Dr. Winfried Teschers
Anton-Günther-Straße 26c
91083 Baiersdorf
winfried.teschers@t-online.de

Projektdokument

ASBA

Axiome, Sätze, Beweise und Auswertungen

Projekt zur maschinellen Überprüfung von mathematischen Beweisen und deren Ausgabe in lesbarer Form

Winfried Teschers

30. September 2017

Es wird ein Programmsystem beschrieben, das zu eingegebenen Axiomen, Sätzen, und Beweisen letztere prüft, Auswertungen generiert und zu gegebenen Ausgabeschemata eine Ausgabe der Elemente in üblicher Formelschreibweise im LaTeX-Format erstellt.

Inhaltsverzeichnis

1. Analyse

In der Mathematik gibt es eine unüberschaubare Menge an Axiomen, Sätzen, Beweisen, *Fachbegriffen*[1)] und *Fachgebieten*[2)]. Zu den meisten Fachgebieten gibt es noch ungelöste Probleme.

Es fehlt ein System, das einen Überblick bietet und die Möglichkeit, Beweise automatisch zu überprüfen. Außerdem sollte all dies in üblicher mathematischer Schreibweise ein- und ausgegeben werden können. In diesem Dokument werden die Grundlagen für das zu entwickelnde Programmsystem, das **A**xiome, **S**ätze, **B**eweise und **A**uswertungen behandeln kann (ASBA) behandelt.

Ein Programmsystem mit ähnlicher Aufgabenstellung findet sich im GitHub Projekt *Hilbert II* (siehe [18, 19]). Einige Ideen sind von dort übernommen worden.

1.1. Fragen

Einige der Fragen, die in diesem Zusammenhang auftauchen, werden nun formuliert:

1. *Grundlagen*: Was sind die Grundlagen? Z. B. welche Logik und Mengenlehre.
2. *Basis*: Welche wichtigen Axiome, Sätze, Beweise, Fachbegriffe und Fachgebiete gibt es? Welche davon sind Standard?
3. *Axiome*: Welche Axiome werden bei einem Satz oder Beweis vorausgesetzt? Allgemein anerkannte oder auch strittige, wie z. B. den *Satz vom ausgeschlossenen Dritten* (*tertium non datur*) oder das *Auswahlaxiom*.
4. *Beweis*: Ist ein Beweis fehlerfrei?
5. *Konstruktion*: Gibt es einen konstruktiven Beweis?
6. *Vergleiche*: Welcher Beweis ist besser? Nach welchem Kriterium? Z. B. elegant, kurz, einsichtig oder wenige Axiome. Was heißt eigentlich *elegant*?
7. *Definitionen*: Was ist mit einem Fachbegriff jeweils genau gemeint? Z. B. *Stetigkeit*, *Integral* und *Analysis*.
8. *Abhängigkeiten*: Wie heißt ein Fachbegriff in einer anderen Sprache? Ist wirklich dasselbe gemeint? Was ist mit Fachbegriffen in verschiedenen Fachgebieten?
9. *Überblick*: Ist ein Axiom, Satz, Beweis oder Fachbegriff schon einmal – ggf. abweichend – definiert, formuliert oder bewiesen worden?
10. *Darstellung*: Wie kann man einen Satz und den zugehörigen Beweis – ggf. auch spezifisch für ein Fachgebiet – darstellen?

---

1) *Fachbegriffe* sind Namen für mathematische Elemente und Konstruktionen, z. B. Axiome, Sätze, Beweise und Fachgebiete. *Symbole* können als spezielle Fachbegriffe aufgefasst werden.

2) Ein *Fachgebiet* ist ein Teil der Mathematik mit einer zugehörigen Basis an Axiomen, Sätzen, Fachbegriffen und Darstellungen, z. B. *Logik*, *Mengenlehre* und *Gruppentheorie*. Ein Fachgebiet kann sehr klein sein und im Extremfall kein einziges Element enthalten. *Umgebung* wäre eine bessere Bezeichnung, ist aber schon ein verbreiteter Fachbegriff, so dass hier die Bezeichnung Fachgebiet verwendet wird.
Statt „Fachgebiet“ könnte man auch „Theorie“ nehmen. An Theorien werden aber bestimmte Anforderungen gestellt, die aber nicht notwendigerweise überprüft werden sollen. Theorien sind allerdings i. Alg. auch Fachgebiete.

11. *Forschung*: Welche Probleme gibt es noch zu erforschen.

1.2. Eigenschaften

ASBA soll ausgehend von den Fragen in Abschnitt 1.1 auf der vorherigen Seite entwickelt werden, und die folgenden Eigenschaften haben:

1. *Daten*: Axiome, Sätze, Beweise, Fachbegriffe und Fachgebiete können in formaler Form gespeichert werden – auch nicht oder unvollständig bewiesene Sätze. Dabei soll die übliche mathematische Schreibweise verwendet werden können.
2. *Definitionen*: Es können Fachbegriffe für Axiome, Sätze, Beweise und Fachgebiete – letztere mit eigenen Axiomen, Sätzen, Beweisen, Fachbegriffen und über- oder untergeordneten Fachgebieten – definiert werden. Die Definitionen dürfen wiederum an dieser Stelle schon bekannte Fachbegriffe und Fachgebiete verwenden.
3. *Prüfung*: Vorhandene Beweise können automatisch geprüft werden.
4. *Ausgaben*: Die Axiome, Sätze und Beweise können in üblicher Schreibweise – abhängig von Sprache und Fachgebiet – ausgegeben werden.
5. *Auswertungen*: Zusätzlich zur Ausgabe der gespeicherten Daten sind verschiedene Auswertungen möglich, unter anderem für die meisten der unter Abschnitt 1.1 auf der vorherigen Seite behandelten Fragen.

Damit ASBA nicht umsonst erstellt wird und möglichst breite Verwendung findet, werden noch zwei Punkte angefügt:

6. *Lizenz*: Die Software ist *Open Source*.
7. *Akzeptanz*: ASBA wird von Mathematikern akzeptiert und verwendet.

Tabelle 1.1 auf der nächsten Seite zeigt, wie sich die Eigenschaften zu den Fragen im Abschnitt 1.1 auf der vorherigen Seite verhalten. Mit einem X werden die Spalten einer Zeile markiert, deren zugehörige Eigenschaften zur Beantwortung der entsprechenden Frage beitragen sollen. Idealerweise sollte die Erfüllung aller angegebenen Eigenschaften alle gestellten Fragen beantworten, was allerdings illusorisch ist.

1.3. Ziele

Um die Eigenschaften von Abschnitt 1.2 zu erreichen, werden für ASBA die folgenden Ziele[3)] gesetzt:

1. *Daten*: Es enthält möglichst viele wichtige Axiome, Sätze, Beweise, Fachbegriffe, Fachgebiete und Ausgabeschemata[4)].
2. *Form*: Die Daten liegt in formaler, geprüfter Form vor.
3. *Eingaben*: Die Eingabe von Daten erfolgt in einer formalen Syntax unter Verwendung der üblichen mathematischen Schreibweise.
4. *Prüfung*: Vorhandene Beweise können automatisch geprüft werden.

[3)] Es sind eigentlich Anforderungen. Da dieser Begriff auch im Kapitel 3 auf Seite 24 verwendet wird, werden die Anforderungen hier *Ziele* genannt.
[4)] Um den Punkt 4 von Abschnitt 1.2 erfüllen zu können, werden noch fachgebietsspezifische Ausgabeschemata benötigt, welche die Art der Ausgaben beschreiben.

Frage \ Eigenschaft	1 Daten	2 Definitionen	3 Prüfung	4 Ausgaben	5 Auswertungen	6 Lizenz	7 Akzeptanz
1 Grundlagen	X	X	-	X	X	-	-
2 Basis	X	X	-	X	X	-	-
3 Axiome	X	X	-	X	X	-	-
4 Beweis	X	-	X	X	-	-	-
5 Konstruktion	X	-	-	X	-	-	-
6 Vergleiche	X	-	-	-	X	-	-
7 Definitionen	X	X	-	X	-	-	-
8 Abhängigkeiten	X	-	-	X	-	-	-
9 Überblick	X	-	-	-	X	-	-
10 Darstellung	-	X	-	X	-	-	-
11 Forschung	X	-	-	-	X	-	-

Tabelle 1.1.: 1.1 Fragen → 1.2 Eigenschaften

5. *Ausgaben*: Die Ausgabe kann in einer eindeutigen, formalen Syntax gemäß vorhandener Ausgabeschemata erfolgen.

6. *Auswertungen*: Zusätzlich zur Ausgabe der Daten sind verschiedene Auswertungen möglich. Insbesondere kann zu jedem Beweis angegeben werden, wie lang er ist und welche Axiome und Sätze[5)] er benötigt.

7. *Anpassbarkeit*: Fachbegriffe und die Darstellung bei der Ausgabe können mit Hilfe von – gegebenenfalls unbenannten – untergeordneten Fachgebieten angepasst werden.

8. *Individualität*: Axiome und Sätze können für jeden Beweis individuell vorausgesetzt werden. Dabei sind fachgebietsspezifische Fachbegriffe erlaubt.

9. *Internet*: Die Daten können auf mehrere Dateien verteilt sein. Ein Teil davon – oder sogar alle – können im Internet liegen.

10. *Kommunikation*: Die Kommunikation mit ASBA kann mit den Fachbegriffen der einzelnen Fachgebiete erfolgen.

11. *Zugriff*: Der Zugriff auf ASBA kann lokal und über das Internet erfolgen.

12. *Unabhängigkeit*: ASBA kann online und offline arbeiten.

13. *Rekursion*: Es kann rekursiv über alle verwendeten Dateien – auch solchen, die im Internet liegen – ausgewertet werden.

14. *Bedienbarkeit*: ASBA ist einfach zu bedienen.

15. *Lizenz*: Die Software ist *Open Source*.

Der Punkt 16 wurde noch eingefügt, damit ASBA effizient arbeiten kann und um die Akzeptanz zu erhöhen:

16. *Zwischenspeicher*: Wichtige Auswertungen können an vorhandenen Dateien angehängt oder separat in eigenen Dateien gespeichert werden.

5) Sätze, die quasi als Axiome verwendet werden.

Die Tabelle 1.2 zeigt wieder, wie sich die Ziele zu den Eigenschaften im Abschnitt 1.2 auf Seite 5 verhalten. Mit einem X werden wieder die Spalten einer Zeile markiert, deren zugehörige Ziele zur Sicherstellung der entsprechenden Eigenschaft beitragen sollen. Idealerweise sollte durch Erreichen aller aufgestellten Ziele ASBA alle angegebenen Eigenschaften aufweisen, was wahrscheinlich ebenfalls illusorisch ist.

Eigenschaft \ **Ziel**	1 Daten	2 Form	3 Eingaben	4 Prüfung	5 Ausgaben	6 Auswertungen	7 Anpassbarkeit	8 Individualität	9 Internet	10 Kommunikation	11 Zugriff	12 Unabhängigkeit	13 Rekursion	14 Bedienbarkeit	15 Lizenz	16 Zwischenspeicher
1 Daten	X	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-
2 Definitionen	X	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-
3 Prüfung	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-
4 Ausgaben	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-
5 Auswertungen	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-
6 Lizenz	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-
7 Akzeptanz	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Tabelle 1.2.: 1.2 Eigenschaften → 1.3 Ziele

1.4. Zusammenfassung

Die Tabelle 1.3 auf der nächsten Seite ist eine Kombination der Tabellen 1.1 und 1.2 und zeigt, wie sich die Ziele im Abschnitt 1.3 auf Seite 5 zu den Fragen im Abschnitt 1.1 auf Seite 4 verhalten. Auch hier werden mit einem X die Spalten einer Zeile markiert, deren zugehörige Ziele für die Beantwortung der entsprechenden Frage nötig sind. Mit einem kleinen x werden sie markiert, wenn sie zur Beantwortung der Fragen nicht nötig, aber von Interesse sind. Idealerweise sollte das Erreichen aller aufgestellten Ziele alle gestellten Fragen beantworten, was natürlich auch illusorisch ist.

Frage \ **Ziel**		1 Daten	2 Form	3 Eingaben	4 Prüfung	5 Ausgaben	6 Auswertungen	7 Anpassbarkeit	8 Individualität	9 Internet	10 Kommunikation	11 Zugriff	12 Unabhängigkeit	13 Rekursion	14 Bedienbarkeit	15 Lizenz
1	Grundlagen	X	X	X	-	X	X	x	-	-	-	-	-	-	-	-
2	Basis	X	X	X	-	X	X	x	x	-	-	-	-	-	-	-
3	Axiome	X	X	X	-	X	X	x	-	-	-	-	-	-	-	-
4	Beweis	X	X	X	X	X	-	-	x	-	-	-	-	-	-	-
5	Konstruktion	X	X	X	-	X	-	-	x	-	-	-	-	-	-	-
6	Vergleiche	X	X	X	-	-	X	-	x	-	-	-	-	-	-	-
7	Definitionen	X	X	X	-	X	-	x	-	-	-	-	-	-	-	-
8	Abhängigkeiten	X	X	X	-	X	-	x	-	-	-	-	-	-	-	-
9	Überblick	X	X	X	-	-	X	x	-	-	-	-	-	-	-	-
10	Darstellung	X	-	X	-	X	-	x	-	-	-	-	-	-	-	-
11	Forschung	X	X	X	-	-	X	x	-	-	-	-	-	-	-	-
Die nächsten beiden Punkte sind Eigenschaften aus Abschnitt 1.2 auf Seite 5:																
6	Lizenz	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X
7	Akzeptanz	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Tabelle 1.3.: 1.1 Fragen → 1.3 Ziele

1.5. Die Umgebung von ASBA

In der Abbildung 1.1 wird beschrieben, welche Interaktionen ASBA mit der Umgebung hat, d. h. welche Ein- und Ausgaben existieren und woher sie kommen bzw. wohin sie gehen. In den in der

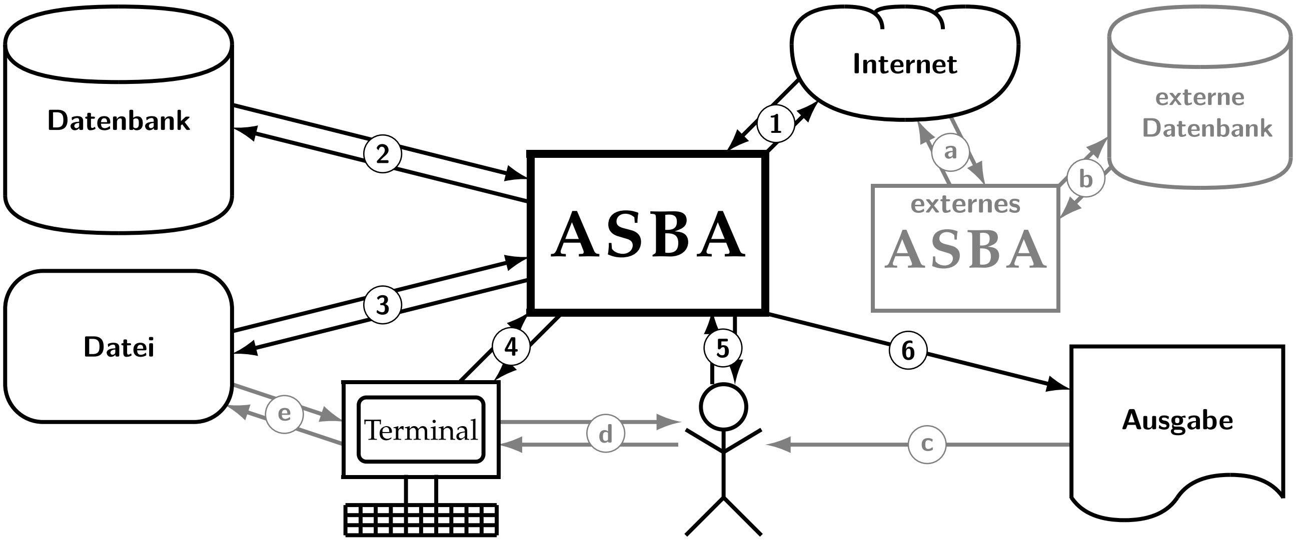

Abbildung 1.1.: Die Umgebung von ASBA

Abbildung 1.1 abgebildeten Datenflüssen (1) bis (6) und (a) bis (e) werden die folgenden Daten übertragen:

(1) **ASBA → Internet** Inhalte der Datenbank.

Internet → ASBA Inhalte der externen Datenbank.

(2) **Datenbank → ASBA** Inhalte der Datenbank und Antworten auf Datenbankanweisungen.

ASBA → Datenbank Inhalte der Datei, der externen Datenbank und Datenbankanweisungen.

(3) **Datei → ASBA** Inhalte der Datei.

ASBA → Datei Die Datei wird um zusätzliche Auswertungen ergänzt, z. B. ob die Beweise korrekt sind, welche Axiome und Sätze – auch externe aus dem Internet – verwendet wurden, Länge des Beweises usw.

(4) **Terminal → ASBA** Anweisungen, Daten und Batchprogramme.

ASBA → Terminal Antworten auf Anweisungen, Auswertungen usw.

Außerdem interaktive Ein- und Ausgabe durch einen Anwender, wie in (5) beschrieben.

(5) **Anwender ↔ ASBA** Interaktive Ein- und Ausgaben durch einen Anwender mit Komponenten von (3), (4) und (6). – Die Kommunikation läuft i. Alg. über ein Terminal.

(6) **ASBA → Ausgabe** Inhalte von Datei und Datenbank in lesbarer Form, u. a. mit Hilfe von Ausgabeschemata auch in Formelschreibweise. Die Ausgabe kann auch in eine Datei erfolgen, z. B. im LaTeX-Format.

(a) Nur die für ASBA relevanten Daten:

Internet → externes ASBA Inhalte der Datenbank.

externes ASBA → Internet Inhalte der externen Datenbank.

(b) Nur die für ASBA relevanten Daten:

externe Datenbank → externes ASBA Inhalte der externen Datenbank.

externes ASBA → externe Datenbank Inhalte der Datenbank.

(c) **Ausgabe → Anwender** Alle Daten der Ausgabe.

(d) **Anwender ↔ Terminal** Interaktive Ein- und Ausgabe durch einen Anwender, wie in (5) beschrieben.

(e) **Terminal ↔ Datei** Erstellen und Bearbeiten der Datei durch einen Anwender. – siehe (d)

Die Datenflüsse (a) bis (e) erfolgen außerhalb von ASBAund werden nicht weiter behandelt.

Die Datenbank und die Datei enthalten im Prinzip die gleichen Daten, wobei sie in der Datei im Textformat in lesbarer Form und in der Datenbank in einem internen Format vorliegen. Zudem enthält die Datenbank i. Alg. sehr viel mehr Daten. Es handelt sich dabei jeweils um die folgenden Daten:

Axiome Ein Axiom ist eine Aussage oder Behauptung, die nicht aus anderen Aussagen abgeleitet werden kann. Es können wie bei Sätzen Voraussetzungen vorhanden sein, aber keine Beweis.

Sätze Ein Satz besteht aus einer Anzahl von Voraussetzungen, einer Behauptung und einem Beweis, der die Behauptung aus den Voraussetzungen ableitet. Letztere können Axiome und andere Sätze sein, auf die dann verwiesen wird.

Beweise Ein Beweis besteht aus einer Folge von Beweisschritten, die aus gegebenen Voraussetzungen eine Behauptung ableitet.

Fachbegriffe Ein Fachbegriff ist ein Name für ein Objekt bzw. eine Eigenschaft in einem bestimmten Fachgebiet.

Fachgebiete Ein Fachgebiet ist ein Teil der Mathematik mit einer zugehörigen Basis von Axiome, Sätze, Fachbegriffe und Ausgabeschemata, quasi eine untergeordnete Datenbank.

Ausgabeschemata Eine Beschreibung, wie ein bestimmtes mathematisches Objekt ausgegeben werden soll. Dies kann z. B. ein Stück LaTeX-Code mit entsprechenden Parametern sein.

Auswertungen Statistische und andere Auswertungen, die bestimmten Elementen der Datei bzw. Datenbank zugeordnet sind. Z. B. können zu einem Satz alle für einen Beweis notwendigen Axiome angegeben werden – als Verweise.

Die Daten können interne und externe Verweise enthalten.

1.6. Basis von Beweisen

Da ein Computerprogramm erstellt werden soll, muss die Grundstruktur des Vorgehens bei Beweisen definiert werden.[6]

Die logischen Darstellung von mathematischen Aussagen, wozu auch Axiome und Sätze gehören, erfolgt, da es sich immer um Formeln handelt, an besten mit *Zeichenketten*.[7] Mehrdimensionale Formeln, wie z. B. Matrizen, Baumstrukturen, Funktionsschemata und anderes, können auch als (eindimensionale) Zeichenketten dargestellt werden. Beweise sind letztendlich nichts anderes, als erlaubte Transformationen dieser Zeichenketten.

Bausteine, also Grundelemente, auch *Zeichen* oder *(Satz-)Buchstaben* genannt, aus denen die Zeichenketten bestehen dürfen, müssen definiert werden.

Formationsregeln, mit denen festgelegt wird, wie man aus den Bausteinen Ausdrücke erzeugen kann, müssen ebenfalls definiert werden.

[6] siehe [30]

[7] Die interne Darstellung der Zeichenketten kann zur Optimierung des Programms von der logischen abweichen.

Sätze lassen sich als eine Menge von Formeln, den Voraussetzungen, wozu auch Axiome gehören können, einer weiteren Menge von Formeln (Zeichenketten), den Folgerungen, und der Angabe eines Beweises darstellen.

Beweise zu gegebenen Voraussetzungen und Folgerungen lassen sich als Folge von zulässigen Transformationen, beginnend mit den Voraussetzungen und endend mit den Folgerungen, darstellen.

Transformationsregeln definieren, welche Transformationen mit gegebenen Formelmengen zulässig sind.[8)]

8) siehe [1, 33, 34]

2. Mathematische Grundlagen

Die mathematischen Grundlagen werden einerseits gebraucht, um die erlaubten Beweisschritte zu definieren (siehe Abschnitt **??** auf Seite ??), andererseits dienen sie auch zum Testen von ASBA. Daher werden sie in diesem Kapitel ausführlicher behandelt, als für die Erstellung von ASBA erforderlich ist. Alle hier aufgeführten Axiome, Sätze und Beweise sollen dazu kodiert und die Beweise dann von ASBA verifiziert werden.

2.1. Notationen

Die in diesem Abschnitt 2.1 aufgeführten Notationen werden in diesem Kapitel 2 verwendet, ohne nochmals erläutert zu werden. Abweichungen davon werden jeweils gesondert angegeben.

Sätze mit „wir“ bestimmen Elemente, die nur für dieses Dokument gelten. Bei allgemeingültigen Elementen wird „wir“ nicht verwendet. Die Verwendung von „wir“ ist allerdings nicht konsistent und soll nur als Hinweis dienen.

2.1.1. Symbole und Bezeichnungen

$\mathbb{N}$ Menge der natürlichen Zahlen ohne 0.

$\mathbb{N}_0$ Menge der natürlichen Zahlen einschließlich 0.

2.1.2. Quotierung

Ein Zeichen oder Wort der natürlichen Sprache kann verschiedene Bedeutungen haben. Z. B. steht das Wort „sin“ für das Objekt Sinusfunktion, für den Namen, mit dem die Sinusfunktion bezeichnet wird, oder für eine Folge von Buchstaben. Letztere bezeichnen wir als *Zeichenfolge*, wenn Zwischenraum nicht zählt, sonst als *Zeichenkette*. Unter einer *Formel*[1)] verstehen wir in diesem Dokument stets eine mathematische Formel. Zur Unterscheidung verwenden wir verschiedene Quotierungen und Schriftarten:

sin	Ein Objekt	die Sinusfunktion.
‹sin›	Ein Symbol (Zeichen, Buchstabe)	für das Objekt.
«sin»	Eine Formel (Zeichenfolge)	bestehend aus dem einem Symbol ‹sin›
«*sin*»	Eine Formel (Zeichenfolge)	bestehend aus den drei Symbolen ‹*s*›, ‹*i*› und ‹*n*›
„`sin`“	Zeichenkette	bestehend aus den drei Zeichen ‹s›, ‹i› und ‹n›

Die Bezeichnung eines Objekts kann auch aus mehreren Symbolen bestehen, d. h. einer Zeichenfolge oder sogar einer ganzen Formel; z. B. ist die Bezeichnung für das indizierte Objekt a_i gleich «a_i». Die Bezeichnung für die Sinusfunktion hingegen wird als einzelnes Symbol behandelt.

1) Eine Formel kann auch mehrdimensional sein, lässt sich aber mittels geeigneter Definitionen immer eindeutig als eine Zeichenfolge schreiben.

Man beachte, dass in Formeln Worte in normaler (lateinischer) Schrift jeweils genau ein Symbol repräsentieren sollen. Die kursive Schrift ist Variablen vorbehalten und ein kursiv geschriebener Buchstabe ist immer ein einzelnes Symbol.

Auch Symbole, Formeln, Mengen, Zeichenfolgen, Zahlen, usw. werden wir als Objekte betrachten, Aussagen i. Alg. jedoch nicht.

2.1.3. Binäre Relationen und Operatoren

Seien ‹$\lhd$›, ‹$\rhd$›, ‹$\sim$›, ‹$\simeq$› und ‹$\nsim$› Beispielsymbole für Relationen.[2)] Wenn nichts anderes gesagt wird, gelte stets:

$$(A \lhd B) \quad :\Leftrightarrow \quad (B \rhd A) \tag{2.1}$$
$$(A \nsim B) \quad :\Leftrightarrow \quad [(A \sim B) \text{ gilt nicht}]$$
$$(A \simeq B) \quad :\Leftrightarrow \quad ((A \sim B) \,||\, (A = B)) \tag{2.2}$$
$$(A \sim B) \quad :\Leftrightarrow \quad ((A \simeq B) \,\&\, (A \neq B)) \tag{2.3}$$

In (2.1) lassen sich ‹$\lhd$› und ‹$\lhd$› auch vertauschen. Man beachte, dass weder (2.2) aus (2.3) folgt noch umgekehrt. Beispiele dazu sind in der Tabelle 2.1 angegeben.

	a, a	a, b	b, a	b, b	
$=$	$a = a$			$b = b$	
$\simeq$	$a \simeq a$	$a \simeq b$		$b \simeq b$	Es gilt (2.2)
$\sim$		$a \sim b$			und (2.3)
$\simeq$	$a \simeq a$	$a \simeq b$		$b \simeq b$	Es gilt (2.2)
$\sim$		$a \sim b$		$b \sim b$	aber nicht (2.3)
$\simeq$	$a \simeq a$	$a \simeq b$			Es gilt (2.3)
$\sim$		$a \sim b$			aber nicht (2.2)

Tabelle 2.1.: Beispiele

Als Beispielsymbol für binäre Operatoren wird ‹$\bullet$› verwendet. Mit ‹$\bullet$› zusammenhängende Verabredungen werden hier nicht getroffen.

Es sei noch angemerkt, dass wegen (2.1) die Definition von ‹$\Leftarrow$› im Unterabschnitt 2.2.1 auf der nächsten Seite überflüssig ist. Für den Fall fehlender Klammern sind die Prioritäten in der Tabelle 2.2 auf Seite 16 angegeben. Damit sind auch alle Klammern in diesem Unterabschnitt 2.1.3 überflüssig.

2.2. Metasprache

Wenn man über eine Sprache spricht, braucht man eine zweite Sprache, mit *Metasprache* bezeichnet, in der Aussagen über die erstere getroffen werden können.[3)] Wenn die zuerst genannte Sprache die der Mathematik ist, wählt man üblicherweise die natürliche Sprache als Metasprache. Leider ist diese oft

[2)] Die Relationen müssen weder Ordnungen noch Äquivalenzrelationen sein, auch wenn die angegebenen Symbole das nahe legen.

[3)] Die beiden Sprachen können auch übereinstimmen, z. B. wenn man über die natürliche Sprache spricht.

ungenau, nicht immer eindeutig und abhängig vom Zusammenhang, in dem sie gesprochen wird.[4)] Um diese Probleme in den Griff zu bekommen, kann die Metasprache teilweise formalisiert werden. Durch diese Formalisierung erinnert sie dann schon an mathematische Formeln. Die Sprachebenen sollten aber sorgfältig unterschieden werden.

2.2.1. Aussagen und Metaoperatoren

Beispiele für Aussagen in Metasprache sind (a) „Morgen scheint die Sonne.", (b) „Ich bin 1,83 m groß.", (c) „Ich habe ein rotes Auto und das kann 200 km/h schnell fahren.", usw. Wie Beispiel (c) zeigt, kann eine Aussage auch aus anderen Aussagen zusammengesetzt sein. In diesem Fall bezeichnen wir sie als , ansonsten als . – Wir betrachten auch Relationen einschließlich ihrer Operanden als Aussagen.[5)]

Während die Beispiele (a) und (b) unzerlegbare Aussagen sind, ist Beispiel (c) zerlegbar. Für alle drei Aussagen lässt sich feststellen, ob sie richtig sind oder nicht; für (a) allerdings nur im Nachhinein und für den zweiten Teil von (c) nur weil klar ist, worauf sich „das" bezieht. Natürlich muss auch der Zusammenhang, in dem eine Aussage formuliert wird, bekannt sein. Z. B. ist die Bedeutung von „Ich" nur dann bekannt, wenn man weiss, von wem die Aussage ist. Auf eine exakte Definition von Aussage wird verzichtet, weil das intuitive Verständnis hier ausreicht.

Zerlegbare Aussagen wie (c) können zum Teil formalisiert werden. Dies wird mit den folgenden Definitionen erreicht:

$A \Rightarrow B$	steht für „*Wenn* A [gilt] *dann* [gilt] [auch] B".
$A \Leftarrow B$	steht für „A [gilt] *sofern* B [gilt]".
$A \Leftrightarrow B$	steht für „A [gilt] *genau dann wenn* B [gilt]".
$A \mathbin{\&} B$	steht für „A *und* B".
$A \mathbin{\|\|} B$	steht für „A *oder* B".

Offensichtlich sind das alles ebenfalls Aussagen, jetzt aber teilweise formalisiert. (c) lässt sich dann ausdrücken als „‚Ich habe ein rotes Auto' & ‚das kann 200 km/h schnell fahren.'". „$A \Leftarrow B$" ist nur eine andere Schreibweise für „$B \Rightarrow A$". – Ein Symbol für „nicht" wird hier nicht gebraucht.

Aussagen können auch geklammert werden, um die Reihenfolge der Auswertung eindeutig zu machen. $\Rightarrow$, $\Leftarrow$, $\Leftrightarrow$, & und || heißen *Metaoperatoren*. Für den Fall fehlender Klammern sind ihre Prioritäten in der Tabelle 2.2 auf Seite 16 angegeben.

Um Verwechslungen mit den logischen Operatoren zu vermeiden, verwenden wir für die metasprachlichen Operatoren „und" und „oder" die Symbole ‹&› und ‹||›. A und B können als Operanden von $\Leftrightarrow$, & und || vertauscht werden, ohne das Ergebnis zu ändern.[6)] Wird in einer (Teil-)Aussage nur einer der Operatoren & oder || verwendet, können die Klammern dort weggelassen und die Operationen in beliebiger Reihenfolge ausgewertet werden, wiederum ohne das Ergebnis zu ändern.[7)] Zusammengefasst ist die Reihenfolge der Operatoren und der Auswertung dort beliebig.

[4)] Man betrachte die beiden Aussagen „Studenten und Rentner zahlen die Hälfte." und „Studenten oder Rentner zahlen die Hälfte.", die beide das gleiche meinen. – Entnommen aus [1] Abschnitt 1.2 Bemerkung 1.
Ein weiteres Problem ist, dass man unauflösbare Widersprüche formulieren kann, z. B. „Der Barbier ist der Mann im Ort, der genau die Männer im Ort rasiert, die sich nicht selbst rasieren.". Und der Barbier? Wenn er sich selbst rasiert, dann rasiert er sich nicht selbst, und wenn er sich nicht selbst rasiert, dann rasiert er sich selbst. Was denn nun? – Quelle unbekannt) – Das Problem ist verwandt mit dem Problem der „Menge aller Mengen, die sich nicht selbst enthalten".

[5)] Wird statt des Symbols der Name der zugehörigen Relation verwendet, ist dies unmittelbar einleuchtend. So wird z. B. aus der Formel «$A < B$» die Aussage „A ist kleiner als B".

[6)] D. h. die Operatoren $\Leftrightarrow$, & und || sind *kommutativ*.

[7)] D. h. die Operatoren & und || sind *assozativ*.

2.2.2. Mit Gleichheit verwandte Relationen

2.2.2.1. Vergleichbar

Zwei Objekte A und B sind *vergleichbar*, wenn beide von derselben Art sind, d. h. wenn z. B. jeweils beide Mengen, Zeichenfolgen, Zahlen, usw. sind. Dabei muss bei Formeln zwischen der Formel an sich und dem Ergebnis der Formel unterschieden werden. Siehe Beispiel (a).

Intuitiv scheint klar zu sein, was damit gemeint ist. Wenn aber entschieden werden muss, ob z. B. (a) „1+1“ gleich „2“ oder (b) „1+1“ gleich „1 + 1“ ist, muss man erst entscheiden, von welcher Art die beiden zu vergleichenden Ausdrücke sind, d. h. *wie* verglichen wird. Wenn sie als jeweiliges Ergebnis der beiden Formeln verglichen werden, dann ist (a) richtig. Wenn sie als Formeln, d. h. als Zeichenfolgen, verglichen werden, ist (a) falsch. Wenn die Ausdrücke in (b) als Zeichenfolgen verglichen werden, dann ist (b) richtig. Wenn sie als Zeichenketten[8)] verglichen werden, ist (b) falsch.

Bei mathematischen Formeln sind Zwischenräume i. Alg. ohne Bedeutung. Daher berücksichtigen wir beim Vergleich zweier Formeln keine Zwischenräume, d. h. sie gelten als Zeichenfolgen. Die Zwischenräume dienen dann nur der besseren Lesbarkeit. Bei den Zeichenketten werden Leerstellen berücksichtigt; mehrere hintereinander sind dabei auch mehrere Zeichen.

Die folgende Tabelle fasst dass zusammen:

A	B	A gleich B
$1+1$	2	richtig
«$1+1$»	«2»	falsch
«$1+1$»	«1 + 1»	richtig
„`1+1`“	„`1 + 1`“	falsch

2.2.2.2. Vergleiche

A und B seien Objekte[9)]. Dann definieren wir:

$=$ ***Gleichheit*** «$A = B$» heißt, dass A und B sich in den interessierenden Eigenschaften für $=$ nicht unterscheiden.[10)] – „A ist *dasselbe* wie B“ oder „A ist *identisch* zu B“ – Inwieweit die Begriffe *Gleichheit* und *Identität* korrelieren, wird hier nicht erörtert. (siehe [29])

Gleichheit ist eine Äquivalenzrelation.[11)]

$\neq$ ***Ungleichheit*** «$A \neq B$» heißt, dass A und B sich in mindestens einer der interessierenden Eigenschaften für $=$ unterscheiden. „A ist *nicht dasselbe* wie B“ (aber vielleicht das gleiche; siehe $\equiv$) oder „A ist *nicht identisch* zu B“.

$\equiv$ ***Äquivalenz*(relation)** «$A \equiv B$» heißt, dass A und B sich in den interessierenden Eigenschaften für $\equiv$ nicht unterscheiden. – „A ist *das gleiche* wie B“ (aber nicht unbedingt dasselbe; siehe $=$) oder „A ist *so wie* B“.

Es kann auch mehrere Äquivalenzrelationen geben, für die dann verschiedene Bezeichnungen verwendet werden.

8) In Zeichenketten zählen Leerstellen, in Zeichenfolgen nicht.

9) siehe Unterabschnitt 2.1.2 auf Seite 12

10) Z. B. sind zwei logische Operatoren üblicherweise dann gleich, wenn sie stets denselben *Wahrheitswert* liefern. Ihre Bezeichnungen oder Symbole können dabei durchaus verschieden sein, interessieren bei der Feststellung der Gleichheit aber nicht. Andernfalls wären sie nicht gleich.

11) Eine Relation $\sim$ ist eine *Äquivalenzrelation*, wenn sie *reflexiv* ($A \sim A$), *transitiv* ($((A \sim B) \,\&\, (B \sim C)) \Rightarrow (A \sim C)$) und *symmetrisch* ($(A \sim B) \Rightarrow (B \sim A)$) ist – jeweils für alle A, B und C.

$\not\equiv$ ***Kontravalenz*** «$A \not\equiv B$» heißt, dass A und B sich in mindestens einer der interessierenden Eigenschaften für $\not\equiv$ unterscheiden. – „A ist *nicht das gleiche* wie B“ oder „A ist *nicht so wie* B“.

$=, \neq, \equiv$ und $\not\equiv$ bezeichnen wir als *Vergleichsoperatoren*.

Jede interessierende Eigenschaft für $\equiv$ oder eine andere Äquivalenzrelation muss auch eine für $=$ sein. Daraus folgt insbesondere, dass mit «$(A = B)$» auch «$(A \equiv B)$» und mit «$(A \not\equiv B)$» auch «$(A \neq B)$» gilt.

2.2.2.3. Definitionen

Seien $\overline{A}$ und $\overline{B}$ Aussagen und A und B Objekte[12)].

$:\Leftrightarrow$ ***Metadefinition*** «$\overline{A} :\Leftrightarrow \overline{B}$» heißt, dass die Aussage $\overline{A}$ *definitionsgemäß gleich* der Aussage $\overline{B}$ ist. Gewissermaßen ist $\overline{A}$ nur eine andere Schreibweise für $\overline{B}$. „$\overline{A}$ *steht für* $\overline{B}$“. $\overline{A}$ und $\overline{B}$ können sich gegenseitig ersetzten.

$:=$ ***Definition*** «$A := B$» heißt, dass das Objekt A *definitionsgemäß gleich* dem Objekt B ist. Gewissermaßen ist A nur eine andere Schreibweise für B. „A *steht für* B“. A und B können sich gegenseitig ersetzten.[13)]

Man beachte, dass $:\Leftrightarrow$ und $:=$ verschiedene Sprachebenen sind.

2.2.2.4. Prioritäten

In in der Tabelle 2.2 sind für fehlende Klammern die Prioritäten der bisher behandelten Elemente angegeben. Wenn von diesen Prioritäten abgewichen wird, wird dies eigens erwähnt.

Klammern	$(\ )\quad \langle\ \rangle\quad «\ »$
Binäre Operatoren [1]	$\bullet$
Binäre Relationen [1]	$\lhd\ \rhd\ \unlhd\ \unrhd$ $\sim\ \simeq\ \nsim$
Vergleichsoperatoren [2]	$=\ \neq\ \equiv\ \not\equiv$
Definitionen [3]	$:=$
Metaoperatoren [4]	$\&$ $\Vert$ $\Leftarrow\ \Rightarrow$ $\Leftrightarrow$
Metadefinition [3]	$:\Leftrightarrow$
Innerhalb natürlicher Sprache deren Strukturelemente, z. B. Satzzeichen[5]	. , ; usw.

[1] siehe Unterabschnitt 2.1.3 auf Seite 13
[2] siehe Paragraph 2.2.2.2 auf der vorherigen Seite
[3] siehe Paragraph 2.2.2.3
[4] siehe Unterabschnitt 2.2.1 auf Seite 14
[5] Innerhalb von Formeln können Satzzeichen eine andere Bedeutung und Priorität haben.

Tabelle 2.2.: Prioritäten in abnehmender Reihenfolge

12) Die Anforderungen an die Aussage $\overline{A}$ und das Objekt A sind intuitiv klar. Insbesondere darf $\overline{B}$ bzw. B nicht von dem bisher undefinierten Teil von $\overline{A}$ bzw. A abhängig sein.

13) Nach den Definitionen von $:\Leftrightarrow$ und $:=$ sind zwei Ausdrücke P und Q schon dann gleich, wenn nach der Ersetzung aller Vorkommen von A durch B sowohl in P als auch in Q die resultierenden Ausdrücke $\overline{P}$ und $\overline{Q}$ gleich sind.

2.3. Beweise in ASBA

Die Regeln zur Formulierung und Prüfung der Beweise müssen in ASBA fest codiert werden. Sie sind quasi die Axiome von ASBA und sollten daher möglichst wenig voraussetzen. Dazu wird ein *Genzen-Kalkül*[14] verwendet.

Ein Beweis[15] in ASBA besteht aus

einer Menge	$\mathcal{V} = \{V_n \| 0 < n \leqslant N\}$	von *Voraussetzungen* V_n
einer Menge	$\mathcal{F} = \{F_m \| 0 < m \leqslant M\}$	von *Folgerungen* F_m
einer Folge	$\mathcal{S} = (S_0, S_1, ..., S_K)$	von *Beweisschritten* S_k

wobei N, M und K Elemente von $\mathbb{N}_0$, die Voraussetzungen und Folgerungen Aussagen und die Beweisschritte *Schlussregeln*[16] sind. Mit

$$\begin{aligned} \mathcal{T}_k &:= \{S_0, S_1, ..., S_k\} \qquad \text{für } -1 \leqslant k \leqslant K \\ \mathcal{T} &:= \mathcal{T}_K \end{aligned}$$

[17] muss jeder Beweisschritt S_k für $0 \leqslant k \leqslant K$ entweder

eine Voraussetzung aus $\mathcal{V}$ oder

ein Ergebnis[18] der Anwendung einer *zulässigen Schlussregel* auf eine Teilmenge von $\mathcal{T}_{k-1}$ sein.

sein. Schließlich muss noch gelten:

$$\mathcal{F} \subseteq \mathcal{S}$$

Die damit bewiesene Aussage (z. B. ein mathematischer Satz) kann dann folgendermaßen formuliert werden:[19]

$$\frac{\mathcal{V}}{\mathcal{F}} \quad \text{bzw.} \quad \frac{V_1 \mid V_2 \mid ... \mid V_n}{F_1 \mid F_2 \mid ... \mid F_m} \qquad \text{(formaler Satz)} \tag{FS}$$

Ein formaler Satz ist gleichzeitig eine Schlussregel. Bevor die Schlussregeln behandelt werden, werden noch Elemente der *Aussagenlogik* und der *Prädikatenlogik* behandelt.

2.4. Aussagenlogik

2.4.1. Konstante und Operatoren

Die Tabelle 2.3 auf der nächsten Seite[20] definiert für die zweiwertige Logik Konstanten- und Operatorsymbole über die Wahrheitswerte ihrer Anwendung. So ergeben sich, abhängig von den Wahrheitswerten der Operanden <A> und <B>[21], die in der Tabelle angegebenen Wahrheitswerte für die Operationen. Die mit 0, 1 und 2 benannten Spalten werden jeweils nur für die 0-, 1- und 2-stelligen Operatoren, d. h. für die Konstanten, die unären und die binären Operatoren ausgefüllt. Dabei werden die Konstanten als 0-stellige Operatoren angesehen. Hat der Inhalt einer Zelle keine Relevanz, steht dort ein Minuszeichen, ist kein Wert bekannt, so bleibt sie leer.

14) siehe [1] Kapitel 1.4 und vergleiche [33, 34]
15) siehe [1] Kapitel 1.6 und 3.6
16) Schlussregeln werden später behandelt.
17) $\mathcal{T}_{-1}$ ist die leere Menge $\varnothing$
18) eine weitere zulässige Schlussregel oder eine Formel
19) | steht für „und“, bindet aber wesentlich schwächer als & .
20) Die Tabelle basiert auf den Wahrheitstafeln in [27] Kapitel 2.2 und [1] Kapitel 1.1 Seite 3.
21) <A> und <B> können hier beliebige Aussagen sein – auch Formeln –, die jeweils genau einen Wahrheitswert repräsentieren.

A	-	W	F	W	W	F	F	-	Aussage A	-
B	-	-	-	W	F	W	F	-	Aussage B	-
Junktor[1]	**0**	**1**		**2**				**Name**	**Sprechweise**[2]	**Prio**[3]
$\top$	W	-	-	-	-	-	-	Verum	Wahr	-
$\bot$	F	-	-	-	-	-	-	Falsum	Falsch	-
	-	W	W	-	-	-	-			-
$(\ldots)$	-	W	F	-	-	-	-	Klammerung	A ist geklammert	-[4]
$\neg$	-	F	W	-	-	-	-	Negation	Nicht A	1[5]
	-	F	F	-	-	-	-			-
	-	-	-	W	W	W	W	Tautologie		-
$\vee$	-	-	-	W	W	W	F	Disjunktion; Adjunktion; Alternative	A oder B	3
$\leftarrow \Leftarrow \subset$	-	-	-	W	W	F	W	Replikation; Konversion; konverse Implikation	A folgt aus B	4
$\rfloor$	-	-	-	W	W	F	F	Präpendenz	Identität von A	-
$\rightarrow \Rightarrow \supset$	-	-	-	W	F	W	W	Implikation; Subjunktion; Konditional	Wenn A so B; Aus A folgt B; A nur dann wenn B	4
$\lfloor$	-	-	-	W	F	W	F	Postpendenz	Identität von B	-
$\leftrightarrow \Leftrightarrow$	-	-	-	W	F	F	W	Äquivalenz; Bijunktion; Bikonditional	A genau dann wenn B; A dann und nur dann wenn B	5
$\wedge$ & $\cdot$	-	-	-	W	F	F	F	Konjunktion	A und B; Sowohl A als auch B	2
$\uparrow \barwedge \mid$	-	-	-	F	W	W	W	NAND; Unverträglichkeit; Sheffer-Funktion	Nicht zugleich A und B	2
$+ \dot\vee \veebar \oplus$	-	-	-	F	W	W	F	XOR; Antivalenz; ausschließende Disjunktion	Entweder A oder B	3
$\nleftrightarrow \nLeftrightarrow \not\equiv$	-	-	-	"	"	"	"	Kontravalenz		-
$\lceil$	-	-	-	F	W	F	W	Postnonpendenz	Negation von B	-
$\nrightarrow \nRightarrow \not\supset$	-	-	-	F	W	F	F	Postsektion		-
$\rceil$	-	-	-	F	F	W	W	Pränonpendenz	Negation von A	-
$\nleftarrow \nLeftarrow \not\subset$	-	-	-	F	F	W	F	Präsektion		-
$\downarrow \overline{\triangledown}$	-	-	-	F	F	F	W	NOR; Nihilation; Peirce-Funktion	Weder A noch B	3
	-	-	-	F	F	F	F	Kontradiktion		-

[1] *Operatorsymbole*. Sie stehen meistens für die Operatoren selbst. Der Einfachheit halber werden auch die beiden Konstanten bzw. ihre Symbole ‹$\top$› und ‹$\bot$› als Operatoren bzw. *Junktoren* bezeichnet.
Die Operatoren ‹$\subset$›, ‹$\supset$›, ‹$\not\subset$› und ‹$\not\supset$› haben hier nicht die Bedeutung der entsprechenden Operatoren der Mengenlehre und dürfen nicht damit verwechselt werden; entsprechendes gilt für ‹$+$› und ‹$\cdot$› mit Addition und Multiplikation.

[2] Ist eine Zelle in dieser Spalte leer, so ist die zugehörige Zeile nur vorhanden, um alle binären Operatoren aufzuführen.

[3] Je kleiner die Zahl, je höher die Priorität.

[4] Klammerung ist genau genommen keine Operation und wird nicht nur bei logischen, sondern auch bei anderen Ausdrücken verwendet. Ihre Priorität - sofern man überhaupt davon sprechen kann - kann nur höher als die aller (anderen) Operatoren sein.

[5] Die Priorität der unären Operatoren muss höher sein als die aller mehrwertigen, also auch der binären Operatoren. Wenn alle unären Operatoren auf derselben Seite des Operanden stehen, brauchen sie eigentlich keine Priorität, da die Auswertung nur von innen (dem Operanden) nach außen erfolgen kann. Nur wenn es sowohl links-, als auch rechtsseitige unäre Operatoren gibt, muss man für diese Prioritäten definieren.

Tabelle 2.3.: Definition von aussagenlogischen Symbolen.

Für einige *Junktoren*[22], Namen und Sprechweisen sind auch Alternativen angegeben. Die durchgestrichenen (d. h. negierten) Symbole sind ungebräuchlich und nur aus formalen Gründen aufgeführt. Wenn für eine bestimmte Kombination von Wahrheitswerten mehr als eine Zeile angegeben ist, so sind die zugehörigen Operationen in der zweiwertigen Aussagenlogik alle gleich. Bei der formalen Definition wird aber keine Zweiwertigkeit vorausgesetzt, so dass je nach Definition die Operationen verschiedene Ergebnisse liefern können.

Um vollständig zu sein, d. h. alle 22 möglichen Kombinationen von Wahrheitswerten für höchstens zwei Variable zu berücksichtigen, enthält die Tabelle auch viele ungebräuchliche Junktoren und Operationen. Die Zeilen mit den Klammern und den gebräuchlichsten Operatoren sind in der Tabelle grau hinterlegt. Hellgrau hinterlegt sind Zeilen mit weniger gebräuchlichen Operatoren. Die restlichen Operatoren sind uninteressant und brauchen daher keine Priorität.

2.4.2. Klammerregeln

Zur Klammerersparnis werden die üblichen Regeln verwendet, d. h. dass Operatoren mit höherer Priorität stärker binden, als solche mit niedrigerer Priorität.

Für Operatoren derselben Priorität gilt Rechtsklammerung[23]. Im Folgenden wird nur noch ein Teil der logischen Operatoren aus der Tabelle 2.3 auf der vorherigen Seite und der Metaoperatoren aus Unterabschnitt 2.2.1 auf Seite 14 berücksichtigt. Diese werden in der Tabelle 2.4 auf der nächsten Seite mit abnehmender Priorität aufgelistet.

Die Prioritäten der logischen Operatoren wurden aus [1] Kapitel 1.1 Seite 5 entnommen und ergänzt und die der Metaoperatoren daran angeglichen. Wie üblich bindet ein Operator *stärker* als jeder andere mit einer niedrigeren Priorität und *schwächer* als jeder andere mit höherer Priorität.

2.4.3. Formalisierung

Da sie die Grundlage – quasi das Fundament – des mathematischen Inhalts von ASBA sind, müssen die Axiome, Sätze, Beweise, usw. der Aussagenlogik (und später der Prädikatenlogik) in streng formaler Form vorliegen. Die Formalisierung stützt sich auf [28]; siehe auch [21, 24]. Da Computerprogramme mit der *Polnischen Notation*[24] besser umgehen können und Klammern dort überflüssig sind, werden viele Formeln auch in die Polnische Notation überführt. Dies wird auch in ASBA so gehandhabt.

2.4.3.1. Bausteine der aussagenlogischen Sprache

Zur Einteilung der aussagenlogischen Junktoren werden die folgenden Mengen definiert:

$$\mathcal{C} := \{\top, \bot\}$$, Menge der *aussagenlogischen Konstanten*

$$\mathcal{U} := \{\neg\}$$, Menge der *unären aussagenlogischen Operatoren*

$$\mathcal{B} := \{\wedge, \vee, \rightarrow, \leftrightarrow, \leftarrow, \uparrow, \downarrow, \cdot, +\}$$, Menge der *binären aussagenlogischen Operatoren*

---

22) Ein *Junktor* oder *Operatorsymbol* ist ein Symbol, dass für einen bestimmten Operator verwendet wird.

23) Unäre Operatoren stehen hier stets links *vor* dem Operanden, so dass es nur Rechtsklammerung geben kann. Zur Rechtsklammerung bei binären Operationen ein Zitat aus [1] Kapitel 1.1 Seite 5: „Diese hat gegenüber Linksklammerung Vorteile bei der Niederschrift von Tautologien in →, [...]“

24) Bei der *Polnischen Notation* wird eine zweistellige Operation $(A \bullet B)$ dargestellt als $\bullet\, AB$. Eine Zwischenstufe ist $\bullet(A, B)$, bei der noch die redundanten Gliederungszeichen Komma und Klammern – auch andere als die runden – hinzukommen, so dass die Operationen optisch besser getrennt und dadurch für Menschen besser lesbar werden. Durch einfaches Weglassen der Gliederungszeichen ergibt sich dann die Polnische Notation.

Klammern	( )
Unäre logische Operatoren	$\neg$
Binäre logische Operatoren	$\wedge$ $\cdot$ $\uparrow$ $\vee$ $+$ $\downarrow$ $\leftarrow$ $\rightarrow$ $\leftrightarrow$
Mit Gleichheit verwandte Operatoren; ihre Prioritäten untereinander sind nicht eindeutig und bleiben daher undefiniert.	$=$ $\neq$ $\equiv$ $\not\equiv$
Ableitungsrelation[1] Substitution[1] Definition	$\vdash$ $\leftarrow\!\!\prec$ $:=$
Metaoperatoren[2]	$\&$ $\|\|$ $\Leftarrow$ $\Rightarrow$ $\Leftrightarrow$
Metadefinition	$:\Leftrightarrow$
Innerhalb natürlicher Sprache deren Strukturelemente, z. B. Satzzeichen[3]	. , ; usw.

[1] siehe Unterabschnitt **??** auf Seite ??
[2] siehe Unterabschnitt 2.2.1 auf Seite 14
[3] Innerhalb von Formeln können Satzzeichen eine andere Bedeutung und Priorität haben.

Tabelle 2.4.: Prioritäten von Metaoperatoren in abnehmender Reihenfolge

Um damit Formeln zu bilden, werden noch Variable gebraucht:

$$\mathcal{Q} \quad := \quad \{q_n \mid n \in \mathbb{N}_0\}$$, Menge der aussagenlogischen *Variablen*

Die Mengen $\mathcal{C}, \mathcal{U}, \mathcal{B}$ und $\mathcal{Q}$ müssen paarweise disjunkt sein. – Damit sind alle in der Tabelle 2.3 auf Seite 18 verwendeten wesentlichen Konstanten und Operatoren[25)] und die Variablen erfasst und es können die folgende Mengen definiert werden:

$\mathcal{J} := \mathcal{C} \cup \mathcal{U} \cup \mathcal{B}$, Menge der aussagenlogischen *Junktoren*
$\mathcal{A} := \mathcal{Q} \cup \mathcal{J}$, *Alphabet der aussagenlogischen Sprache (für $\mathcal{J}$)*
$\mathcal{J}_x \subseteq \mathcal{J}$, eine Teilmenge der Junktoren für eine Indexvariable x
$\mathcal{A}_x := \mathcal{Q} \cup \mathcal{J}_x$, Alphabet der aussagenlogischen Sprache *für $\mathcal{J}_x$*

Für Elemente aus $\mathcal{Q}$ werden hier normalerweise die großen lateinischen Buchstaben $<A>$, $<B>$, $<C>$, usw. verwendet. Sie werden auch *Satzbuchstaben* oder kurz *Atome* genannt.

2.4.3.2. Aussagenlogische Formeln

Neben dem Alphabet $\mathcal{A}$ bzw. $\mathcal{A}_x$ werden noch Klammern als Gliederungszeichen verwendet. Damit können nun rekursiv für jede Teilmenge $\mathcal{J}_x$ von $\mathcal{J}$ zwei Mengen von Formeln definiert werden:

[25)] Jeweils nur die ersten der grau hinterlegten Zeilen sowie $\langle\cdot\rangle$.

$\mathcal{L}_x$ sei die Menge der auf folgende Weise definierten *aussagenlogischen Formeln mit Klammerung*:

$$\begin{array}{rcrl} & & \mathcal{Q} \subset \mathcal{L}_x & \\ & & \mathcal{J}_x \cap \mathcal{C} \subset \mathcal{L}_x & \\ A \in \mathcal{L}_x & \Rightarrow & (\bullet\, A) \in \mathcal{L}_x & \text{, für} \quad \bullet \in \mathcal{U} \cap \mathcal{J}_x \\ A, B \in \mathcal{L}_x & \Rightarrow & (A \bullet B) \in \mathcal{L}_x & \text{, für} \quad \bullet \in \mathcal{B} \cap \mathcal{J}_x \end{array}$$

Nur die auf diese Weise konstruierten Formeln sind Elemente von $\mathcal{L}_x$.
Für $\mathcal{J} = \mathcal{J}_x$ sei noch $\mathcal{L} := \mathcal{L}_x$.

$\mathcal{L}_x^p$ sei die Menge der auf folgende Weise definierten aussagenlogischen Formeln in *Polnischer Notation*:

$$\begin{array}{rcrl} & & \mathcal{Q} \subset \mathcal{L}_x^p & \\ & & \mathcal{J}_x \cap \mathcal{C} \subset \mathcal{L}_x^p & \\ A \in \mathcal{L}_x^p & \Rightarrow & (\bullet\, A) \in \mathcal{L}_x^p & \text{, für} \quad \bullet \in \mathcal{U} \cap \mathcal{J}_x \\ A, B \in \mathcal{L}_x^p & \Rightarrow & (A \bullet B) \in \mathcal{L}_x^p & \text{, für} \quad \bullet \in \mathcal{B} \cap \mathcal{J}_x \end{array}$$

Nur die auf diese Weise konstruierten Formeln sind Elemente von $\mathcal{L}_x^p$. Schließlich sei noch $\mathcal{L}^p := \mathcal{L}_x^p$ falls $\mathcal{J}_x = \mathcal{J}$.

Wie man leicht sieht, gilt

$$\mathcal{J}_x \subset \mathcal{J}_y \subseteq \mathcal{J} \Rightarrow \begin{cases} \mathcal{A}_x \subset \mathcal{A}_y \subseteq \mathcal{A} \\ \mathcal{L}_x \subset \mathcal{L}_y \subseteq \mathcal{L} \\ \mathcal{L}_x^p \subset \mathcal{L}_y^p \subseteq \mathcal{L}^p \end{cases}$$

und weiterhin gibt es eine bijektive Abbildung von $\mathcal{L}$ nach $\mathcal{L}^p$. Auf einen Beweis verzichten wir. Durch Anwendung der Klammerregeln von Paragraph 2.4.3.1 auf Seite 19 lassen sich in der Regel noch viele Klammern der Formeln aus $\mathcal{L}_x$ einsparen. Die Formeln aus $\mathcal{L}_x^p$ sind frei von Klammern. Die Namen der Operatoren finden sich in der Tabelle 2.3 auf Seite 18. Für aussagenlogische Formeln, d. h. von Elementen aus $\mathcal{L}$ bzgl. $\mathcal{L}^p$, werden hier normalerweise die kleinen griechischen Buchstaben α, β, γ, usw. verwendet. Sie können dabei auch als *atomare Formel* bezeichnet werden, d. h. Formeln, die sich nicht weiter zerlegen lassen.[26)]

2.4.4. Definition aussagenlogischer Operatoren durch andere

Im folgenden gelte für zwei aussagenlogische Formeln α und β:

$\alpha = \beta \quad :\Leftrightarrow \quad$ α und β stimmen als Zeichenkette überein.

$\alpha \equiv \beta \quad :\Leftrightarrow \quad$ α und β können mit Hilfe erlaubter Substitutionen und geltender Axiome – siehe Unterabschnitt 2.4.5 auf Seite 23 – ineinander überführt werden.

[26)] Nur die Elemente von $\mathcal{Q}$ und $\mathcal{C}$ sind unzerlegbar, sofern letztere nicht durch andere Formeln definiert werden.

Es werden verschiedene Teilmengen von $\mathcal{J}$ eingeführt, die jeweils ausreichen alle anderen Elemente aus $\mathcal{J}$ zu definieren:

$$
\begin{aligned}
\mathcal{J}_{\text{bool}} &:= \{\neg, \wedge, \vee\} \qquad (\text{Boolsche Signatur})\\
\mathcal{J}_{\text{and}} &:= \{\neg, \wedge\}\\
\mathcal{J}_{\text{or}} &:= \{\neg, \vee\}\\
\mathcal{J}_{\text{imp}} &:= \{\neg, \rightarrow\}\\
\mathcal{J}_{\text{rep}} &:= \{\neg, \leftarrow\}\\
\mathcal{J}_{\text{nand}} &:= \{\uparrow\}\\
\mathcal{J}_{\text{nor}} &:= \{\downarrow\}
\end{aligned}
$$

Solche Teilmengen heißen logische Signatur. Im Folgenden stehen jeweils links die Formeln in üblicher Schreibweise vollständig geklammert und rechts in Polnischer Notation (ohne Klammern). Ferner seien α und β beliebige, nicht notwendig verschiedene Formeln aus der passenden Menge $\mathcal{L}_x$ bzgl. der um die mit Hilfe der Definitionen erweiterten Formelmenge.

Ausgehend von den Operatoren aus der Boolschen Signatur $\mathcal{J}_{\text{bool}}$ werden die restlichen Operatoren aus $\mathcal{J}$ definiert. Die Definitionen sind in zwei Gruppen eingeteilt, und zwar die mit den Operatoren aus $\mathcal{J}_{\text{and}}$:

$$(\alpha \rightarrow \beta) := (\neg(\alpha \wedge (\neg\beta))) \qquad \rightarrow\alpha\beta := \neg\wedge\alpha\neg\beta \tag{2.4}$$

$$(\alpha \leftarrow \beta) := (\neg(\beta \wedge (\neg\alpha))) \qquad \leftarrow\beta\alpha := \neg\wedge\beta\neg\alpha \tag{2.5}$$

$$(\alpha \leftrightarrow \beta) := ((\alpha \rightarrow \beta) \wedge (\alpha \leftarrow \beta)) \qquad \leftrightarrow\alpha\beta := \wedge\rightarrow\alpha\beta\leftarrow\alpha\beta$$

$$\bot := (q_0 \wedge (\neg q_0)) \qquad \bot := \wedge q_0 \neg q_0$$

$$(\alpha \cdot \beta) := (\alpha \wedge \beta) \qquad \cdot\alpha\beta := \wedge\alpha\beta$$

$$(\alpha \uparrow \beta) := (\neg(\alpha \wedge \beta)) \qquad \uparrow\alpha\beta := \neg\wedge\alpha\beta \tag{2.6}$$

und die mit den Operatoren aus $\mathcal{J}_{\text{or}}$:

$$(\alpha \downarrow \beta) := (\neg(\alpha \vee \beta)) \qquad \downarrow\alpha\beta := \neg\vee\alpha\beta \tag{2.7}$$

$$(\alpha + \beta) := ((\alpha \vee \beta) \wedge (\neg(\alpha \wedge \beta))) \qquad +\alpha\beta := \wedge\vee\alpha\beta\neg\wedge\alpha\beta$$

$$\top := (q_0 \vee (\neg q_0)) \qquad \top := \vee q_0 \neg q_0$$

Ist ‹$\vee$› oder ‹$\wedge$› nicht vorgegeben, d. h. wird von den Elementen aus $\mathcal{J}_{\text{and}}$ bzgl. $\mathcal{J}_{\text{or}}$ statt von denen aus $\mathcal{J}_{\text{bool}}$ ausgegangen, so muss man den fehlenden Operator mittels der passenden der beiden folgenden Definitionen einführen:

$$(\alpha \vee \beta) := (\neg((\neg\alpha) \wedge (\neg\beta))) \qquad \vee\alpha\beta := \neg\wedge\neg\alpha\neg\beta$$

$$(\alpha \wedge \beta) := (\neg((\neg\alpha) \vee (\neg\beta))) \qquad \wedge\alpha\beta := \neg\vee\neg\alpha\neg\beta$$

Nun sind wieder alle Operatoren definiert.

Entsprechend wird bei Vorgabe von $\mathcal{J}_{\text{imp}}$ bzgl. $\mathcal{J}_{\text{rep}}$ die passende der beiden folgenden Definitionen ausgewählt:

$$(\alpha \vee \beta) := ((\neg\alpha) \rightarrow \beta) \qquad \vee\alpha\beta := \rightarrow\neg\alpha\beta$$

$$(\alpha \wedge \beta) := (\neg((\neg\beta) \leftarrow \alpha)) \qquad \wedge\alpha\beta := \neg\leftarrow\neg\beta\alpha$$

woraufhin dann (2.4) bzgl. (2.5) als Gleichung nachzuweisen ist. Da aus (2.5) durch Vertauschung der Variablen unmittelbar

$$(\alpha \leftarrow \beta) \equiv (\beta \rightarrow \alpha) \qquad \leftarrow\alpha\beta \equiv \rightarrow\beta\alpha$$

folgt, vermindert sich der Aufwand dazu erheblich.

Bei Vorgabe von $\mathcal{J}_{\text{nand}}$ bzgl. $\mathcal{J}_{\text{nor}}$ schließlich werden die passenden Definition aus

$$\begin{aligned} (\neg\alpha) &:= (\alpha \downarrow \alpha) & \neg\alpha &:= \downarrow\alpha\alpha \\ (\neg\alpha) &:= (\alpha \uparrow \alpha) & \neg\alpha &:= \uparrow\alpha\alpha \end{aligned}$$

und, da ‹¬› jetzt definiert ist, aus

$$\begin{aligned} (\alpha \vee \beta) &:= (\neg(\alpha \downarrow \beta)) & \vee\alpha\beta &:= \neg\downarrow\alpha\beta \\ (\alpha \wedge \beta) &:= (\neg(\alpha \uparrow \beta)) & \wedge\alpha\beta &:= \neg\uparrow\alpha\beta \end{aligned} \tag{2.8}$$

ausgewählt und es ist (2.6) bzgl. (2.7) als Gleichung nachzuweisen.

Abschließend ist noch nachzuweisen, dass mit Hilfe der jeweils passenden der Definitionen (2.4) bis (2.8), ausgehend vom jeweils passenden $\mathcal{L}_x$, genau die gesamte Formelmenge $\mathcal{L}$ erzeugt werden kann.

2.4.5. Aussagenlogisches Axiomensystem

Ausgehend von der logischen Signatur $\mathcal{J}_{\text{and}} = \{\neg, \wedge\}$ und der Definition 2.4 auf der vorherigen Seite von ‹→› werden die folgenden vier logischen Axiome definiert:

$$\begin{aligned} &(\alpha \to \beta \to \gamma) \to (\alpha \to \beta) \to (\alpha \to \gamma) && \to\to\alpha\to\beta\gamma\to\to\alpha\beta\to\alpha\gamma \\ &\alpha \to \beta \to \alpha \wedge \beta && \to\alpha\to\beta\wedge\alpha\beta \\ &\alpha \wedge \beta \to \alpha\,; \quad \alpha \wedge \beta \to \beta && \to\wedge\alpha\beta\alpha\,; \quad \to\wedge\alpha\beta\beta \\ &(\alpha \to \neg\beta) \to (\beta \to \neg\alpha) && \to\to\alpha\neg\beta\to\beta\neg\alpha \end{aligned}$$

> > > Aussagenlogik weiter bearbeiten. < < <

2.5. Prädikatenlogik

> > > Prädikatenlogik bearbeiten. < < <

2.6. Mengenlehre

> > > Mengenlehre bearbeiten. < < <

3. Design

Dieses Projekt soll Open Source sein. Daher gilt für die Dokumente die *GNU Free Documentation License (FDL)* und für die Software die *GNU Affero General Public License (APGL)*. Die *GNU General Public License (GPL)* reicht für die Software nicht aus, da das Programm auch mittels eines Servers betrieben werden kann und soll. Damit das Projekt gegebenenfalls durch verschiedene Entwickler gleichzeitig bearbeitet werden kann und wegen des Konfigurationsmanagements wurde es als ein GitHub Projekt erstellt (siehe [20]).

Wenn die Lizenzen nicht mitgeliefert wurden, können sie unter `http://www.gnu.org/licenses/` gefunden werden.

3.1. Anforderungen

Die Anforderungen ergeben sich zunächst aus den Zielen im Abschnitt 1.3 auf Seite 5. Die beiden Ziele 1 *Daten* und 15 *Lizenz* sind für die Entwicklung von ASBA von sekundärer Bedeutung und werden daher in diesem Abschnitt nicht übernommen. Die anderen Ziele werden noch verfeinert.

> > > Ziele aus Abschnitt "Ziele" in Anforderungen umwandeln. < < <

1. *Form*: Die Daten liegt in formaler, geprüfter Form vor. (siehe Ziel 2 auf Seite 5)
2. *Eingaben*: Die Eingabe von Daten erfolgt in einer formalen Syntax unter Verwendung der üblichen mathematischen Schreibweise. Folgende Daten können eingegeben werden:
   a) Axiome
   b) Sätze
   c) Beweise
   d) Fachbegriffe
   e) Fachgebiete
   f) Ausgabeschemata

   Dabei sind alle Begriffe nur innerhalb eines Fachgebiets und seiner untergeordneten Fachgebiete gültig, solange sie nicht umdefiniert werden. Das oberste Fachgebiet ist die ganze Mathematik. – siehe Ziel 3 auf Seite 5
3. *Prüfung*: Vorhandene Beweise können automatisch geprüft werden. – siehe Ziel 4 auf Seite 5
4. *Ausgaben*: Die Ausgabe kann in einer eindeutigen, formalen Syntax gemäß vorhandener Ausgabeschemata erfolgen. – siehe Ziel 5 auf Seite 6
5. *Auswertungen*: Zusätzlich zur Ausgabe der Daten sind verschiedene Auswertungen möglich. Insbesondere kann zu jedem Beweis angegeben werden, wie lang er ist und welche Axiome und Sätze[1] er benötigt. – siehe Ziel 6 auf Seite 6

1) Sätze, die quasi als Axiome verwendet werden.

6. *Anpassbarkeit*: Fachbegriffe und die Darstellung bei der Ausgabe können mit Hilfe von – gegebenenfalls unbenannten – untergeordneten Fachgebieten angepasst werden. – siehe Ziel 7 auf Seite 6
7. *Individualität*: Axiom und Sätze können für jeden Beweis individuell vorausgesetzt werden. Dabei sind fachgebietsspezifische Fachbegriffe erlaubt. – siehe Ziel 8 auf Seite 6)
8. *Internet*: Die Daten können auf mehrere Dateien verteilt sein. Ein Teil davon – oder sogar alle – können im Internet liegen. – siehe Ziel 9 auf Seite 6
9. *Kommunikation*: Die Kommunikation mit ASBA kann mit den Fachbegriffen der einzelnen Fachgebiete erfolgen. – siehe Ziel 10 auf Seite 6
10. *Zugriff*: Der Zugriff auf ASBA kann lokal und über das Internet erfolgen. – siehe Ziel 11 auf Seite 6
11. *Unabhängigkeit*: ASBA kann offline und online arbeiten. – siehe Ziel 12 auf Seite 6
12. *Rekursion*: Es kann rekursiv über alle verwendeten Dateien – auch solchen, die im Internet liegen – ausgewertet werden. – siehe Ziel 13 auf Seite 6
13. *Bedienbarkeit*: ASBA ist einfach zu bedienen. – siehe Ziel 14 auf Seite 6

3.2. Axiome

> > > Axiome auswählen und definieren. < < <

3.3. Beweise

> > > Schlussregeln auswählen und Beweise definieren. < < <

3.4. Datenstruktur

> > > Datenstruktur abstrakt und in XML definieren. < < <

3.5. Bausteine

> > > Bausteine? definieren. < < <

A. Anhang

A.1. Werkzeuge

Da dies ein Open Source Projekt sein soll, müssen alle Werkzeuge, die zum Ablauf der Software erforderlich sind, ebenfalls Open Source sein. Für die reine Entwicklung sollte das auch gelten, muss es aber nicht.

Werkzeuge zur Übersetzung der Quelldateien

1. Ein Übersetzer für LaTeXQuellcode (*.tex). – Verwendet wird *MiKTeX*.
2. Ein Übersetzer für C++ Quellcode (*.c, *.cpp, *.h, *.hpp). – Verwendet wird *Visual Studio Community 2017*.

Nicht unbedingt nötig, aber sinnvoll:

3. Ein Dokumentationssystem für in C++ Quellcode und darin enthaltene Doxygen Kommentare (*.c, *.cpp, *.h, *.hpp). – Verwendet wird *Doxygen* mit Konfigurationsdatei „Doxyfile“.
4. Ein Konfigurationsmanagementsystem zur Verwaltung der Quelldateien. – Verwendet wird *GitHub*.

Werkzeuge für die Entwicklung

5. *GitHub* als Online Konfigurationsmanagementsystem zur Zusammenarbeit verschiedener Entwickler. → `https://github.com/` – Lizenz siehe [7]
6. GitHub benötigt *Git* als Konfigurationsmanagementsystem. → `https://git-scm.com/` – Lizenz siehe [7]
7. *MiKTeX* für Dokumentation und Ausgaben in LaTeX. → `https://miktex.org/` – Lizenz siehe [11]
8. angedacht: *Visual Studio Community 2017*[1)] (*VS*) als Entwicklungsumgebung für C++. → `https://www.visualstudio.com/downloads/` – Lizenz siehe [10]
9. angedacht: In *Visual Studio Community 2015* integrierte Datenbank für Axiome, Sätze, Beweise, Fachbegriffe und Fachgebiete. – Lizenz siehe [10]
10. angedacht: *RapidXml* für Ein- und Ausgabe in XML. → `http://rapidxml.sourceforge.net/index.htm` – Lizenz siehe wahlweise [3] oder [13] [2)]
11. angedacht: *Doxygen* als Dokumentationssystem für C++. → `http://www.stack.nl/~dimitri/doxygen/` – Lizenz siehe [7]
12. angedacht: Doxygen benötigt *Ghostscript* als Interpreter für Postscript und PDF. → `http://ghostscript.com/` – Lizenz siehe [5]

1) Visual Studio Community ist zwar nicht Open Source, darf aber zur Entwicklung von Open Source Software unentgeltlich verwendet werden.

2) RapidXml stellt eine C++ Header-Datei zur Verfügung. Wenn diese im Quellcode eines Programms enthalten ist, gilt das ganze Programm als Open Source. Wenn diese Header-Datei nur in einer Bibliothek innerhalb eines Projekts verwendet wird, so gilt nur diese Bibliothek als Open Source.

13. angedacht: Doxygen benötigt *Graphviz* mit *Dot* zur Erzeugung und Visualisierung von Graphen. → http://www.graphviz.org/Home.php – Lizenz siehe [4]

Werkzeuge zur Bearbeitung der Quelldateien

14. *TEXstudio* als Editor für LaTeX. → http://www.texstudio.org/ – Lizenz siehe [7]
TEXstudio benötigt einen Interpreter für Perl:

15. *Strawberry Perl* als Interpreter für Perl. → http://strawberryperl.com/ – Lizenz: Various OSI-compatible Open Source licenses, or given to the public domain

16. *Notepad*++ als Text-Editor. → https://notepad-plus-plus.org/ – Lizenz siehe [6]

17. *WinMerge* zum Vergleich von Dateien und Verzeichnissen. → http://winmerge.org/ – Lizenz siehe [6]

A.2. Offene Aufgaben

1. TODOs bearbeiten.
2. Eingabeprogramm erstellen (liest XML).
3. Prüfprogramm erstellen.
4. Ausgabeprogramm erstellen (schreibt XML).
5. Formelausgabe erstellen (erzeugt LaTeX aus XML).
6. Axiome sammeln und eingeben.
7. Sätze sammeln und eingeben.
8. Beweise sammeln und eingeben.
9. Fachbegriffe und Symbole sammeln und eingeben.
10. Fachgebiete sammeln und eingeben.
11. Ausgabeschemata sammeln und eingeben.

B. Verzeichnisse

Tabellenverzeichnis

Abbildungsverzeichnis

Literaturverzeichnis

[1] Wolfgang Rautenberg, *Einführung in die Mathematische Logik*: Ein Lehrbuch, 3. Auflage, Vieweg+Teubner 2008

[2] *Apache License*, Version 2.0 →[1] http://www.apache.org/licenses/LICENSE-2.0 02.01.2004[2]

[3] *Boost Software License* 1.0 → http://www.boost.org/users/license.html 17.08.2003

[4] *Eclipse Public License* Version 1.0 → http://www.eclipse.org/org/documents/epl-v10.php 09.03.2017

[5] *GNU Affero General Public License* → http://www.gnu.org/licenses/agpl 19.11.2007

[6] *GNU General Public License* → http://www.gnu.org/licenses/old-licenses/gpl-1.0 02.1989

[7] *GNU General Public License*, Version 2 → http://www.gnu.org/licenses/old-licenses/gpl-2.0 06.1991

[8] *GNU Lesser General Public License*, Version 2.1 → http://www.gnu.org/licenses/old-licenses/lgpl-2.1 02.1999

[9] Lizenz für *Clover* → https://www.atlassian.com/software/clover 2017

[10] Lizenz für *Microsoft Visual Studio Express 2015* → https://www.visualstudio.com/de/license-terms/mt171551/ 2017

[11] Lizenz für *MikTeX* → https://miktex.org/kb/copying 14.01.2014

[12] Lizenz für *SAX* → http://www.saxproject.org/copying.html 05.05.2000

[13] *MIT License* → https://opensource.org/licenses/MIT/ 09.03.2017

[14] *Oracle Binary Code License Agreement* → http://java.com/license 02.04.2013

[15] *OSI Certified Open Source Software* → https://opensource.org/pressreleases/certified-open-source.php 16.06.1999

[16] *W3C Document License* → http://www.w3.org/Consortium/Legal/2015/doc-license 01.02.2015

[17] *W3C Software Notice and License* → http://www.w3.org/Consortium/Legal/2002/copyright-software-20021231.html 13.05.2015

[18] *Hilbert II – Introduction* → http://www.qedeq.org/ 20.01.2014

[19] *Formal Correct Mathematical Knowledge*: GitHub Repository vom Projekt Hilbert II → https://github.com/m-31/qedeq/ 04.08.2016

[20] *ASBA – Axiome, Sätze, Beweise und Auswertungen*. Projekt zur maschinellen Überprüfung von mathematischen Beweisen und deren Ausgabe in lesbarer Form: GitHub Repository vom Projekt ASBA – in Bearbeitung → https://github.com/Dr-Winfried/ASBA

[1] Der Pfeil (→) verweist stets auf einen Link zu einer Seite im Internet.

[2] Das Datum hinter dem Link gibt – je nachdem welches bekannt ist – das Datum der letzten Änderung, den Stand der Seite oder das Datum, an dem die Seite angeschaut wurde an. Sind mehrere Daten vorhanden, wird das erste vorhandene in der angegebenen Reihenfolge genommen. – Dies gilt für alle hier aufgelisteten Seiten im Internet.

[21] Meyling, Michael: *Anfangsgründe der mathematischen Logik*
→ http://www.qedeq.org/current/doc/math/qedeq_logic_v1_de.pdf 24. Mai 2013 (in Bearbeitung)

[22] Meyling, Michael: *Formale Prädikatenlogik*
→ http://www.qedeq.org/current/doc/math/qedeq_formal_logic_v1_de.pdf 24. Mai 2013 (in Bearbeitung)

[23] Meyling, Michael: *Axiomatische Mengenlehre*
→ http://www.qedeq.org/current/doc/math/qedeq_set_theory_v1_de.pdf 24. Mai 2013 (in Bearbeitung)

[24] Meyling, Michael: *Elements of Mathematical Logic*
→ http://www.qedeq.org/current/doc/math/qedeq_logic_v1_en.pdf 24. Mai 2013 (in Bearbeitung)

[25] Meyling, Michael: *Formal Predicate Calculus*
→ http://www.qedeq.org/current/doc/math/qedeq_formal_logic_v1_en.pdf 24. Mai 2013 (in Bearbeitung)

[26] Meyling, Michael: *Axiomatic Set Theory*
→ http://www.qedeq.org/current/doc/math/qedeq_set_theory_v1_en.pdf 24. Mai 2013 (in Bearbeitung)

[27] Wikipedia: *Aussagenlogik* Kapitel 2.2 *Mögliche Junktoren*
→ https://de.wikipedia.org/wiki/Junktor#M.C3.B6gliche_Junktoren 20.01.2016

[28] Wikipedia: *Aussagenlogik* Kapitel 4 *Formaler Zugang*
→ https://de.wikipedia.org/wiki/Aussagenlogik#Formaler_Zugang 13.02.2017

[29] Wikipedia: *Identität (Logik)* Kapitel 2.3 *Identität in der Informatik* → https://de.wikipedia.org/wiki/Identit%C3%A4t_(Logik)#Identit.C3.A4t_in_der_Informatik 18.05.2017

[30] Wikipedia: *Kalkül* → https://de.wikipedia.org/wiki/Kalk%C3%BCl 26.02.2017

[31] Wikipedia: *Mengenlehre* → https://de.wikipedia.org/wiki/Mengenlehre 03.03.2017

[32] Wikipedia: *Prädikatenlogik erster Stufe*
→ https://de.wikipedia.org/wiki/Pr%C3%A4dikatenlogik_erster_Stufe 24.02.2017

[33] Wikipedia: *Schlussregel* → https://de.wikipedia.org/wiki/Schlussregel 01.05.2017

[34] Wikipedia: *Natürliches Schließen*
→ https://de.wikipedia.org/wiki/Systeme_nat%C3%BCrlichen_Schlie%C3%9Fens 01.05.2017

Index

Symbolverzeichnis

Glossar

(**FS**) *formaler Satz*. 32

$\mathcal{A}$ Das Alphabet der aussagenlogischen Sprache. 20, 32

$\mathcal{A}_x$ Eine Teilmenge des Alphabets $\mathcal{A}$ der aussagenlogischen Sprache. 20, 32

$\mathcal{B}$ Die Menge der aussagenlogischen, binären Operatoren. 19, 32

$\mathcal{C}$ Die Menge der aussagenlogischen Konstanten. 19, 32

$\mathcal{L}_x$ Eine Teilmenge der Menge $\mathcal{L}$ der aussagenlogischen Formeln mit Klammerung. 21, 32

$\mathcal{L}_x^p$ Eine Teilmenge der Menge $\mathcal{L}$ der aussagenlogischen Formeln in polnischer Notation. 21, 32

$\mathcal{J}$ Die Menge der aussagenlogischen Junktoren (Operatorsymbole). 20, 32

$\mathcal{J}_x$ Eine Teilmenge der Menge $\mathcal{J}$ der aussagenlogischen Operatoren . 20, 32

$\mathcal{U}$ Die Menge der aussagenlogischen unären Operatoren. 19, 32

$\mathcal{Q}$ Die Menge der aussagenlogischen Variablen. 20, 32

$:=$ Ein *Metaoperator*: ... definitionsgemäß gleich ... 16, 32

$=$ Ein *Metaoperator*: ... gleich (ist dasselbe wie, ist identisch zu) ... 15, 32

$\equiv$ Ein (Meta-)Operator: ... äquivalent (ist das gleiche wie, ist so wie) zu ... 15, 32

$\perp$ Eine Aussagenlogische Konstante: Falsch. 18, 32

$\lhd$ Ein Beispielsymbol für eine Relation mit Umkehrrelation $\rhd$. 13, 32

$\top$ Eine Aussagenlogische Konstante: Wahr. 18, 32

& Ein *Metaoperator*: ... und ... Die Priorität ist höher als die von ‹|›. 14, 32

$:\Leftrightarrow$ Ein *Metaoperator*: ... definitionsgemäß gleich (definitionsgemäß genau dann, wenn) ... 16, 32

$\Leftrightarrow$ Ein *Metaoperator*: ... genau dann wenn ... 14, 32

$\Rightarrow$ Ein *Metaoperator*: ... dann auch ... 14, 32

$||$ Ein *Metaoperator*: ... oder ... 14, 32

$\Leftarrow$ Ein *Metaoperator*: ... sofern ... 14, 32

$\neq$ Ein *Metaoperator*: ... ungleich (nicht dasselbe wie, nicht identisch zu) ... 15, 32

$\not\equiv$ Ein (Meta-)Operator: ... nicht äquivalent (ist nicht das gleiche wie, ist nicht so wie) ... 16, 32

$\sim$ Ein Beispielsymbol für eine Relation. 13, 32

$\simeq$ Ein Beispielsymbol für eine Relation mit Gleichheit. 13, 32

$\nsim$ Verneinung von $\sim$. 13, 32

$\rhd$ Ein Beispielsymbol für eine Relation mit Umkehrrelation $\lhd$. 13, 32

Ableitungsrelation Die Relation ‹ $\vdash$ ›. 20, 31

ASBA Programmsystem, das **A**xiome, **S**ätze, **B**eweise und **A**uswertungen behandeln kann. 4–7, 9, 10, 12, 17, 19, 24, 25, 28, 31

atomare Formel Eine Formel, die sich nicht weiter zerlegen lässt. 21, 31

Ausgabeschema Ein Schema, mit dem bestimmte mathematische *Objekte* ausgegeben werden sollen. 5, 6, 9, 10, 24, 31

Aussage Eine Aussage in natürlicher Sprache oder als Formel, die einen *Wahrheitswert* liefert. 13, 14, 16–18, 31

Aussagenlogik siehe Abschnitt 2.4 auf Seite 17. 17, 19, 31

Axiom Eine Formel, die unbewiesen als wahr angesehen wird. 4–6, 8–12, 17, 19, 21, 23, 24, 26, 31

Beweis Eine zulässige Ableitung von Folgerungen aus gegebenen Voraussetzungen. 4–6, 8–12, 17, 19, 21, 24, 26, 31

Beweisschritt Eine Vorschrift, wie aus vorgegebenen *Aussagen* (den *Voraussetzungen*) eine weitere (die *Folgerung*) folgt. 12, 17, 31

Boolsche Signatur Die *logische Signatur* $\{\neg, \wedge, \vee\}$. 22, 31

Fachbegriff Ein Name für einen mathematischen Begriff. 4–6, 10, 24, 26, 31

Fachgebiet Ein Teil der Mathematik mit einer zugehörigen Basis aus Axiomen, Sätzen und spezifischen Fachbegriffen und Darstellungen. 4–6, 10, 24, 26, 31

Folgerung Die Folgerungen einer *Schlussregel* sind die *Aussagen* über ihrem Querstrich. . 17, 31

formaler Satz Formale Darstellung eines mathematischen Satzes – siehe *FS*. 17, 31

Formel Unter einer *Formel* verstehen wir in diesem Dokument stets eine mathematische Formel. Diese kann auch mehrdimensional sein, lässt sich aber mittels geeigneter Definitionen immer eindeutig als eine Zeichenfolge schreiben. 12–17, 19–22, 31

interessierende Eigenschaft Solche Eigenschaften von *Objekten*, die im aktuellen Zusammenhang von Interesse sind. 15, 16, 31

logische Signatur Eine Teilmengen von $\mathcal{J}$, die ausreicht, alle anderen Elemente aus $\mathcal{J}$ zu definieren. 22, 31

Metaoperator Ein Operator der Metasprache, dargestellt duch ein Symbol: $\Rightarrow$, $\Leftarrow$, $\Leftrightarrow$, &, || und |. 14, 16, 19, 20, 31

Metasprache Eine Sprache, in der Aussagen über Elemente einer anderen Sprache getroffen werden können. In diesem Dokument ist dies immer die normale Sprache. siehe Abschnitt 2.2 auf Seite 13. 13, 31

Objekt Symbole, Formeln und Aussagen sowie Mengen, Zeichenfolgen, Zahlen, ganz allgemein reale oder Gedachte Dinge an sich. 12, 13, 15, 16, 31, 35

Prädikatenlogik siehe Abschnitt 2.5 auf Seite 23. 17, 19, 31

Satz Eine mathematische Aussage, dass eine bestimmte Folgerung aus gegebenen Voraussetzungen abgeleitet werden kann. 4–6, 9–12, 17, 19, 24, 26, 31

Schlussregel Eine Regel für eine (zulässige) Umwandlung von Formeln. 17, 31

Substitution Die Ersetzung von einem, mehreren oder allen *formalen Elementen* (α) in einem anderen *formalen Element* (γ) durch ein drittes *formales Element* (β) – formal: $\gamma(\alpha \leftharpoondown \beta)$. Wenn alle α in γ durch β ersetzt werden, ist die *Substitution vollständig*. (siehe Unterabschnitt **??** auf Seite ??). 20, 21, 31

vergleichbar Zwei Objekte A und B sind *vergleichbar*, wenn beide von derselben Art sind, d. h. wenn z. B. jeweils beide Mengen, Zeichenfolgen, Zahlen, usw. sind. Dabei muss bei Formeln zwischen der Formel an sich und dem Ergebnis der Formel unterschieden werden. – siehe Abschnitt **??** auf Seite ?? . 15, 31, 35

Vergleichsoperator Eine mit der Gleichheit verwandte Relation: $=$, $\neq$, $\equiv$ und $\not\equiv$. 16, 31

Voraussetzung Die Voraussetzungen einer *Schlussregel* sind die *Aussagen* über ihrem Querstrich. . 17, 31

Wahrheitswert Wahrheitswerte sind die Werte ‹wahr› und ‹falsch›, oft auch als ‹true› und ‹false› oder einfach ‹1› und ‹0› bezeichnet. 15, 17, 19, 31