werden. Mathematische Aussagen können durch *logische Operatoren* zu neuen mathematischen Aussagen verknüpft werden, und dieser Verknüpfung kann dann ebenfalls ein Wahrheitswert zugeordnet werden.

Mathematische Aussagen beziehen sich auf *mathematische Objekte*, denen gewisse *Prädikate* zuerkannt werden. Ausgehend von bestimmten Grundaussagen, den *Axiomen*, welche als wahr angesehen werden, können nun mit den *Schlussregeln* der Prädikatenlogik neue mathematische Aussagen abgeleitet werden, die ebenfalls als wahr gelten. Mithilfe einfacher, an den Anfang gestellter Prädikate können dann im weiteren Verlauf neue, kompliziertere Prädikate *definiert* werden.

Historische Einbettung

Die *mathematische Logik* untersucht die Grundlagen mathematischer Beweisführung. Sie wird auch *formale Logik*, *symbolische Logik*, *Logistik* oder *theoretische Logik* genannt und unterscheidet sich von anderen Gestalten der Logik. So besitzt sie eine formalistische Methodik, die sich in Form eines *Kalküls* beschreiben lässt. Dieses formale System behandelt nur die Gestalt von Zeichen und beschäftigt sich nicht mit der Deutung dieser Symbole.⁸

Die von den Arabern eingeführten mathematischen Methoden haben den Spanier R. Lullus (ca. 1235-1315) angeregt zu seiner berühmten *Ars Magna*, die dem Wunsch nach einer schematischen Erzeugung aller wahren Aussagen entsprang. Obwohl die Ideen von Lullus von geringem mathematischen Wert sind, hatten sie einen großen Einfluss auf die Nachwelt: noch zweihundert Jahre später veröffentlichte Cardano seine algebraischen Algorithmen als Teil der *Ars Magna*.

⁶Dabei besteht der Anspruch, dass im Prinzip jeder Mensch die mathematischen Argumentationen nachvollziehen kann. Obwohl mathematische Grundlagen erst erlernt werden müssen, so scheint die Fähigkeit zum mathematischen Denken doch bei jedem Menschen vorhanden zu sein, wenn auch in unterschiedlicher Stärke.

⁷Über die Einschränkung durch die Verwendung des Wortes „meistens“ wird im Folgenden noch eingegangen. Die Frage, ob das Wesen der Mathematik in einer formalen Sprache überhaupt erfasst werden kann, stellt sich für **Hilbert II** nicht, da in diesem Projekt auch die natürliche Sprache Eingang findet.

⁸Psychologische, erkenntnistheoretische und metaphysische Fragen werden in der mathematischen Logik nicht behandelt.

G. W. Leibniz formulierte als einer der ersten die Idee einer formalen Logik. Als eigentlicher Begründer der mathematischen Logik gilt G. Boole, der unter anderem im Jahr 1854 das Werk *An Investigation of the laws of thought on which are founded the mathematical theories of logic and probabilities* veröffentlichte. C. S. Peirce erweiterte diesen Ansatz auf die Prädikatenlogik. Neben vielen weiteren Logikern ist G. Frege hervorzuheben. Seine 1879 publizierte *Begriffschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens* enthält bereits ein vollständiges Kalkül der Junktoren- und Quantorenlogik. G. S. Peanos Symbolik floss in das epochale Werk *Principia Mathematica* (1910-1913) von A. N. Whitehead and B. Russell ein. In diesem Werk wird ein Kalkül vorgestellt, der nicht es nicht nur gestattet die Logik, sondern auch große Teile der Mathematik aus wenigen Axiomen herzuleiten.

+++ Bedeutung der Principia Mathematica

+++ Hilbert, alles ist formalisierbar

+++ Ergebnisse von Gödel

Ausblick

Ist durch dieses letzte Ergebnis von Gödel nicht jeglicher Versuch, die Mathematik zu formalisieren, überflüssig? Gödels Beweis lässt sich auch innerhalb einer geeigneten formalen Sprache führen. Dabei kann sich ein solches System auch selbst betrachten, ohne diese Selbstbezüglichkeit zu kennen. Innerhalb dieses Systems wird dann nachgewiesen, dass das diesem System nachgebildete innere formale System unvollständig ist. Da also selbst die Unvollständigkeit in einem formalen System erfasst werden kann, ist in diesem Sinne die **gesamte** Mathematik formalisierbar.

Kapitel 1

Ziele der Darstellung

Mathematik besteht aus mathematischem Wissen. Zur Vermittlung und Kommunikation muss dieses Wissen dargestellt werden. In diesem Kapitel werden, in dem durch das Vorwort und die Einleitung abgesteckten Rahmen, Anforderungen für die Darstellung von mathematischem Wissen formuliert. Damit werden die Ansprüche und Ziele definiert, welche durch **Hilbert II** verwirklicht werden sollen.

1.1 Freie Verfügbarkeit

Mathematisches Wissen soll *frei* verfügbar gemacht werden. Alle mathematischen Texte von **Hilbert II** unterliegen der *GNU Free Documentation License*. Der Zweck dieser Lizenz ist es, Dokumente frei zu halten: jeder und jedem die effektive Freiheit zu sichern, Dokumente zu kopieren und weiter zu verteilen, mit oder ohne Änderung, entweder kommerziell oder nichtkommerziell. Dabei müssen alle abgeleitete Arbeiten der Dokumente selbst wieder im selben Sinne frei sein. Diese Lizenz gilt auch für dieses Dokument, siehe auch Anhang A *GNU Free Documentation License*.

Verfügbar soll dieses Wissen in *elektronischer* Form sein. Zum einen in Form von Dateien im Format L^AT_EX, PDF, HTML und XML. Zum anderen in dem proprietären Format *Qedeq*¹, welches eine elektronische Auswertung ermöglicht. All diese Dateien werden im *Internet* verfügbar sein und sich gegenseitig referenzieren.

1.2 Formale Form und andere Ansichten

Die mathematischen Sätze und ihre formalen Beweise sind in einer formalen Sprache gehalten, welche die Möglichkeit einer automatischen Kontrolle schafft. Aus dem zugehörigen Qedeq-Dateiformat können die bereits erwähnten Dokumente im Format L^AT_EX, PDF, HTML und XML generiert werden. Diese enthalten die formale Sprache nicht mehr, sondern sind in einer dem üblichen mathematischen Gebrauch möglichst ähnlichen Form gehalten.

Diese Umwandlung bedeutet in der Regel auch einen Informationsverlust, wenn zum Beispiel verschiedene Operatoren des Qedeq-Formates in dasselbe L^AT_EX-Symbol umgewandelt werden. Der besseren Lesbarkeit wegen ist das gewünscht, in Zweifelsfällen über die exakte Bedeutung eines Formelabschnitts ist immer das zugehörige Qedeq-Format heranzuziehen.

¹Die Verwendung des Wortes *Qedeq* leitet sich aus der Abkürzung Q.E.D. ab. Um einen weltweit möglichst einmaligen Zusatz zu haben, wurde die Abkürzung symmetrisch ergänzt.

1.3 Textbuch und Referenzen

Die Dokumente entsprechen im Aufbau und Inhalt einem normalen mathematischen Textbuch. In Kapitel und Abschnitte gegliedert, wird das mathematische Wissen schrittweise entwickelt. Außerdem ist es mithilfe des Qedeq-Formats möglich, alle für einen Abschnitt notwendigen Voraussetzungen² zu ermitteln. Ausgehend von einem mathematischen Satz, können so also alle Wege in das mathematische Fundament verfolgt werden. Dies kann am bequemsten in dem *Qedeq-Viewer* erfolgen. Dieser kann nicht nur analog zum HTML-Format alle Querverweise verfolgen, sondern verfügt über verschiedene Analysemöglichkeiten, die z. B. die Abhängigkeit von Sätzen aufzeigen, Ableitungsregeln begründen oder Definitionen auflösen können.

1.4 Formale Korrektheit

Die Kontrolle, ob die formalen Beweise auf formal logischer Ebene korrekt durchgeführt sind, kann mit Hilfe eines *Proofcheckers* durchgeführt werden. Dazu wird ebenfalls das Qedeq-Format verwendet. Dabei müssen die Beweise in einer hinreichenden Detaillierung vorliegen, so dass eine Beweiszeile jeweils aus den vorherigen nach Anwendung logischer Regeln bzw. syntaktischer Operationen³ folgt. Die Mächtigkeit dieser Regeln bestimmt die Lesbarkeit solcher formalen Beweise. Eine weiteres Strukturierungsmittel bei der Erstellung eines Qedeq-Dokuments ist die Verwendung von einfachen Hilfssätzen. Bei offensichtlicher Klarheit⁴ kann die Leserin dann die Beweise der Hilfssätze überspringen und nur den Beweis des Hauptsatzes unter Verwendung der Hilfssätze nachvollziehen.

1.5 Vernetzung

Mathematik ist kein totes Wissensgebäude sondern erfährt ständig vielfältige Umbauten und Neuorganisationen. Dabei ergeben sich unterschiedliche Standpunkte und immer neue Querverbindungen. Um dieser dezentralen und dynamischen Struktur gerecht zu werden, arbeitet **Hilbert II** in einer dezentralen Client/Server-Architektur. Als Server kann jeder HTTP- oder FTP-Server dienen, nur auf dem Client läuft das eigentliche Programm.⁵ Ausgehend von einer bestimmten Qedeq-Datei werden dann alle vorausgesetzten weiteren Qedeq-Module von den jeweiligen Stellen im Internet angefordert. Bei ausschließlichem Zugriff auf die Formate HTML und PDF ist der Client ein einfacher Web-Browser.⁶

1.6 Sprachen

Qedeq-Module können Texte verschiedener Sprachen enthalten, Standardsprache ist jedoch Englisch. Bei der Erzeugung von Ansichten kann eine Sprache

²Dabei handelt es sich um andere Abschnitte, gegebenenfalls auch aus anderen Qedeq-Dokumenten. Falls für einen Satz kein formaler Beweis vorhanden ist, kann diese Abhängigkeit nicht vollständig ermittelt werden.

³Eine syntaktische Operation ist z. B. die Verwendung von Abkürzungen bzw. deren Eliminierung.

⁴Das ist natürlich immer eine sehr subjektive Einschätzung.

⁵In späteren Ausbaustufen kann die Architektur zur besseren Verteilung und Skalierung verfeinert werden.

⁶Eventuell lässt sich die Programmsuite von **Hilbert II** auch in ein *Plugin* für einen Webbrowser einbinden.

ausgewählt werden, für die dieses Qedeq-Modul Texte enthält. So können beispielsweise aus einem Qedeq-Modul PDF-Dateien in englischer und in deutscher Sprache erzeugt werden.⁷

1.7 Detailgrad

Innerhalb eines Qedeq-Moduls können auch nichtformale Beweise unterschiedlichen Detailgrades vorhanden sein. So kann beispielsweise in der obersten Stufe ein Beweis oder sogar ein Hilfssatz fehlen („trivialerweise gilt“), in der darunterliegenden Stufe könnte „aus den Definitionen ergibt sich sofort“ stehen und in der nächsten Stufe könnte der formal korrekte Beweis angegeben sein. Bei der Erzeugung von Ansichten ist es dann möglich, die Detailtiefe anzugeben.

1.8 Metaregeln und Versionen

Die Abhängigkeiten von Qedeq-Modulen bilden eine komplexe Struktur.⁸ Wichtigste Quelle ist dabei ein Qedeq-Modul, welches die Axiome der Mengenlehre enthält. Daran anschliessend werden aus den Axiomen Folgerungen gezogen, und durch die Definition neuer Strukturen ergeben sich neue Objekte, deren Eigenschaften analysiert werden. Die Sicht kompliziert sich durch die Benutzung von *Metaregeln*. Metaregeln sind Spracherweiterungen, die eine einfachere Darstellung oder Argumentation ermöglichen. Im Prinzip lassen sie sich jedoch immer durch die ursprüngliche Sprachsyntax ersetzen. Die Benutzung von Metaregeln setzt eine Erweiterung des Proofcheckers und aller anderen Programmteile von **Hilbert II** voraus. Daher ist jedes Qedeq-Modul mit einer Regel-Versionsnummer versehen. Jede Version von **Hilbert II** kann nur mit Modulen bis zu einer bestimmten Regelversionsnummer umgehen. Deshalb kann es vorkommen, dass ein Qedeq-Modul von einer älteren **Hilbert II**-Version nicht überprüft werden kann. Es empfiehlt sich daher, ein Qedeq-Modul auch in älteren Regelversionen vorzuhalten. Dabei ist zu beachten, dass benötigte weitere Qedeq-Module in derselben oder kleineren Regelversionen vorhanden sein müssen, so dass der Proofchecker die Korrektheit komplett prüfen kann.

Zur Ableitung der logischen Regeln, die Vorbedingung für die mathematische Argumentation sind, gibt es zwei weitere Basis-Qedeq-Module. Diese enthalten die Axiome und Definitionen der Aussagen- und Pädikatenlogik. Daraus werden dann neue Sätze entwickelt, die zu neuen logischen Metaregeln führen. Diese neuen Metaregeln komplettieren dann die logischen Regeln und bilden die Basis, mit der dann die mathematische Theorie entfaltet werden kann. So wird bei der Entwicklung der Mengenlehre gleich eine höhere Regelversion verwendet, damit sich die logische Argumentation komfortabel gestaltet und die Mengenlehre den Schwerpunkt bildet kann.

1.9 Erzeugung und Bearbeitung

Die Erzeugung und Bearbeitung von Qedeq-Modulen ist – analog zur Arbeit mit L^AT_EX-Texten – mit einem einfachen Texteditor möglich. Bei Benutzung des XML-Formats kann ein XML-Editor zum Einsatz kommen. Ein spezieller komfortabler Editor für Qedeq-Module befindet sich in Planung.

⁷Das Qedeq-Modul muss dazu natürlich englische und deutsche Texte enthalten.

⁸Die Abhängigkeiten bilden einen schwach zusammenhängenden, zykelfreien Diagraphen. Die Axiome bilden die Quellen und die Senken sind die Endpunkte der mathematischen Theorie. Eine topologische Sortierung der Knoten, d. h. der Qedeq-Module, liefert eine Lesereihenfolge.

Kapitel 2

Mathematische Ausgangspunkte

Mathematik ist das Ziel, aber auch der Ausgangspunkt für **Hilbert II**. In diesem Kapitel werden die mathematischen Grundlagen formuliert, auf denen das Projekt fußt. Zunächst werden die logischen Gesetzmäßigkeiten für die Prädikatenlogik erster Stufe definiert. Anschließend werden die Axiome der Mengenlehre notiert. Aus diesen Grundbausteinen kann dann die restliche Mathematik entwickelt werden.

Die mathematischen Grundlagen, insbesondere die der Mengenlehre, werden in einem gesonderten Dokument gepflegt: http://www.qedeq.org/0_01_05/mengenlehre_1.pdf. Dort ist ein weiterentwickelter Stand zu finden, der an einigen Stellen auch grundsätzlich abweichen kann. In diesem Sinne handelt es sich bei dem Folgendem nur um eine vorläufige und grobe Skizze der mathematischen Ausgangspunkte für dieses Projekt.

2.1 Logische Axiome

Die Grundlagen der bei **Hilbert II** verwendeten Logik werden hier zusammengestellt. Die folgende Kalkülsprache und ihre Axiome basieren auf den Formulierungen von *D. Hilbert*, *W. Ackermann*, *P. Bernays* und *P. S. Novikov*. Aus den hier angegebenen logischen Axiomen und den elementaren Schlussregeln können weitere Gesetzmäßigkeiten abgeleitet werden. Erst diese neuen Metaregeln führen zu einer komfortablen logischen Argumentation. Siehe auch unter http://www.qedeq.org/mathematics_de.html.

Als Symbole kommen die logischen Symbole $L = \{ \neg, \vee, \wedge, \leftrightarrow, \rightarrow, \forall, \exists \}$, die Prädikatenkonstanten $C = \{c_i^k \mid i, k \in \omega\}$, die Funktionsvariablen¹ $F = \{f_i^k \mid i, k \in \omega \wedge k > 0\}$, die Funktionskonstanten² $H = \{h_i^k \mid i, k \in \omega\}$, die Subjektvariablen $V = \{v_i \mid i \in \omega\}$, sowie die Prädikatenvariablen $P = \{p_i^k \mid i, k \in \omega\}$ vor.³ Unter der *Stellenzahl* eines Operators wird der obere Index verstanden. Die Menge der nullstelligen Prädikatenvariablen wird auch als

¹Funktionsvariablen dienen der einfacheren Notation und werden beispielsweise zur Formulierung eines identitätslogischen Satzes benötigt: $x = y \rightarrow f(x) = f(y)$. Außerdem bereitet ihre Einführung die spätere Syntaxerweiterung zur Anwendung von funktionalen Klassen vor.

²Funktionskonstanten dienen ebenfalls der Bequemlichkeit und werden später für direkt definierte Klassenfunktionen verwendet. So zum Beispiel zur Potenzklassenbildung, zur Vereinigungsklassenbildung und für die Nachfolgerfunktion. All diese Funktionskonstanten können auch als Abkürzungen verstanden werden.

³Unter ω werden die natürlichen Zahlen, die Null eingeschlossen, verstanden. Alle bei den Mengenbildungen beteiligten Symbole werden als paarweise verschieden vorausgesetzt. Das bedeutet z. B.: $f_i^k = f_{i'}^{k'} \rightarrow (k = k' \wedge i = i')$ und $h_i^k \neq v_j$.

Menge der Aussagenvariablen bezeichnet: $A := \{p_i^0 \mid i \in \omega\}$. Für die Subjektvariablen werden abkürzend auch bestimmte Kleinbuchstaben geschrieben. Die Kleinbuchstaben stehen für verschiedene Subjektvariablen: $v_1 = 'u'$, $v_2 = 'v'$, $v_3 = 'w'$, $v_4 = 'x'$, $v_5 = 'y'$, $v_6 = 'z'$. Weiter werden als Abkürzungen verwendet: für die Prädikatenvariablen $p_1^n = '\phi'$ und $p_2^n = '\psi'$, wobei die jeweilige Stellenanzahl n aus der Anzahl der nachfolgenden Parameter ermittelt wird, für die Aussagenvariablen $a_1 = 'A'$, $a_2 = 'B'$ und $a_3 = 'C'$. Als Abkürzungen für Funktionsvariablen wird festgelegt $f_1^n = 'f'$ und $f_2^n = 'g'$, wobei wiederum die jeweilige Stellenanzahl n aus der Anzahl der nachfolgenden Parameter ermittelt wird. Bei allen aussagenlogischen zweistelligen Operatoren wird der leichteren Lesbarkeit wegen die Infixschreibweise benutzt, dabei werden die Symbole '(' und ')' verwandt. D. h. für den Operator \wedge mit den Argumenten A und B wird $(A \wedge B)$ geschrieben. Es gelten die üblichen Operatorprioritäten und die dazugehörigen Klammerregeln. Insbesondere die äußeren Klammern werden in der Regel weggelassen.

Nachfolgend werden die Operatoren mit absteigender Priorität aufgelistet.

$$\begin{array}{c} \neg, \forall, \exists \\ \wedge \\ \vee \\ \rightarrow, \leftrightarrow \end{array}$$

Der Begriff *Term* wird im Folgenden rekursiv definiert:

1. Jede Subjektvariable ist ein Term.
2. Seien $i, k \in \omega$ und t_1, \dots, t_k Terme. Dann ist auch $h_i^k(t_1, \dots, t_k)$ und falls $k > 0$, so auch $f_i^k(t_1, \dots, t_k)$ ein Term.

Alle nullstelligen Funktionskonstanten $\{h_i^0 \mid i \in \omega\}$ sind demzufolge Terme, sie werden auch *Individuenkonstanten*.⁴

Die Begriffe *Formel*, *freie* und *gebundene* Subjektvariable werden rekursiv wie folgt definiert:

1. Jede Aussagenvariable ist eine Formel, solche Formeln enthalten keine freien oder gebundenen Subjektvariablen.
2. Ist p^k eine k -stellige Prädikatenvariable und c^k eine k -stellige Prädikatenkonstante und sind t_1, t_2, \dots, t_k Terme, so sind $p^k(t_1, t_2, \dots, t_k)$ und $c^k(t_1, t_2, \dots, t_k)$ Formeln. Dabei gelten alle in t_1, t_2, \dots, t_k vorkommenden Subjektvariablen als freie Subjektvariablen, gebundene Subjektvariablen kommen nicht vor.⁵
3. Falls die Formel α die freie Subjektvariable x_1 enthält (+++ ist das tatsächlich erforderlich, dass x_1 in α vorkommt?), dann sind auch $\forall x_1 \alpha$ und $\exists x_1 \alpha$ Formeln⁶, und bis auf x_1 bleiben alle freien Subjektvariablen von α auch frei, und zu den gebundenen Subjektvariablen von α kommt x_1 hinzu.
4. Es seien α, β Formeln, in denen keine Subjektvariablen vorkommen, die in einer Formel gebunden und in der anderen frei sind. Dann sind auch $\neg \alpha$, $(\alpha \wedge \beta)$, $(\alpha \vee \beta)$, $(\alpha \rightarrow \beta)$, $(\alpha \leftrightarrow \beta)$ Formeln. Subjektvariablen, welche in α, β frei (bzw. gebunden) vorkommen, bleiben frei (bzw. gebunden).

⁴Analog dazu könnten Subjektvariablen auch als nullstellige Funktionsvariablen definiert werden. Da die Subjektvariablen jedoch eine hervorgehobene Rolle spielen, werden sie auch gesondert bezeichnet.

⁵Dieser zweite Punkt umfasst den ersten, welcher nur der Anschaulichkeit wegen extra aufgeführt ist.

⁶Dabei wird \forall als *Allquantor* und \exists als *Existenzquantor* bezeichnet

Die folgenden Axiome werden an den Anfang gestellt.

Axiom 1 (Oder-Kürzung).

$$(A \vee A) \rightarrow A$$

Axiom 2 (Oder-Verdünnung).

$$A \rightarrow (A \vee B)$$

Axiom 3 (Oder-Vertauschung).

$$(A \vee B) \rightarrow (B \vee A)$$

Axiom 4 (Oder-Vorsehung).

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

Axiom 5.

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

Axiom 6.

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

Die Konjunktion, die Implikation und die Äquivalenz werden als Abkürzungen definiert.⁷

Definition 1 (Konjunktion).

$$\alpha \wedge \beta : \Leftrightarrow \neg(\neg\alpha \vee \neg\beta)$$

Definition 2 (Implikation).

$$\alpha \rightarrow \beta : \Leftrightarrow \neg\alpha \vee \beta$$

Definition 3 (Äquivalenz).

$$\alpha \leftrightarrow \beta : \Leftrightarrow (\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$$

Regel 1 (Abtrennung, Modus Ponens). Wenn α und $\alpha \rightarrow \beta$ wahre Formeln sind, dann ist auch β eine wahre Formel.

Regel 2 (Einsetzung für Prädikatenvariablen). Es sei α eine Formel, die die n -stellige Prädikatenvariable p enthält und $\beta(x_1, \dots, x_n)$ eine beliebige Formel mit den freien Variablen x_1, \dots, x_n , welche nicht in α vorkommen.⁸ Wenn die folgenden Bedingungen erfüllt sind, dann kann durch die Ersetzung jedes Vorkommens von $p(t_1, \dots, t_n)$ mit jeweils passenden Termen t_1, \dots, t_n in α durch $\beta(t_1, \dots, t_n)$ eine weitere wahre Formel gewonnen werden.

- α ist eine im Kalkül wahre Formel
- die freien Variablen von α sind disjunkt zu den gebundenen Variablen von $\beta(x_1, \dots, x_n)$ und die gebundenen Variablen von α disjunkt zu den freien Variablen von $\beta(x_1, \dots, x_n)$

⁷Eigentlich werden die Abkürzungssymbole $\wedge, \rightarrow, \leftrightarrow$ erst an dieser Stelle definiert und erweitern die Sprachsyntax. Aus Bequemlichkeitsgründen wurden diese Symbole bereits als logische Symbole angegeben. Bei den bisher angegebenen Axiomen muss also strenggenommen die Implikation durch ihre Definition ersetzt werden.

⁸In der Formel $\beta(x_1, \dots, x_n)$ können außer den n Subjektvariablen x_1, \dots, x_n noch weitere Variablen frei vorkommen.

- Liegt das zu ersetzenende $p(t_1, \dots, t_n)$ in α im Wirkungsbereich eines Quantors, so kommt die zugehörige Subjektvariable in $\beta(x_1, \dots, x_n)$ nicht vor.⁹

Regel 3 (Einsetzung für Funktionsvariablen). Es sei α eine Formel, die die n -stellige Funktionsvariable σ enthält und $\tau(x_1, \dots, x_n)$ eine beliebiger Term mit den Subjektvariablen x_1, \dots, x_n , welche nicht in α vorkommen.¹⁰ Wenn die folgenden Bedingungen erfüllt sind, dann kann durch die Ersetzung jedes Vorkommens von $\sigma(t_1, \dots, t_n)$ mit jeweils passenden Termen t_1, \dots, t_n in α durch $\tau(t_1, \dots, t_n)$ eine weitere wahre Formel gewonnen werden.

- α ist eine im Kalkül wahre Formel
- die gebundenen Variablen von α sind disjunkt zu den Subjektvariablen von $\tau(x_1, \dots, x_n)$
- Liegt das zu ersetzenende $\sigma(t_1, \dots, t_n)$ in α im Wirkungsbereich eines Quantors, so kommt die zugehörige Subjektvariable in $\beta(x_1, \dots, x_n)$ nicht vor, d. h. nach dem Einsetzen werden – außer den für x_1, \dots, x_n eingesetzten – keine weiteren Subjektvariablen gebunden.

Regel 4 (Ersetzung für freie Subjektvariablen). Ausgehend von einer wahren Formel kann jede freie Subjektvariable durch einen Term ersetzt werden, der keine in der Formel bereits gebundenen Subjektvariablen enthält. Die Ersetzung muss durchgängig in der gesamten Formel erfolgen.

Regel 5 (Umbenennung für gebundene Subjektvariablen). Aus jeder im Kalkül bereits gewonnenen Formel können weitere Formeln abgeleitet werden: Jede gebundene Subjektvariable kann in eine andere, nicht bereits frei vorkommende, Subjektvariable umbenannt werden. Die Umbenennung braucht nur im Wirkungsbereich eines bestimmten Quantors zu erfolgen.

Regel 6 (Generalisierung). Wenn $\alpha \rightarrow \beta(x_1)$ eine wahre Formel ist und α die Subjektvariable x_1 nicht enthält, dann ist auch $\alpha \rightarrow (\forall x_1 (\beta(x_1)))$ eine wahre Formel.

Regel 7 (Partikularisierung). Wenn $\alpha(x_1) \rightarrow \beta$ eine wahre Formel ist und β die Subjektvariable x_1 nicht enthält, dann ist auch $(\exists x_1 \alpha(x_1)) \rightarrow \beta$ eine wahre Formel.

Die Auflösung und der Einsatz von Abkürzungen ist auch mit der Anwendung von Regeln verbunden. In vielen Texten zur mathematischen Logik werden diese Regeln nicht explizit formuliert, auch dieser Text geht darauf nicht weiter ein. In dem exakten Qedeq-Format gibt es jedoch entsprechende Regeln zur Verwendung von Abkürzungen.

2.2 Identität

Es wird eine zweistellige Funktionskonstante festgelegt, welche in der Interpretation die Identität von Subjekten ausdrücken soll.

Definition 4 (Gleichheit).

$$x = y \Leftrightarrow c_1^2(x, y)$$

Dazu werden zwei weitere Axiome, auch *Gleichheitsaxiome* genannt, formuliert.

⁹D. h. nach dem Einsetzen werden – außer den in den für x_1, \dots, x_n eingesetzten Termen vorkommenden – keine weiteren Subjektvariablen gebunden.

¹⁰In dem Term $\tau(x_1, \dots, x_n)$ können außer den n Subjektvariablen x_1, \dots, x_n noch weitere Variablen vorkommen.

Axiom 7 (Reflexivität der Gleichheit).

$$x = x$$

Axiom 8 (Leibnizsche Ersetzbarkeit).

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

2.3 Eingeschränkte Quantoren

Bei der folgenden Definition muss die für $\psi(x)$ eingesetzte Formel „erkennen lassen“, über welche Subjektvariable quantifiziert wird. Das ist in der Regel darüber zu entscheiden, welche freie Subjektvariable als erstes in der Formel vorkommt.¹¹ In der exakten Syntax des Qedeq-Formats¹² ist die Subjektvariable immer angegeben.

Definition 5 (Eingeschränkter Allquantor).

$$\forall \psi(x) (\phi(x)) \Leftrightarrow \forall x (\psi(x) \rightarrow \phi(x))$$

Dazu passt die folgende Definition für den eingeschränkten Existenzquantor.¹³

Definition 6 (Eingeschränkter Existenzquantor).

$$\exists \psi(x) (\phi(x)) \Leftrightarrow \exists x (\psi(x) \wedge \phi(x))$$

Für die Existenz genau eines Individuums mit einer bestimmten Eigenschaft wird nun ein gesonderter Quantor eingeführt.

Definition 7 (Eingeschränkter Existenzquantor für genau ein Individuum).

$$\exists! \psi(x) (\phi(x)) \Leftrightarrow \exists \psi(x) (\phi(x) \wedge \forall \psi(y) (\phi(y) \rightarrow x = y))$$

Durch die Gültigkeit von $\exists! \psi(x)(\phi(x))$ kann daher eine Subjektkonstante definiert werden, wenn $\phi(x)$ und $\psi(x)$ durch Ausdrücke ersetzt werden, die außer x keine freien Variablen, keine Prädikatsvariablen und keine Funktionsvariablen mehr enthält.

2.4 Axiome der Mengenlehre

Nach der Entwicklung der *naiven Mengenlehre* (1895 - 1897) durch *G. Cantor* und der Entdeckung von Widersprüchen durch *B. Russell* (1902) gab es verschiedene Versuche die Mengenlehre zu axiomatisieren. Zum einen in der *Typentheorie*¹⁴ (1910) von *A. N. Whitehead* und *B. Russell* in ihrem epochalen Werk „Principia Mathematica“. Zum anderen in der Zermeloschen axiomatischen Mengenlehre (1908) mit der Fraenkischen Verbesserung¹⁵ (1921) und der skolemschen Präzisierung der mengentheoretischen Aussagen¹⁶ (1922). Diese sogenannte Zermelo-Fraenkelsche Mengenlehre (*ZF*) erfuhr durch *K. Gödel*

¹¹Beispielsweise ist in der folgenden Formel erkennbar, dass die zweite Quantifikation über die Subjektvariable m läuft: $\forall n \in \mathbb{N} \forall m \in n m < n$.

¹²Siehe unter B.

¹³Passend, da $\neg\forall \psi(x) (\phi(x)) \leftrightarrow \exists x \neg(\psi(x) \rightarrow \phi(x)) \leftrightarrow \exists x (\psi(x) \wedge \neg\phi(x)) \leftrightarrow \exists \psi(x) (\neg\phi(x))$.

¹⁴In der Typentheorie können nur Mengen einer bestimmten Stufe zu Mengen der nächsthöheren Stufe zusammengefaßt werden.

¹⁵Verwendung unendlich vieler Axiome, Aussonderungsschema.

¹⁶Das Aussonderungsschema wird durch das Ersetzungssaxiomschema ersetzt.

und *P. Bernays* eine Verbesserung, die häufig mit *ZFG* abgekürzt wird. Die Verwendung dieser für mengentheoretische Grundlagenforschung immer noch benutzten Theorie ist jedoch für die mathematische Praxis unbequem. Auch die Ergänzungen von *J. v. Neumann* und *P. Bernays* führten zu einer neuen Formulierung, welche mit *NBG* bezeichnet wird. Eine Verallgemeinerung von *NBG* wurde von *J. L. Kelley* (1955) vorgestellt. In der Tradition dieses auch *MK*¹⁷ genannten Systems steht auch das hier verwendete Axiomensystem der Mengenlehre.

Die Interpretationen der Subjektvariablen heißen *Klassen*.

Als neue zweistellige Funktionskonstante wird die Elementbeziehung \in vorausgesetzt.

Definition 8 (Elementbeziehung).

$$x \in y \Leftrightarrow c_2^2(x, y)$$

Definition 9 (Negation der Elementbeziehung).

$$x \notin y \Leftrightarrow \neg(x \in y)$$

Ist eine Klasse Element einer anderen Klasse, dann wird sie auch als *Menge* bezeichnet.

Definition 10 (Mengendefinition).

$$\mathfrak{M}(x) \Leftrightarrow \exists y (x \in y)$$

Zur Klassenbildung wird das Komprehensionsaxiom benötigt.

Axiom 9 (Komprehensionssaxiom).

$$\exists x \forall y (y \in x \leftrightarrow \mathfrak{M}(y) \wedge \phi(y))$$

Werden an $\phi(y)$ zusätzliche Anforderungen gestellt, so bilden die hier aufgelisteten Axiome ein *NBG*-System.¹⁸ In den herkömmlichen Formulierungen dieses Axioms wird als zusätzliche Einschränkung an $\phi(y)$ gefordert, dass in dieser Formel die Subjektvariable x nicht vorkommen darf. Diese Forderung ist aufgrund des Verbots der mehrfachen Quantifizierung mit derselben Subjektvariablen und der Regel 2 erfüllt.

Der Gleichheitsbegriff wird durch das Extentsionalitätsaxiom festgeschrieben.¹⁹ Die Gleichheit zweier Klassen soll dann gegeben sein, wenn sie dieselben Elemente besitzen.

Axiom 10 (Extensionalitätsaxiom).

$$\forall z (z \in x \leftrightarrow z \in y) \rightarrow x = y$$

Definition 11 (Unterklasse).

$$x \subset y \Leftrightarrow \forall z (z \in x \rightarrow z \in y)$$

¹⁷Diese Abkürzung ist nach *A. P. Morse* und *J. L. Kelley* benannt, das Axiomensystem wurde jedoch zuvor auch von anderen Mathematikern vorgestellt.

¹⁸Dazu darf in $\phi(y)$ nur über Mengen quantifiziert werden, d. h. alle in $\phi(y)$ vorkommenden Quantoren haben die Form $\forall \mathfrak{M}(z) (\psi(z))$ oder $\exists \mathfrak{M}(z) (\psi(z))$ mit entsprechenden z und $\psi(z)$. Falls $\phi(y)$ diese Anforderung erfüllt, wird auch von einer *prädiktiven* Formel gesprochen.

¹⁹Die Umfangsgleichheit kann auch als Definition des Gleichheitsprädikates verwendet werden. Das ist jedoch zum Nachweis der Leibnizschen Ersetzbarkeit nicht ausreichend. Dafür muss die Extensionalität durch das folgende Axiom gesichert werden ($x = y \wedge x \in z \rightarrow y \in z$). Damit genügt dann das definierte Gleichheitsprädikat allen Anforderungen an den in Abschnitt 2.2 vorgestellten Identitätsbegriff. Aus Gründen der formalen Vereinfachung wird hier jedoch von einem Prädikatenkalkül mit Identität ausgegangen.

Axiom 11 (Axiom der Potenzmenge).

$$\forall \mathfrak{M}(x) \exists \mathfrak{M}(y) \forall z (z \subset x \rightarrow z \in y)$$

Axiom 12 (Axiom der Paarmenge).

$$\forall \mathfrak{M}(x) \forall \mathfrak{M}(y) \exists \mathfrak{M}(z) (x \in z \wedge y \in z)$$

Axiom 13 (Axiom der Vereinigungsmenge).

$$\forall \mathfrak{M}(x) \exists \mathfrak{M}(y) \forall z (z \in x \rightarrow z \subset y)$$

Regel 8 (Klassenschreibweise). Der Ausdruck $\{x \mid \phi(x)\}$ bezeichnet einen Term, alle Bildungsregeln werden entsprechend erweitert. Ist $\alpha(\{x \mid \phi(x)\})$ eine mit diesem Term gebildete Formel, so wird diese Formel definiert durch $\alpha(y) \wedge \forall x (x \in y \leftrightarrow \mathfrak{M}(x) \wedge \phi(x))$ mit einer noch nicht in α vorkommenden Subjektvariablen y .²⁰ Auch in der abkürzenden Schreibweise gilt die Subjektvariable x als gebunden, die Subjektvariable y ist frei wählbar und wird in der Abkürzung nicht weiter beachtet. Bei der Ersetzung von ϕ durch eine andere Formel muss auch in $\{x \mid \phi(x)\}$ ersetzt werden.

Im Folgenden kann diese abkürzende Schreibweise in prädikativen Aussagen benutzt werden.

Satz 2.4.1. Die Gültigkeit der Ausgangsprädikate drückt sich für diesen neuen Term wie folgt aus:

$$y \in \{x \mid \phi(x)\} \Leftrightarrow \mathfrak{M}(y) \wedge \phi(y) \quad (2.1)$$

$$y = \{x \mid \phi(x)\} \Leftrightarrow \forall z (z \in y \leftrightarrow z \in \{x \mid \phi(x)\}) \quad (2.2)$$

$$\{x \mid \phi(x)\} = \{x \mid \psi(x)\} \Leftrightarrow \forall z (z \in \{x \mid \phi(x)\} \leftrightarrow z \in \{x \mid \psi(x)\}) \quad (2.3)$$

$$\quad \quad \quad \leftrightarrow z \in \{x \mid \psi(x)\} \quad (2.4)$$

$$\{x \mid \phi(x)\} \in \{x \mid \psi(x)\} \Leftrightarrow \forall u \forall v (u = \{x \mid \phi(x)\} \wedge v = \{x \mid \psi(x)\} \rightarrow u \in v) \quad (2.5)$$

$$\{x \mid \phi(x)\} \in y \Leftrightarrow \forall u (u = \{x \mid \phi(x)\}) \rightarrow u \in y \quad (2.6)$$

Definition 12 (Durchschnitt).

$$x \cap y := \{z \mid z \in x \wedge z \in y\}$$

Definition 13 (Leere Menge).

$$\emptyset := \{x \mid x \neq x\}$$

Axiom 14 (Regularitätsaxiom).

$$\forall x (x \neq \emptyset \rightarrow \exists y (y \in x \wedge y \cap x = \emptyset))$$

Definition 14 (Nachfolger).

$$\mathcal{N}(x) := \{y \mid y \in x \vee y = x\}$$

Axiom 15 (Unendlichkeitsaxiom).

$$\exists x (\emptyset \in x \wedge \forall y (y \in x \rightarrow \mathcal{N}(y) \in x))$$

Eine Klasse kann auch durch Angabe ihrer Elemente definiert werden. Insbesondere kann durch Angabe eines Elements die sogenannte *Einerklasse* festgelegt werden.

²⁰Auch die Schreibweise $\exists!y (\alpha(y) \wedge \forall x (x \in y \leftrightarrow \mathfrak{M}(x) \wedge \phi(x)))$ ist möglich.

Definition 15 (Einerklasse).

$$\{x\} := \{u \mid \mathfrak{M}(x) \rightarrow u = x\}$$

Da der Ausdruck $\{x\}$ für jediges x definiert ist, kann er auch für den Fall, dass x eine echte Klasse ist, gebildet werden. In diesem Fall erfüllen alle Mengen u die Bedingung $\mathfrak{M}(u) \wedge (\mathfrak{M}(x) \rightarrow u = x)$ und die Einerklasse ist mit der Allklasse identisch. Das führt zu einem technisch einfacheren Umgang mit der Einerklasse.²¹

Definition 16 (Klassenangabe durch Aufzählung).

$$\{x, y\} := \{x\} \cup \{y\} \quad (2.7)$$

$$\{x, y, z\} := \{x\} \cup \{y\} \cup \{z\} \quad (2.8)$$

$$(2.9)$$

Die Definition eines geordneten Paares $\langle x, y \rangle$ erfolgt nach *N. Wiener* (1914) bzw. *K. Kuratowski* (1921).

Definition 17 (Geordnetes Klassenpaar).

$$\langle x, y \rangle := \{\{x\}, \{x, y\}\}$$

Definition 18 (Relation).

$$\text{Rel}(x) : \Leftrightarrow \forall y \in x \exists \mathfrak{M}(u) \exists \mathfrak{M}(v) y = \langle u, v \rangle$$

Definition 19 (Definitionsbereich).

$$\text{Dmn}(x) := \{u \mid \mathfrak{M}(\text{()})u \wedge \exists \mathfrak{M}(v) \langle u, v \rangle \in x\}$$

Definition 20 (Wertebereich).

$$\text{Rng}(x) := \{v \mid \mathfrak{M}(\text{()})v \wedge \exists \mathfrak{M}(u) \langle u, v \rangle \in x\}$$

Definition 21 (Funktionale Klasse).

$$\text{Fkt}(x) : \Leftrightarrow \text{Rel}(x) \wedge \forall u \in \text{Dmn}(x) \exists !v \langle u, v \rangle \in x$$

Axiom 16 (Ersetzungssaxiom).

$$\text{Fkt}(x) \wedge \mathfrak{M}(\text{Dmn}(x)) \rightarrow \mathfrak{M}(\text{Rng}(x))$$

Axiom 17 (Auswahlaxiom).

$$\text{Rel}(x) \rightarrow \exists \text{Fkt}(y) y \subset x \wedge \text{Dmn}(x) = \text{Dmn}(y)$$

2.5 Literaturverzeichnis

- *A.N. Whitehead, B. Russell*, Principia Mathematica, Cambridge University Press, London 1910
- *P. Bernays*, Axiomatische Untersuchung des Aussagen-Kalkuls der „Principia Mathematica“, Math. Zeitschr. 25 (1926), 305-320
- *D. Hilbert, W. Ackermann*, Grundzüge der theoretischen Logik, Springer, Berlin 1928

²¹Andere Autoren wie z. B. auch K. Gödel, definieren $\{x\}$ durch $\{u \mid u = x\}$.

- *P.S. Novikov*, Grundzüge der mathematischen Logik, VEB Deutscher Verlag der Wissenschaften, Berlin 1973
- *J. D. Monk*, Introduction to Set Theory, McGraw-Hill, New York 1969
- *J. Schmidt*, Mengenlehre I, Bibliographisches Institut, Mannheim 1966
- *U. Friedrichsdorf, A. Prestel*, Mengenlehre für den Mathematiker, Vieweg, Braunschweig 1985
- *G. Takeuti, W. M. Zaring*, Introduction to Axiomatic Set Theory, McGraw-Hill, New York 1971
- *W. Kerby*, Vorlesung „Axiomatische Mengenlehre“, gehalten an der Universität Hamburg, Sommersemester 1992
- *V. Günther*, Vorlesung „Mathematik und Logik“, gehalten an der Universität Hamburg, Wintersemester 1994/1995

Kapitel 3

Technischer Rahmen

Dieses Kapitel geht auf technische Details ein, die für die Realisierung von Bedeutung sind. Das Projekt **Hilbert II** ist ein internationales *Open-Source-Projekt*. Es ist bei Sourceforge unter <http://sourceforge.net/projects/pmii> zu finden. Der zugehörige Programm-Code steht unter der GNU General Public License, die zugehörigen Dokumente unterliegen der GNU Free Documentation License. Die Lizenzbedingungen für die GFDL sind im Anhang A angegeben.

3.1 Kernel

Das mathematische Wissen ist bei **Hilbert II** in Qedeq-Modulen *dezentral* organisiert. Diese Qedeq-Module können irgendwo im Internet liegen und über die TCP/IP-Protokolle `http` und `ftp` abgerufen werden. Durch **Hilbert II** wird ein Kernel bereitgestellt, der verschiedene Funktionen für diese Qedeq-Module anbietet. Zu dessen Basisaufgabe gehört das Einlesen und Parsen von Qedeq-Modulen. Nach dem erfolgreichen Lesen stehen verschiedene Methoden, auf das Modul strukturiert zuzugreifen, zur Verfügung.

Die wichtigste Funktionalität wird durch den *Proofchecker* realisiert. Er kann ein Qedeq-Modul auf Korrektheit überprüfen. Dazu werden alle Beweisschritte einzeln durchgegangen um festzustellen, ob eine Beweiszeile logisch aus den vorherigen folgt. Bei der Verwendung von Ergebnissen anderer Qedeq-Module, z. B. dort bereits bewiesenen Sätzen, kontrolliert der Proofchecker die Korrektheit dieser Qedeq-Module ebenfalls. Dazu verwaltet der Kernel eine Übersicht über bereits überprüfte Qedeq-Module, es können verschiedenste Überprüfungseinstellungen vorgenommen werden. Qedeq-Module können auch lokal zwischengespeichert werden, so dass nur bei geänderten Qedeq-Modulen eine Aktualisierung erfolgen muss.

Auch die Erzeugung von Qedeq-Modulen, die eine kleinere Regelversionsnummer benötigen, ist eine Funktionalität, die vom Kernel angeboten wird. Diese Funktionalität wird von dem *Regeltransformer* zur Verfügung gestellt.

Die Erzeugung von HTML-, XML- und L^AT_EX- und PDF-Dokumenten wird von den verschiedenen *Konvertern* des Kernels übernommen.

3.2 Organisation von Anwendungen

Anwendungen können auf dem Kernel aufsetzen, um eine komfortable Benutzeroberfläche zur Verfügung zu stellen. Der *Qedeq-Viewer* ermöglicht das Betrachten von Qedeq-Modulen und die Verfolgung von Querverweisen. Dazu werden

auch die L^AT_EX-Konstrukte dargestellt. Der Überprüfungsstand eines Moduls kann angezeigt, die Sprache und Detailtiefe eingestellt werden. Auch einfache Abhängigkeitsanalysen und Umwandlungen können innerhalb des Qedeq-Viewers ausgeführt werden. Zur Realisierung wird auf verschiedene Kernelfunktionen zurückgegriffen.

3.3 Qedeq-Format

Qedeq-Module werden in einem bestimmten Format im Dateisystem abgelegt. Ihr Dateiname setzt sich zusammen aus dem Qedeq-Modulnamen gefolgt von einem Unterstrich und der benötigten Regelversionsnummer und wird abgeschlossen mit der Zeichenfolge .Qedeq. Eine Qedeq-Datei besteht aus ASCII-Zeichen und hat die Struktur einer Listenschreibweise in Funktionssyntax: <argument> ::= (<funktionsname> '(' <argument> {',', <argument>} ')') | <number> | <text>;.¹ So wird die Formel $A \wedge B \rightarrow C$ beispielsweise ausgedrückt durch die Zeichenfolge IMPL(AND(PROP(1), PROP(2)), PROP(3)).

Eine Beschreibung dieses Formats ist im Anhang in dem Abschnitt B zu finden. Eine aktuelle und maschinenlesbare Formatbeschreibung ist unter http://www.qedeq.org/0_01_05/qedeq_de.csv zu finden.

Nun folgt ein Beispiel für ein Qedeq-Modul.²

```
MODULE (
  HEADER (
    SPEC (
      NAME ("predaxiom"), VERSION ("1.00.00"),
      LOCATIONS (LOCATION ("http://www.qedeq.org/0_01_05/"))
    ),
    TITLE (
      LATEX("en", "Axioms of predicate calculus"),
      LATEX("de", "Axiome der Pr{\\"a}dikatenlogik")
    ),
    ABSTRACT (
      LATEX ("en", "This example document contains the axioms
and rule declarations for the first order predicate
calculus. Together with the axioms and rules of the
propositional calculus (\QLINK(""propaxiom"")) all
theorems of predicate calculus could be deduced."),
      LATEX ("de", "Dieses Beispieldokument enth{\\"a}lt die
Axiome und die Deklarationen f{\\"u}r die Regeln der
Pr{\\"a}dikatenlogik erster Stufe. Zusammen mit den
Axiomen und Regeln der Aussagenlogik
(\QLINK(""propaxiom"")) k{\\"o}nnen damit alle S{\\"a}tze
der Pr{\\"a}dikatenlogik erster Stufe hergeleitet
werden.")
    ),
    EMAIL ("mime@qedeq.org"),
    AUTHORS (
      AUTHOR (LATEX("Michael Meyling"), EMAIL ("michael@meyling.com"))
    )
  ),
  USED BY (
```

¹Diese Syntaxangabe erfolgte in sogenannter *BNF-Notation*. Siehe auch im Anhang B.

²Die diesem Qedeq-Modul zugrundeliegende Syntax ist noch in Arbeit.

```

SPEC (
  NAME ("predtheo2"), VERSION ("1.00.00"),
  LOCATIONS (LOCATION (".")))
),
SPEC (
  NAME ("predtheo2"), VERSION ("1.02.00"),
  LOCATIONS (LOCATION (".")))
)
),
PARAGRAPHS (
PARAGRAPH (
  LABEL ("axiom5"),
  LATEX( "en", "Axiom of Specialization"),
  LATEX( "de", "Axiom der Spezialisierung"),
  AXIOM (
IMPL (
  FORALL(VAR(1), PREDVAR(1, L(VAR(1)))),
  PREDVAR(1, L(VAR(2))))
)
)
),
PARAGRAPH (
  LABEL ("axiom6"),
  LATEX( "en", "Axiom of Generalization"),
  LATEX( "de", "Axiom der Generalisierung"),
  AXIOM (
IMPL (
  PREDVAR(1, L(VAR(2))),
  EXISTS(VAR(1), PREDVAR(1, L(VAR(1))))
)
)
),
PARAGRAPH (
  LABEL ("predrule1"),
  DECLARERULE (
"RENAMEBOUNDVARIABLE",
"Rename Bound Variable"
),
  LATEX("en", "Here the explanation is still missing \ldots"),
  LATEX("de", "Hier fehlt noch die Erkl\{\"a}rung \ldots")
),
PARAGRAPH (
  LABEL ("predrule2"),
  DECLARERULE (
"RENAMEFREEVARIABLE",
"Rename Free Variable"
),
  LATEX("en", "Here the explanation is still missing \ldots"),
  LATEX("de", "Hier fehlt noch die Erkl\{\"a}rung \ldots")
),
PARAGRAPH (
  LABEL ("predrule3"),
  DECLARERULE (
"REPLACEPREDICATE",
"Particularize"
),
  LATEX("en", "Here the explanation is still missing \ldots"),

```

```

LATEX("de", "Hier fehlt noch die Erkl{\"a}rung \ldots")
),
PARAGRAPH (
  LABEL ("predrule4"),
  DECLARERULE (
  "PARTICULARIZATION",
  "Particularize"
  ),
  LATEX("en", "Here the explanation is still missing \ldots"),
  LATEX("de", "Hier fehlt noch die Erkl{\"a}rung \ldots")
),
PARAGRAPH (
  LABEL ("predrule5"),
  DECLARERULE (
  "GENERALIZATION",
  "Generalize"
  ),
  LATEX("en", "Here the explanation is still missing \ldots"),
  LATEX("de", "Hier fehlt noch die Erkl{\"a}rung \ldots")
)
)
)
)
)

```

3.4 L^AT_EX

In den Texten der Qedeq-Module kommt eine Untergruppe von L^AT_EX-Befehlen zum Einsatz. Damit ist die Formatierung von mathematischen Texten in einfacher Form möglich. Jeder Mathematikerin ist diese Syntax in der Regel geläufig. Da die Qedeq-Module von **Hilbert II** in verschiedenen Ansichten dargestellt werden können, wird – zumindest zunächst – nur ein kleiner Teil der Formatierungsmerkmale von L^AT_EX genutzt.³ Diese Knappheit bietet auch den Vorteil einer stärkeren Normierung des Erscheinungsbildes von Qedeq-Ansichten, was insgesamt zu einer besseren Lesbarkeit führen kann.

Welche Formatierungsmerkmale konkret unterstützt werden, wird erst im weiteren Verlauf des Projekts festgelegt.

3.5 XML

Aus dem Qedeq-Format lässt sich direkt eine XML-DTD ableiten. Zusätzlich bietet es sich an, die L^AT_EX-Anteile in *MathML* umzuwandeln. Dies hätte den Vorteil ein reines XML-Dokument zu erhalten und die vielfältigen Möglichkeiten von XML zu nutzen. Bei einer Neudefinition von Operatoren sollte eine neue DTD erzeugt werden, die dann auch von abhängigen Qedeq-XML-Dateien genutzt werden kann.

3.6 Formaterweiterungen

Die Definition neuer Prädikate und ihrer L^AT_EX-Darstellung bereitet keine Probleme. Wesentliches Element von **Hilbert II** ist auch die Erweiterung der Sprache durch neue Operatoren und Metaregeln. Diese Erweiterungen können auch

³Insbesondere die HTML-Ansicht limitiert die Möglichkeiten. Auch die Programmierung des Qedeq-Viewers wird durch Unterstützung von komplexeren L^AT_EX-Konstrukten erschwert.

den Programmcode betreffen⁴ und dann einen erhöhten Herstellungsaufwand erfordern. Die einfache Definition von neuen Prädikaten oder Funktionen, die dann in bestimmter L^AT_EX-Form ausgegeben werden, ist dagegen ohne Programmänderung jederzeit möglich.

Durch die dezentrale Organisation von Qedeq-Modulen ergibt sich das Problem der mehrfachen Definition eines Operators. Denn in zwei verschiedenen, von einander unabhängigen Qedeq-Modulen kann derselbe Operatorname verwendet werden, um unterschiedliche Operationen zu definieren. Damit kann ein weiteres Modul nicht beide anderen Module importieren, weil die Bedeutung dieses einen Operatornamens nicht eindeutig ist. Es sollte daher möglich sein, einen Operatornamen in späteren Abschnitten zu ändern. Dazu muss die Gültigkeit des Operatornamens stärker kontrolliert werden.

Aus Sicht eines Qedeq-Moduls lässt sich die Abhängigkeit von anderen Qedeq-Modulen stets in serieller Form darstellen.⁵ Ein Modul nach dem anderen wird abgeleitet, bis zum aktuell betrachteten Qedeq-Modul. Dabei kann beispielsweise nur ein Axiomensystem der Mengenlehre zum Einsatz kommen oder die natürlichen Zahlen nur auf eine Art definiert werden. Viele Anwendungen benötigen jedoch nur bestimmte Eigenschaften dieser Strukturen. So könnte es sinnvoll sein, eine Schnittstelle zu schaffen, in der nur diese Eigenschaften beschrieben sind, also beispielsweise die Peano-Axiome der natürlichen Zahlen aufgeführt sind. Diese Schnittstellen können dann zum einen als Axiome aufgefasst werden, zum anderen können sie aber aus verschiedenen anderen Qedeq-Modulen als Sätze abgeleitet werden. Dies wird der komplexen Abhängigkeitsstruktur am besten gerecht und ermöglicht es auch frühzeitig neue mathematische Theorien zu dokumentieren, indem zunächst neue Axiome eingeführt werden. Später können diese dann – eventuell auf verschiedene Weisen – als Sätze erhalten werden.

Wesentliche Formaterweiterungen sind auch die Einführung neuer Ableitungsregeln, die eine bequemere Herleitung von Beweiszeilen ermöglichen. Diese Regeln werden sicherlich jeweils auf bestimmte Objekte ausgerichtet sein, so wie zum Beispiel eine neue Metaregel zur vollständigen Induktion sich auf Funktionen von natürlichen Zahlen beziehen wird. Diese Metaregel lässt sich in jedem konkreten Anwendungsfall durch die Anwendung bisheriger Regeln ersetzen. In **Hilbert II** besteht das Ziel, diese Ersetzung auch automatisch durchzuführen. Wie bereits erwähnt, sind neue Ableitungsregeln stets mit Programmänderungen verbunden.

3.7 Programmiersprache

Die Umsetzung dieses Grobkonzeptes ist nicht an eine bestimmte Programmiersprache gebunden. Die wesentliche Schnittstelle ist das Qedeq-Format.

Die Referenzimplementierung von **Hilbert II** erfolgt in Java. Java ist eine moderne objektorientierte Programmiersprache, die über reichhaltige Bibliotheken verfügt. Darin erzeugte Programme sind ohne Anpassung auf vielen Betriebssystemen ablauffähig.

Weitere Einzelheiten zur Programmentwicklung sind in Kapitel 7 oder im später fertiggestellten Entwicklungshandbuch zu finden.

⁴Neue Metaregeln betreffen immer den Programmcode. Ein Beispiel für einen Operator, der mit Programmänderungen einhergeht, ist beispielsweise die Definition einer Menge bzw. einer Klasse durch Angabe einer Formel, e. g. $\{x \mid \phi(x)\}$.

⁵Siehe Abschnitt 1.8.

Kapitel 4

Prototyp

Es gibt bereits einen funktionierenden Prototypen, der wesentliche Anforderungen von **Hilbert II** bereits unterstützt. Die in Abschnitt 2.1 formulierten logischen Grundlagen wurden, mit Ausnahme der mit dem Begriff „Term“ in Zusammenhang stehenden Objekte, in der Prototypversion 0.00.53 umgesetzt. Die dazugehörige logische Ausgangsbasis ist auch unter http://www.qedeq.org/0_00_53/rules_1.00.00.html zu finden. Die Qedeq-Syntaxbeschreibung ist im Anhang B angegeben.

4.1 Qedeq-Module

Ausgangsbasis für die Aussagenlogik ist das Qedeq-Modul http://www.qedeq.org/0_00_53/propaxiom_1.00.00_1.00.00.qedeq. Daraus werden einfache Folgerungen gezogen und die aussagenlogischen Axiome der Principia Mathematica, dem Whitehead-Russell-Kalkül, abgeleitet. Dabei wird die vierte *primitive Proposition* nach dem Beweis von Bernays [2] erhalten. Anschließend erfolgt der systematische Aufbau der aussagenlogischen Gesetzmäßigkeiten, der sich an dem von Hilbert und Ackermann [3] vorgestellten orientiert. Siehe dazu auch die Datei <http://www.qedeq.org/doc/deduction.txt>.

Dabei werden neue Metaregeln abgeleitet, die die ursprünglichen Ableitungsregeln erweitern. Diese Regeln vereinfachen lediglich die Ableitung und können durch die ursprünglichen Ableitungsregeln ersetzt werden. Die neuen Regeln werden in der Version 1.02.00 zusammengefasst, die in dem Qedeq-Modul **prophilbert1** in genau dieser Version vorgestellt werden. Für alle diese Module wurden auch HTML-, L^AT_EX- und PDF-Dokumente erzeugt, die in der Übersicht http://www.qedeq.org/propositional_de.html zu finden sind.

Analog gibt es das Basismodul für die Prädikatenlogik http://www.qedeq.org/0_00_53/predaxiom_1.00.00_1.00.00.qedeq aus dem Sätze der Prädikatenlogik abgeleitet werden. Unter der Adresse <http://www.qedeq.org/predicate.html> ist eine Übersicht darüber zu finden.

4.2 Funktionalität

Der Prototyp kann seit der Version 0.00.53 auch über eine graphische Benutzeroberfläche bedient werden. Er ist sogar von der Seite http://www.qedeq.org/webstart_de.html aus startbar, falls Ihr Browser das unterstützt.

Er arbeitet mit (prototypspezifischen) Qedeq-Moduldateien zusammen, die sich irgendwo im Internet befinden können. Es kommen die Protokolle **http** und **ftp**

zum Einsatz, aber auch lokale Dateien können angegeben werden. Bei der Eingabe einer URL eines solchen Moduls wird der lokale Datei-Puffer durchsucht. Wenn dieser die Datei nicht enthält, dann wird die durch die URL angegebene Datei heruntergeladen und in dem lokalen Datei-Puffer gespeichert. Anschließend wird versucht das Qedeq-Modul zu laden, dabei findet eine Prüfung auf Korrektheit statt. Falls andere Qedeq-Module referenziert werden, wird versucht diese ebenfalls zu laden. Erst wenn alle notwendigen Qedeq-Module erfolgreich geladen und überprüft worden sind, bekommt das ursprünglich angeforderte Qedeq-Modul seinen „grünen Korrektheitspunkt“. Im Fehlerfalle wird eine detaillierte Problembeschreibung angegeben und auch die Problemstelle im entsprechenden Modul angezeigt.

Aus erfolgreich geladenen und überprüften Modulen können weitere Dokumente erzeugt werden. Der Prototyp kann HTML- und L^AT_EX-Dateien generieren. Der Prototyp kann auch alle Metaregeln automatisch eliminieren und die Beweise so abändern, dass nur die Basisregeln verwendet werden. Dazu wird jede Anwendung einer Metaregel in eine Abfolge von Anwendungen der Basisregeln umgewandelt. Das neu erzeugte Qedeq-Modul wird dann gespeichert. Auch können Qedeq-Module von doppelten Beweiszeilen befreit werden.

Innerhalb des Prototypen können lokale Qedeq-Module erzeugt und editiert werden. Es kann aber auch jeder einfache Texteditor Verwendung finden. Bereits im Netz befindliche Qedeq-Module können durch einfache Referenzierung benutzt werden.

Hauptprojekt benötigt nicht die extreme Detailtiefe der Beweise des Prototypen, aber diese detaillierte Form sollte dort auch generierbar sein.

Kapitel 5

Arbeitsbereiche

Die Projektorganisation ist noch in Entstehung begriffen. Nachfolgend werden einige Arbeitsbereiche angegeben. Dabei gibt es teilweise starke Verbindungen und Überlappungen zwischen verschiedenen Bereichen. Teilweise wird auch durch Sourceforge¹ logistische Unterstützung angeboten. Gerade bei einem Open-Source-Projekt muss die Projektorganisation den Gegebenheiten jeweils dynamisch angepasst werden.

5.1 Erstellung von mathematischen Texten

In späteren Aufbaustufen können weitere Hilfsmittel zur Erstellung von mathematischen Dokumenten zum Einsatz kommen. Hier wird nur das anfängliche Vorgehen beschrieben.

5.1.1 Mathematik

Das Ziel von **Hilbert II** ist im Wesentlichen die konkrete Darstellung von Mathematik. Das Ergebnis ist in erster Näherung ein üblicher mathematischer Artikel. Mathematische Artikel werden im ersten Schritt in nichtformaler Form in einem üblichen Texteditor erfasst. Dabei werden bestimmte Formatierungen verwendet, z. B. `all` für den Allquantor und `in` für die Elementbeziehung.²

5.1.2 Formalisierung

Im nächsten Schritt folgt nun die Übertragung in ein Qedeq-Modul. Dabei können auch Parser zum Einsatz kommen, die beispielsweise aus einer bestimmten bequemen Syntax formale Qedeq-Ausdrücke erzeugen.³ Diese halbformale Form wird dann zu einem formal korrekten Qedeq-Modul ergänzt.

5.1.3 Weitere Ergänzungen

Um in dem Qedeq-Modul auch Texte in anderen Sprachen oder einem anderen Detaillierungsgrad zu haben, kann es erforderlich sein, eine Zusammenführung

¹Siehe auch auf den Projektseiten: <http://sourceforge.net/projects/pmia>.

²Es sogar möglich sein, den mathematischen Text in LATEX zu schreiben.

³Beispielsweise könnte aus der Zeichenfolge `all x in A: x in B impl x in C` der Qedeq-Ausschnitt `FORALL(VAR(1), IN(VAR(1), VAR(28)), IMPL(IN(VAR(1), VAR(27)), IN(VAR(1), VAR(29))))` erzeugt werden.

dieser verschiedenen Anteile vorzunehmen. Dabei sollte zunächst eine Stufe komplett fertiggestellt sein, und dieses formal korrekte Qedeq-Modul wird dann entsprechend ergänzt.

5.2 Qedeq-Syntaxdefinition

Die Syntax des Qedeq-Formats muss überprüft und weiterentwickelt werden. Ein weiterer Punkt ist die Definition der benötigten und unterstützten L^AT_EX-Befehle. Diese Festlegungen sollten in enger Abstimmung mit den Erstellern von Qedeq-Dateien und den Kernel-Entwicklern geschehen. Siehe auch Kapitel 1 und Kapitel 3.

5.3 Kernelentwicklung

Hierzu gehört die Programmentwicklung des Kernels. Mit dabei sind die Implementierung des Proofcheckers, des Regeltransformers und der Konverter in verschiedene andere Formate wie z. B. XML, HTML, L^AT_EX und PDF.

5.4 Anwendungsentwicklung

Der Qedeq-Viewer mit seinen verschiedenen Darstellungs- und Analysemöglichkeiten ist eine große Aufgabe der Anwendungsentwicklung.

5.5 Test und Qualitätssicherung

Vor der Zusammenstellung von neuen Distributionen oder der Veröffentlichung von neuen Qedeq-Dateien wird ein entsprechender Test und eine Qualitätskontrolle erfolgen.

5.6 Übersetzung

Nicht nur die mathematischen Texte, auch die Dokumentation und der Internetauftritt müssen in andere Sprachen übersetzt werden. Insbesondere eine Übertragung in das Englische sollte vorgenommen werden.

5.7 Releasemanagement und Distribution

Die Erstellung einer neuen Version der Programmmpakete von **Hilbert II** erfordert eine gewisse Organisation. Eine CD-Zusammenstellung ist angedacht. Auch die Verteilung des neuen Release ist aktiv anzustößen. Das neue Programmmpaket muss zusammengestellt und verteilt werden. Denkbar ist auch die Definition eines Debian-Paketes⁴, eventuell auch aufbauend auf dem *GNU Compiler für Java* (<http://gcc.gnu.org/java/>) oder *Kaffe* (<http://www.kaffe.org/>), einer freien Implementierung von Java.⁵

⁴Das Debian-Projekt ist eine Gemeinschaft von Individuen, die in Gemeinschaftsarbeite ein freies Betriebssystem entwickeln. Dieses Betriebssystem wird Debian GNU/Linux genannt, oder einfach nur Debian. Debian-Systeme verwenden im Augenblick den Linux-Kern. Linux ist ein von Linus Torvalds begonnenes komplett freies Stück Software, das von tausenden Programmierern weltweit entwickelt wird.

⁵Siehe Abschnitt 3.7.

5.8 Support und Fehlerverfolgung

Die Sammlung und Organisation von Anfragen und Fehlermeldungen und deren Beantwortung gehören zu diesem Arbeitsbereich. Auch die Betreuung von Anwender-Foren gehört dazu.

5.9 Organisation des Internet-Auftritts

Die Internetseiten bedürfen der wöchentlichen Pflege und Aktualisierung. Die aktuellen Web-Seiten von <http://www.qepeq.org> werden zur Zeit mit einem eigenen Programm generiert.⁶

5.10 Dokumentation und Redaktion

Die Dokumentation ist ein häufig vernachlässigtes Gebiet. Dazu gehören nicht nur Programmbeschreibungen, sondern auch Anleitungen⁷ zur Lösung der verschiedenen Aufgabenstellungen. Auch die mathematischen Texte benötigen häufig einen guten Editor.

5.11 Gestaltung und Design

Die Dokumente und Internet-Seiten dieses Projekts bedürfen der Gestaltung. Auch ein Logo oder sogar ein CD-Cover können auf der Designliste stehen.

5.12 Öffentlichkeitsarbeit und Kontaktpflege

Das Projekt **Hilbert II** muss bekanntgemacht, E-Mails müssen beantwortet, Foren betreut werden. Eventuell gehört auch das Einwerben von Fördermitteln in diesen Arbeitsbereich.

5.13 Gesamtkoordination

Die Organisation des gesamten Projekts ist eine nicht einfache Aufgabe. Die Koordination der verschiedenen Arbeitsbereiche und die Abstimmung und Planung der weiteren Entwicklungen gehören dazu.

⁶Dieser Auftritt könnte beispielsweise komplett neu aufgestellt werden.

⁷Eine solche Anleitung wird oft auch *HowTo* genannt.

Kapitel 6

Planung

Eine Projektplanung im eigentlichen Sinne kann zu diesem Zeitpunkt noch nicht vorgenommen werden und wird auch in Zukunft nicht Bestandteil dieses Dokuments werden. In diesem Kapitel werden die Arbeitspakete für verschiedene Ausbaustufen von **Hilbert II** definiert.

6.1 Umfang der Version 1.00

Für die Version 1.00 wird die Syntax-Beschreibung der Qedeq-Dateien soweit festgelegt, dass zumindest die in dem Kapitel 2 angegebenen Formeln bequem ausgedrückt werden können. Sie beinhaltet einen Kernel, der eine Qedeq-Datei auf dieser syntaktischen Ebene überprüfen kann. D. h. es kann beispielsweise festgestellt werden, dass eine Formel nicht formal korrekt gebildet ist, da über die Variable x zweimal quantifiziert wurde. Es ist dem Kernel aber noch nicht möglich zu kontrollieren, ob in einem formalen Beweis eine Beweiszeile logisch aus den vorherigen folgt. Des Weiteren sollen in der Qedeq-Syntax die Anfangsgründe der Mengenlehre dargestellt werden können. Die Generierung von L^AT_EX-Dokumenten ist ebenfalls möglich.

Mathematisch werden die Anfangsgründe der Mengenlehre in knapper Form in Qedeq-Dateien dargelegt. Ausgehend von den grundlegenden Axiomen, Definitionen und Notationen werden die ersten Sätze der axiomatischen Mengenlehre formuliert. Idealerweise erfolgt die Darstellung in mehreren Detaillierungsstufen und in den Sprachen Deutsch und Englisch.

6.2 Umfang der Version 1.01

Der Umfang der Version 1.01 beinhaltet die Fähigkeit, formale Überprüfungen der in Version 1.00 erstellten Qedeq-Module vorzunehmen. Wsentliches Element dieser Version ist also der *ProofChecker*. Dazu kann es erforderlich sein, dass die bereits erstellten Qedeq-Dateien korrigiert oder ergänzt werden müssen.

+++

6.3 Umfang der Version 1.02

In dieser Version ist ein rudimentärer Qedeq-Viewer einsatzbereit. Er ermöglicht die Betrachtung von Qedeq-Dateien unter Berücksichtigung der einstellbaren

Parameter Detailtiefe und Sprache. Die Auflösung von Referenzen und Querverbindungen sind ebenfalls möglich.

Zur Darstellung der Texte und Formeln ist es notwendig, dass der Qedeq-Viewer elementare, oder zumindest die notwendigen L^AT_EX-Konstrukte unterstützt.

+++

6.4 Umfang der Version 1.03

Ein Kernel, mit dem Reduzierungsmöglichkeiten für die logischen Regeln aus Version 1.01 möglich sind, ist Bestandteil dieser Version. Inhaltlich wird damit an den Prototyp angeknüpft. Dazu gehört eine Ausarbeitung sämtlicher aussagen- und prädikatenlogischer Sätze.

+++

6.5 Weitere Aufgaben

Über die vorherigen Paketbeschreibungen hinaus gibt es natürlich die im Kapitel 5 angegebenen Aufgabenbereiche, die eine permanente Mitarbeit erfordern. Insbesondere die Erstellung von mathematischen Dokumenten gehört dazu.

Kapitel 7

Technische Realisierung

An dieser Stelle folgen Informationen, die später in ein Entwicklungshandbuch ausgelagert werden. Die Referenzimplementation erfolgt in Java 1.4.1 oder einer größeren Version.

In den ersten Versionen erfolgt keine Organisation in einer Multi-Tier-Architektur, sondern in einer Stand-Alone-Applikation. Die Qedeq-Dateien werden bei der Initialisierung des Kernels bzw. auf Anforderung direkt eingelesen und im Hauptspeicher verwaltet.

Es existieren bereits prototypische Basisklassen, am Designentwurf wird noch gefeilt.

+++

+++ PDF direkt und nicht über L^AT_EX erzeugen?

Kapitel 8

Einsatzszenarien

Als grober Richtwert für die Größe formaler Beweise im Vergleich zu nicht formalen Beweisen wird häufig der Faktor vier genannt, siehe zum Beispiel bei Wiedijk [9]. Dieser Wert wird auch *de Bruijn Faktor* genannt. Falls also die Qedeq-Syntax von **Hilbert II** entsprechend komfortabel ist, kann damit der Speicherbedarf für verschiedene Theorien abgeschätzt werden. In den folgenden Überlegungen wird der Einfachheit von einer Standalone-Applikation ausgegangen, alle Daten werden gleichzeitig im Hauptspeicher gehalten.

8.1 Ausstattungen

Es wird für „einige Jahre in der Zukunft“ geplant. Eine grobe Abschätzung ist, dass in ein paar Jahren 512 MB Arbeitsspeicher als Standard gelten.¹ Die Festplattenkapazität wird schätzungsweise 120 GB betragen. Als Kapazität der Netzanbindung wird von dem Durchsatz eines DSL-Anschlusses² ausgegangen.

8.2 Mengengerüst

Sicherheitshalber nehmen wir einen de Bruijn Faktor von zehn an. Die Größe dieses Faktors hängt von der Bequemlichkeit der Qedeq-Syntax, von der Qualität des Proofcheckers³ und den Detaillierungsgraden der Qedeq-Dateien ab.

Bei einem Ansatz von einem Speicherverbrauch von 2 MB pro mathematischem Textbuch ergibt sich ein Bedarf von 20 MB pro Qedeq-Modul.⁴ Bei der Annahme, dass das mathematische Basiswissen in 15 Büchern⁵ beschrieben werden kann, ergibt sich ein dynamischer Speicherbedarf von 300 MB. Bei dieser Größe ist noch genug Platz für die Anwendung und das Betriebssystem vorhanden. In späteren Jahren wird der verfügbare dynamische Speicher sicherlich zunehmen,

¹Das Wort „Standard“ bezieht sich auf die westlichen Industrienationen.

²Unter *DSL* wird eine Datenübertragungsmethode verstanden, die eine Übertragungsgeschwindigkeit im MegaBit-Bereich über die Telefonleitung ermöglicht.

³Hier zeigt sich, dass **Hilbert II** idealerweise über einen hervorragenden Theorembeweiser verfügen sollte, der in der Lage ist, die in mathematischen Texten üblicherweise äußerst knapp gehaltenen Beweise durch die erforderlichen Zwischenschritte zu ergänzen. Dabei könnte sich die Möglichkeit ergeben auch etablierte externe Theorembeweiser anzuschließen.

⁴Der Speicherverbrauch im Dateisystem ist sicherlich höher, da die Operatoren dort nicht abgekürzt vorkommen. Allerdings wäre der Verbrauch im Dateisystem geringer, wenn die Qedeq-Module dort in komprimierter Form abgelegt werden würden.

⁵Die Bücheraufteilung könnte beispielsweise so aussehen: Mengenlehre 1, Zahlentheorie 2, Algebra 2, Geometrie 1, Analysis 3, Stochastik 1, Numerik 1, Funktionentheorie 1, Topologie 1, Kombinatorik 1, Graphentheorie 1

so dass dann gleichzeitig auch die höhere Mathematik im Speicher gehalten werden kann.⁶

Die Module werden initial aus dem Internet geladen und lokal im Dateisystem gespeichert. Nur bei Änderungen ist eine Aktualisierung nötig. Die Größe des Plattspeichers ist zwar größer als der Bedarf an dynamischem Speicher, dennoch reichen einige GB sicherlich völlig aus.

Eine Abschätzung der Performance des Kernels, des Qedeq-Viewers und anderer Programmteile von **Hilbert II** ist zur Zeit nicht so einfach möglich. Es gibt jeweils viele starke Einflussgrößen, die nur schwer zu bewerten sind. Insbesondere ist zuvor eine genaue Spezifikation dieser Komponenten erforderlich.

8.3 Verwendung

Wie bereits im Kapitel 1 formuliert, ist das Ziel von **Hilbert II** eine Darstellung der Mathematik bzw. ihrer Grundlagen. Diese Darstellung hat bestimmte Eigenschaften, die Rahmenbedingungen für die Einsatzmöglichkeiten abstecken. Zunächst einmal ist diese Darstellung *frei* in dem von Anhang A definierten Sinne. Verfügbar ist sie – in unterschiedlicher Form – an jedem Ort mit Internetzugang.

Die Darstellung bietet die Grundlagen der Mathematik in textbuchartiger Form dar. Das bedeutet, dass **Hilbert II** als Online-Bibliothek für mathematisches Wissen dienen kann. Die erzeugten PDF-Dokumente können ganz normal als Skripte oder Textbücher benutzt werden.

Es kann aber auch, beispielsweise von einem mathematischen Satz ausgehend, zurückverfolgt werden, welche Definitionen und andere Sätze für den Beweis benötigt werden. Sogar die automatische Rückführung auf die mengentheoretischen Axiome ist möglich oder die Beantwortung der Frage, ob ein bestimmter Satz irgendwo in den Beweis eingeht.

Das durch **Hilbert II** formal erfasste mathematische Wissen ist auch formal korrekt, das wird durch den Proofchecker gewährleistet. In diesem Sinne kann **Hilbert II** auch als Kontrollinstanz für mathematisches Wissen fungieren.

⁶Zum einen ist allein die Bereitstellung von Qedeq-Dateien für das Basiswissen zeitaufwendig genug, um zum Zeitpunkt der Fertigstellung auf eine höhere Zahl von dynamischem Speicher zu hoffen. Zum anderen sollte es auch möglich sein *Schnittstellenmodule* zu schaffen, in denen die Beweise fehlen, die jedoch bei Bedarf in zugehörigen vollständigen Qedeq-Modulen nachgeladen werden können. Dieses Vorgehen könnte generell den Speicherverbrauch reduzieren.

Anhang A

GNU Free Documentation License

Version 1.2, November 2002

Copyright © 2000,2001,2002 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim copies of this license document,
but changing it is not allowed.

PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful
document “free” in the sense of freedom: to assure everyone the effective freedom to copy
and redistribute it, with or without modifying it, either commercially or noncommercially.
Secondarily, this License preserves for the author and publisher a way to get credit for their
work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document
must themselves be free in the same sense. It complements the GNU General Public
License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free
software needs free documentation: a free program should come with manuals providing
the same freedoms that the software does. But this License is not limited to software
manuals; it can be used for any textual work, regardless of subject matter or whether it
is published as a printed book. We recommend this License principally for works whose
purpose is instruction or reference.

APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice
placed by the copyright holder saying it can be distributed under the terms of this License.
Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that
work under the conditions stated herein. The “Document”, below, refers to any such
manual or work. Any member of the public is a licensee, and is addressed as “you”. You
accept the license if you copy, modify or distribute the work in a way requiring permission
under copyright law.

A “Modified Version” of the Document means any work containing the Document or a
portion of it, either copied verbatim, or with modifications and/or translated into another
language.

A “Secondary Section” is a named appendix or a front-matter section of the Document
that deals exclusively with the relationship of the publishers or authors of the Document
to the Document’s overall subject (or to related matters) and contains nothing that could

fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

A section “Entitled XYZ” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “Acknowledgements”, “Dedications”, “Endorsements”, or “History”.) To “Preserve the Title” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section “COPYING IN QUANTITY”.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections "VERBATIM COPYING" and "COPYING IN QUANTITY" above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

1. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
2. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
3. State on the Title page the name of the publisher of the Modified Version, as the publisher.
4. Preserve all the copyright notices of the Document.
5. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
6. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
7. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

8. Include an unaltered copy of this License.
9. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
10. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
11. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
12. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
13. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
14. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
15. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section "MODIFICATIONS" above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment

to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section "COPYING IN QUANTITY" is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section "MODIFICATIONS". Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section "MODIFICATIONS") to Preserve its Title (section "APPLICABILITY AND DEFINITIONS") will typically require changing the actual title.

TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright © year *your name*.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with...Texts.” line with this:

with the Invariant Sections being *list their titles*, with the Front-Cover Texts being *list*, and with the Back-Cover Texts being *list*.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Anhang B

Qedeq-Format

Die Dateiform von Qedeq-Modulen ist eine zentrale Schnittstelle von **Hilbert II**. Der im Prototypen vorliegende Stand ist unter http://www.qedeq.org/0_00_53/qedeq_de.csv¹ abrufbar und beschreibt das Format der Version 0.00.53. Diese Formatbeschreibung kann auch als *XML-DTD* unter http://www.qedeq.org/0_00_53/qedeq_de.dtd abgerufen werden.

Das Format einer Qedeq-Datei wird im Folgenden in *BNF-Notation*² angegeben. Unter http://purl.net/stefan_ram/pub/programmieren_bnf_de ist eine detaillierte Beschreibung dieser Notation zu finden. Die Regeln dieser Metasprache haben folgendes Aussehen:

```
<Identifier> ::= <Rule>;
```

„`<Identifier>`“ ist ein sogenanntes Nonterminal. Wann immer dieses Nonterminal in einer BNFRegel vorkommt, muss es durch seinen Regelteil „`<Rule>`“ ersetzt werden. Diese Ersetzungen erfolgen solange, bis keine weiteren Ersetzungen mehr möglich sind. Das Ergebnis ist ein korrekter Ausdruck des durch die BNF-Notation beschriebenen Kalküls. Die BNF-Regeln sind wie folgt aufgebaut:

- *Terminals*, das sind Zeichenketten, die zwischen zwei einfachen Anführungszeichen stehen. Diese Terminals werden ohne weitere Ersetzungen übernommen.
- *Nonterminals* sind Symbole, die durch die dazugehörigen Regeln ersetzt werden müssen.
- *Startterminal*, mit diesem Terminal muss begonnen werden.
- *Alternativen*, alternative Regelteile sind durch einen senkrechten Strich getrennt aufgeführt. Eine dieser so gekennzeichneten Alternativen muss übernommen werden.
- *Wiederholungen*, alles was durch geschweiften Klammern umschlossen wird, muss kein-, ein- oder mehrmals übernommen werden.
- *Auswahl*, alles in eckigen Klammern stehende, muss kein oder genau einmal übernommen werden.
- *Klammerung*, runde Klammern fassen zusammengehörende Regelteile zusammen.
- Eine Regel wird durch ein Semikolon abgeschlossen.

¹Die dort vorliegende Beschreibung ist in einem etwas anderen Format gehalten.

²Backus-Naur-Form.

Das Resultat einer bis zum Ende geführten Ersetzung ist also eine Folge von Terminals, die eine Zeichenfolge darstellen.

Die grundsätzliche Struktur des Qedeq-Formates lässt sich durch folgende Angaben beschreiben:

```
<argument> ::= (<funktionsname> '(' [ <argument> { ',' <argument>} ] ')') | <numerisches Atom> | <Textatom>
<ndigit> ::= '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9';
<digit> ::= '0' | <ndigit>;
<number> ::= '0' | ( [ '-' ] <ndigit> { <digit> } );
<text> ::= "" { <ASCII-Zeichen ohne Anführungszeichen> | """ , } """
;
```

Es gibt zwei Atome, das numerische Atom, welches eine beliebige ganze Zahl darstellt, und das Textatom, das einen beliebigen Text enthalten kann.

Für die 0.00.53-Version, welche vom Prototypen verwendet wird, gilt die folgende Syntaxbeschreibung. Dabei gilt das Komma als Operator: es wird ein ',' eingefügt, falls kein ')' folgt.

<MODULE>	Startterminal für das Qedeq-Format
::= 'MODULE(, <HEADER> , [<IMPORTS>] , [<USEDBY>] , <PARAGRAPHS> ')';	Kopfsatz Angaben zu importierten Modulen diese Module benutzen dieses Modul Abschnitte
<HEADER>	Kopfsatz eines Moduls
::= 'HEADER(, <SPEC> , <HEADLINE> , <DESCRIPTION> , <EMAIL> , <AUTHORS> ')';	Spezifikation dieses Moduls Titel dieses Moduls Beschreibung des Moduls ("abstract") Email-Adresse des Modul-Administrators Autorenliste
<SPEC>	Spezifikation eines Moduls
::= 'SPEC(, <NAME> , <VERSION> , <VERSION> , <LOCATIONS> ')';	Name des Moduls Version des Moduls benötigte Regelversion Liste von absoluten oder relativen URLs, an denen das Modul zu finden ist
<NAME>	Name
::= 'NAME(, <text> ')';	
<VERSION>	Versionsnummer im Format nn.nn.nn
::= 'VERSION(, <text> ')';	

<LOCATIONS>	Liste von Modul-URLs
::= 'LOCATIONS(, <LOCATION>, {<LOCATION>})';	mindestens eine URL muss angegeben sein
<LOCATION>	relative oder absolute URL, die auf ein Modul zeigt
::= 'LOCATION(, <text>)';	
<HEADLINE>	Titel des Moduls
::= 'HEADLINE(, <text>)';	
<DESCRIPTION>	Beschreibung des Moduls ("abstract")
::= 'DESCRIPTION(, <text>)';	
<EMAIL>	Email-Adresse
::= 'EMAIL(, <text>)';	
<AUTHORS>	Liste von Autoren
::= 'AUTHORS(, <AUTHOR>, {<AUTHOR>})';	
<AUTHOR>	
::= 'AUTHOR(, <text>, <EMAIL>)';	Autorname Email-Adresse des Autors
<IMPORTS>	Liste der importierten Module
::= 'IMPORTS(, <IMPORT>, {<IMPORT>})';	
<IMPORT>	Angaben zu einem zu importierenden Modul
::= 'IMPORT(, <SPEC>, <LABEL>)';	Spezifikation des gewünschten Moduls Abkürzung für das importierte Modul

<LABEL>	Anker, auf den verwiesen werden kann
::= 'LABEL(, ⟨text⟩);';	
<USEDBY>	Liste der importierten Module
::= 'USEDBY(, ⟨SPEC⟩, {⟨SPEC⟩});';	
<PARAGRAPHS>	Abschnitte
::= 'PARAGRAPHS(, ⟨PARAGRAPH⟩, {⟨PARAGRAPH⟩});';	
<PARAGRAPH>	Abschnitt
::= 'PARAGRAPH(, [⟨text⟩], (⟨ABBREVIATION⟩ ⟨AXIOM⟩ ⟨DECLARERULE⟩ ⟨PROPOSITION⟩), [⟨text⟩]);';	LaTeX-text or list LaTeX-text
<ABBREVIATION>	Definition einer Abkürzung
::= 'ABBREVIATION(, ⟨FORMULA⟩, ⟨FORMULA⟩);';	zu definierender Operator mit Pattern-Variablen durch diesen Ausdruck wird der Operator definiert, es werden dieselben Pattern-Variablen wie zuvor verwendet
<FORMULA>	eine Formel der Prädikatenlogik
(⟨PROP⟩ ⟨NOT⟩ ⟨AND⟩ ⟨OR⟩ ⟨IMPL⟩ ⟨EQUI⟩ ⟨PREDVAR⟩ ⟨FORALL⟩ ⟨EXISTS⟩ ⟨PATTERN⟩));';	Aussagenvariable Negation Konjunktion Diskunktion Implikation Äquivalenz Prädikatsvariable Allquantor Existenzquantor Pattern-Variable or list

$\langle \text{PROP} \rangle$	Aussagenvariable
$::= \text{'PROP('},$	
$\quad \langle \text{number} \rangle$	Nummer der Aussagenvariable
')');	
$\langle \text{NOT} \rangle$	Negation
$::= \text{'NOT('},$	
$\quad \langle \text{FORMULA} \rangle$	Formel
')');	
$\langle \text{AND} \rangle$	Konjunktion
$::= \text{'AND('},$	
$\quad \langle \text{FORMULA} \rangle,$	Formel
$\quad \langle \text{FORMULA} \rangle$	Formel
')');	
$\langle \text{OR} \rangle$	Disjunktion
$::= \text{'OR('},$	
$\quad \langle \text{FORMULA} \rangle,$	Formel
$\quad \langle \text{FORMULA} \rangle$	Formel
')');	
$\langle \text{IMPL} \rangle$	Implikation
$::= \text{'IMPL('},$	
$\quad \langle \text{FORMULA} \rangle,$	Formel
$\quad \langle \text{FORMULA} \rangle$	Formel
')');	
$\langle \text{EQUI} \rangle$	Äquivalenz
$::= \text{'EQUI('},$	
$\quad \langle \text{FORMULA} \rangle,$	Formel
$\quad \langle \text{FORMULA} \rangle$	Formel
')');	
$\langle \text{PREDVAR} \rangle$	Prädikatsvariable
$::= \text{'PREDVAR('},$	
$\quad \langle \text{number} \rangle,$	Nummer der Prädikatsvariable
$\quad \langle L \rangle$	Liste der Subjektvariablen
')');	
$\langle L \rangle$	Liste der Subjektvariablen
$::= \text{'L('},$	
$\quad \{\langle \text{VAR} \rangle\}$	
')');	
$\langle \text{VAR} \rangle$	Subjektvariable
$::= \text{'VAR('},$	
$\quad \langle \text{number} \rangle$	Nummer der Subjektvariable
')');	

<FORALL>	Allquantor
::= 'FORALL(, <VAR> , <FORMULA>)';	über diese Subjektvariable wird quantifiziert Formel, die obige Subjektvariable frei enthält
<EXISTS>	Existenzquantor
::= 'EXISTS(, <VAR> , <FORMULA>)';	über diese Subjektvariable wird quantifiziert Formel, die obige Subjektvariable frei enthält
<PATTERN>	Pattern-Variable
::= 'PATTERN(, <number>)';	Nummer der Pattern-Variable
<AXIOM>	Festlegung eines Axioms, d. h. einer augezeichneten Formel
::= 'AXIOM(, <FORMULA>)';	diese Formel ist das eigentliche Axiom
<DECLARERULE>	Deklaration einer neuen Regel
::= 'DECLARERULE(, <text> , <text> , {<LINK>})';	Name der Regel, die deklariert werden soll kurze Beschreibung der Regel Referenzen auf benötigte Axiome, Sätze
<LINK>	Referenziert ein Axiom, Satz, etc.
::= 'LINK(, <text>)';	Name des Labels
<PROPOSITION>	eine Proposition besteht aus einem mathematischem Satz und seinem Beweis
::= 'PROPOSITION(, <SENTENCE> , <PROOF>)';	der Satz der Beweis, die Formel der letzten Beweiszeile muss mit der Formel des Satzes übereinstimmen
<SENTENCE>	mathematischer Satz
::= 'SENTENCE(, <FORMULA>)';	wahre mathematische Formel

<PROOF>

```
::= 'PROOF('
    ⟨LINE⟩,
    {⟨LINE⟩}
  ')';
```

Beweis eines mathematischen Satzes, er besteht aus lauter Beweiszeilen, die aufeinander aufbauen. Eine einzelne Beweiszeile kann durch Anwendung der logischen Regeln aus vorherigen Beweiszeilen oder Axiomen, Abkürzungen oder Definitionen abgeleitet werden

<LINE>

```
::= 'LINE('
```

```
    ⟨FORMULA⟩,
```

```
(
```

```
    ⟨ADDAXIOM⟩ |
```

```
    ⟨ADDSENTENCE⟩ |
```

```
    ⟨MODUSPONENS⟩ |
```

```
    ⟨REPLACEPROP⟩ |
```

```
    ⟨USEABBREVIATION⟩ |
```

```
    ⟨VERSEABBREVIATION⟩ |
```

Hinzufügen eines Axioms

Hinzufügen eines bereits bewiesenen Satzes

Ausführung von Modus Ponens

Ersetzung einer Aussagenvariable

Benutzung einer Abkürzung

Rückgängigmachen einer Abkürzung

```
|
```

```
    ⟨RENAMEFREEVARIABLE⟩ | Umbenennung einer freien Subjektvariablen
```

```
    ⟨RENAMEBOUNDVARIABLE⟩ | Umbenennung einer gebundenen Subjektvariablen
```

```
|
```

```
    ⟨REPLACEPREDICATE⟩ | Einsetzung einer Formel für ein Prädikat
```

```
    ⟨GENERALIZATION⟩ | Anwendung der Generalisierung
```

```
    ⟨PARTICULARIZATION⟩ | Anwendung der Partikularisierung
```

```
)
```

or list

```
'');
```

<ADDAXIOM>

```
::= 'ADDAXIOM('
```

```
    ⟨LINK⟩
```

```
');';
```

Hinzufügen eines Axioms

Referenz auf ein Axiom

<ADDSENTENCE>

```
::= 'ADDSENTENCE('
```

```
    ⟨LINK⟩
```

```
');';
```

Hinzufügen eines bereits bewiesenen Satzes

<MODUSPONENS>

```
::= 'MODUSPONENS('
```

```
    ⟨number⟩,
```

```
    ⟨number⟩
```

```
');';
```

Modus Ponens

Beweiszeilennummer, die die Formel A referenziert

Beweiszeilennummer, die die Formel $A \rightarrow B$ referenziert

<REPLACEPROP>

```
::= 'REPLACEPROP('
```

```
    ⟨number⟩,
```

Einsetzung für eine Aussagenvariable

Beweiszeile, auf die die Regel angewendet werden soll

<pre> ⟨PROP⟩, ⟨PROP⟩ '');;</pre>	<p>diese Aussagenvariable soll ersetzt werden Aussagenvariable, die eingesetzt wird</p>
<pre> ⟨USEABBREVIATION⟩ ::= 'USEABBREVIATION(' ⟨number⟩, ⟨LINK⟩, ⟨number⟩ ')';;</pre>	<p>Rückgängig machen einer Abkürzung</p>
<pre> ⟨REVERSEABBREVIATION⟩ Verwendung einer Abkürzung ::= 'REVERSEABBREVIATION(' ⟨number⟩, ⟨LINK⟩, ⟨number⟩ ')';;</pre>	<p>Beweiszeile, auf die die Regel angewendet werden soll Referenz auf die Abkürzung dieses Vorkommen der Abkürzung soll rückgängig gemacht werden</p>
<pre> ⟨RENAMEFREEVARIABLE⟩ Umbenennung einer freien Subjektvariablen ::= 'RENAMEFREEVARIABLE(' ⟨number⟩, ⟨VAR⟩, ⟨VAR⟩ ')';;</pre>	<p>Beweiszeile, auf die die Regel angewendet werden soll ersetze diese freie Subjektvariable durch diese (nicht gebundene) Subjektvariable</p>
<pre> ⟨RENAMEBOUNDVARIABLE⟩ Umbenennung einer gebundenen Subjektvariablen ::= 'RENAMEBOUNDVARIABLE(' ⟨number⟩, ⟨VAR⟩, ⟨VAR⟩, ⟨number⟩ ')';;</pre>	<p>Beweiszeile, auf die die Regel angewendet werden soll ersetze diese gebundene Subjektvariable durch diese (nicht freie) Subjektvariable, die dadurch nicht doppelt gebunden werden darf dieses Vorkommen des gebundenen Quantors mit der angegebenen Subjektvariablen soll ersetzt werden</p>
<pre> ⟨REPLACEPREDICATE⟩ Ersetzung eines Prädikats durch eine Formel ::= 'REPLACEPREDICATE(' ⟨number⟩, ⟨PREDVAR⟩, ⟨FORMULA⟩ ')';;</pre>	<p>Beweiszeile, auf die die Regel angewendet werden soll mit paarweise verschiedenen Pattern-Variablen als Einträgen welche dieselben Pattern-Variablen enthält, die Menge der freien Subjektvariablen der Formel muss disjunkt zu der Menge der gebundenen Subjektvariablen der Beweiszeilenformel sein, umgekehrt muss die Menge der gebundenen Subjektvariablen dieser Formel disjunkt zu der Menge der freien Subjektvariablen der Beweiszeilenformel sein, und die Prädikatenvariable darf nicht im Wirkungsbereich eines Quantors stehen, der in der Ersetzungsformel vorkommt</p>

') ' ;

<GENERALIZATION>
 ::= 'GENERALIZATION('
 <number>,

 <VAR>
 ') ' ;

Generalisierung

Beweiszeile, auf die die Regel angewendet werden soll,
 sie muss die Form $A \rightarrow B(x)$ haben, dabei darf die
 Subjektvariable x in A nicht vorkommen
 über diese Subjektvariable wird generalisiert

<PARTICULARIZATION>
 ::= 'PARTICULARIZATION('
 <number>,

 <VAR>
 ') ' ;

Partikularisierung

Beweiszeile, auf die die Regel angewendet werden soll,
 sie muss die Form $A(x) \rightarrow B$ haben, dabei darf die
 Subjektvariable x in B nicht vorkommen
 über diese Subjektvariable wird partikularisiert

Literaturverzeichnis

- [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, Springer, Berlin 1928
- [4] *P.S. Novikov*, Grundzüge der mathematischen Logik, VEB Deutscher Verlag der Wissenschaften, Berlin 1973 31 31
- [5] *J. D. Monk*, Introduction to Set Theory, McGraw-Hill, New York 1969
- [6] *J. Schmidt*, Mengenlehre I, Bibliographisches Institut, Mannheim 1966
- [7] *U. Friedrichsdorf, A. Prestel*, Mengenlehre für den Mathematiker, Vieweg, Braunschweig 1985
- [8] *G. Takeuti, W. M. Zaring*, Introduction to Axiomatic Set Theory, McGraw-Hill, New York 1971
- [9] *F. Wiedijk*, The de Bruijn factor, <http://www.cs.kun.nl/~freek/notes/index.html> 41

Index

- Abtrennungsregel, 17
- Allquantor, 16
 - eingeschränkter, 19
- Aufgaben, 37
- Aussagenvariable, 16
- Axiom
 - Auswahl-, 22
 - Ersetzungs-, 22
 - Extensionalitäts-, 20
 - Komprehensions-, 20
 - Paarmenge, 21
 - Potenzmenge, 21
 - Regularität, 21
 - Vereinigungsmenge, 21
- Axiome
 - der Mengenlehre, 19
 - Gleichheits-, 18
 - logische, 15, 17
- Bruijn, de, 41
- de Bruijn Faktor, 41
- Debian, 34
- Definition
 - Menge, 20
- Detailgrad, 13
- DSL, 41
 - Einerklasse, 22
 - Elementbeziehung, 20
 - Entwicklungshandbuch, 39
 - Existenzquantor, 16
 - eingeschränkter, 19
 - Extensionalität, 20
 - Formel, 16
 - frei, 11
 - freie Subjektvariable, 16
 - freier Zugang, 7, 11
 - Funktionskonstanten, 15
 - Funktionsvariablen, 15
 - gebundene Subjektvariable, 16
 - Generalisierung, 18
 - GFDL, 43
 - Gleichheit, 18, 20
 - GPL, 25
 - Identität, 18
- Java, 29
- Kernel, 25
 - Entwicklung, 34
- Klasse, 20
 - Angabe durch Aufzählung, 22
 - Einer-, 22
- Klassenpaar
 - geordnetes, 22
- Komprehension, 20
- Konstante
 - Funktions-, 15
 - Prädikaten-, 15
- Konverter, 25
- Korrekttheit, formale, 12
- Korrektheitsgrad, 6
- L^AT_EX, 28
- Leibnizsche Ersetzbarkeit, 19
- Literatur
 - Mengenlehre, 22
- Logik, 15
- MathML, 28
- Menge, 20
- Mengengerüst, 41
- Mengenlehre, 19
- Metaregeln, 13
- MK, 20
- Modus Ponens, 17
- NBG, 20
- Operatordefinition, mehrfache, 29
- Partikularisierung, 18
- Principia Mathematica, 10, 31
- Programmiersprache, 29
- Projektplanung, 37
- Proofchecker, 5, 12, 34, 41
- Prototyp, 31, 50
- Prädikatenkonstante, 15
- Prädikatenvariable, 15
- Qedeq, 11, 26
 - Format, 49
 - Beispiel, 26
 - Erweiterungen, 28
 - formale Korrektheit, 12

-Viewer, 12, 25
Quantor
 All-, 16
 eingeschränkter, 19
 Existenz-, 16
Regeltransformer, 25, 34
Regelversion, 25, 26
Regelversionen, 13
Schnittstellenmodul, 42
Sprachen, 12
Subjektvariable, 15
 freie, 16
 gebundene, 16
Term, 16
Theorembeweiser, 5, 41
Variable
 Aussagen-, 16
 Funktions-, 15
 Prädikaten-, 15
 Subjekt-, 15
Vernetzung, 12
Version, 26
Versionen, 13
Verwendung, 42
XML, 28
ZFG, 20
Zusammenfassung, 5
Zwischenschritte, 41