

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

25. Juni 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.

Copyright © 2017 Winfried Teschers

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. You should have received a copy of the GNU Free Documentation License along with this document. If not, see <http://www.gnu.org/licenses/>.

Inhaltsverzeichnis

1. Analyse	4
1.1. Fragen	4
1.2. Eigenschaften	5
1.3. Ziele	6
1.4. Zusammenfassung	7
1.5. Umgebung von ASBA	8
1.6. Basis von Beweisen	9
2. Mathematische Grundlagen	11
2.1. Metasprache	11
2.1.1. Metasprachliche Ausdrücke	11
2.1.2. Mit Gleichheit verwandte Symbole	12
2.1.2.1. Allgemeine Voraussetzungen	12
2.1.2.2. Definition der mit Gleichheit verwandten Symbole	12
2.2. Formale Elemente	13
2.3. Schlussregeln	14
2.3.1. Basisregeln	14
2.3.2. Identitätsregeln	16
2.3.3. Weitere Schlussregeln	16
2.3.4. Beispiel einer Ableitung	17
2.4. Aussagenlogik	18
2.4.1. Konstante und Operatoren	18
2.4.2. Klammerregeln	18
2.4.3. Formalisierung	18
2.4.3.1. Bausteine der aussagenlogischen Sprache	20
2.4.3.2. Aussagenlogische Formeln	21
2.4.4. Definition aussagenlogische Operatoren durch andere	21
2.4.5. Aussagenlogisches Axiomensystem	23
2.5. Prädikatenlogik	23
2.6. Mengenlehre	23
3. Design	24
3.1. Anforderungen	24
3.2. Axiome	25
3.3. Beweise	25
3.4. Datenstruktur	25
3.5. Bausteine	25
A. Anhang	26
A.1. Werkzeuge	26
A.2. Offene Aufgaben	27
B. Verzeichnisse	28
Tabellenverzeichnis	28
Abbildungsverzeichnis	28
Literaturverzeichnis	29

Index	31
Symbolverzeichnis	32
Glossar	33

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 **Axiome, Sätze, Beweise** und **Auswertungen** 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 hier 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?
11. *Forschung*: Welche Probleme gibt es noch zu erforschen.

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 und spezifischen 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.

1.2. Eigenschaften

Ausgehend von den Fragen in Abschnitt 1.1 auf der vorherigen Seite soll ASBA entwickelt werden, das die folgenden Eigenschaften hat:

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 zeigt, wie sich die Eigenschaften zu den Fragen in 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.

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

1.3. Ziele

Um die Eigenschaften von Abschnitt 1.2 auf der vorherigen Seite zu erreichen, werden für ASBA die folgenden Ziele³⁾ gesetzt:

1. *Daten*: Es enthält möglichst viele wichtige Axiome, Sätze, Beweise, Fachbegriffe, Fachgebiete und Ausgabeschemata⁴⁾.
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.
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 viele Beweisschritte und welche Axiome und Sätze⁵⁾ er verwendet.
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.

Die Tabelle 1.2 auf der nächsten Seite zeigt wieder, wie sich die Ziele zu den Eigenschaften in Abschnitt 1.2 auf der vorherigen Seite 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.

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 auf der vorherigen Seite erfüllen zu können, werden noch fachgebietsspezifische Ausgabeschemata benötigt, welche die Art der Ausgaben beschreiben.

⁵⁾ Sätze, die quasi als Axiome verwendet werden.

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

Frage \ Ziel	Ziel	Daten	Form	Eingaben	Prüfung	Ausgaben	Auswertungen	Anpassbarkeit	Individualität	Internet	Kommunikation	Zugriff	Unabhängigkeit	Rekursion	Bedienbarkeit	Lizenz
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
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

Die Tabelle 1.3 ist eine Kombination aus den Tabellen 1.1 auf Seite 5 und 1.2 und zeigt, wie sich die Ziele in Abschnitt 1.3 auf der vorherigen Seite zu den Fragen in 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.

1.5. 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.

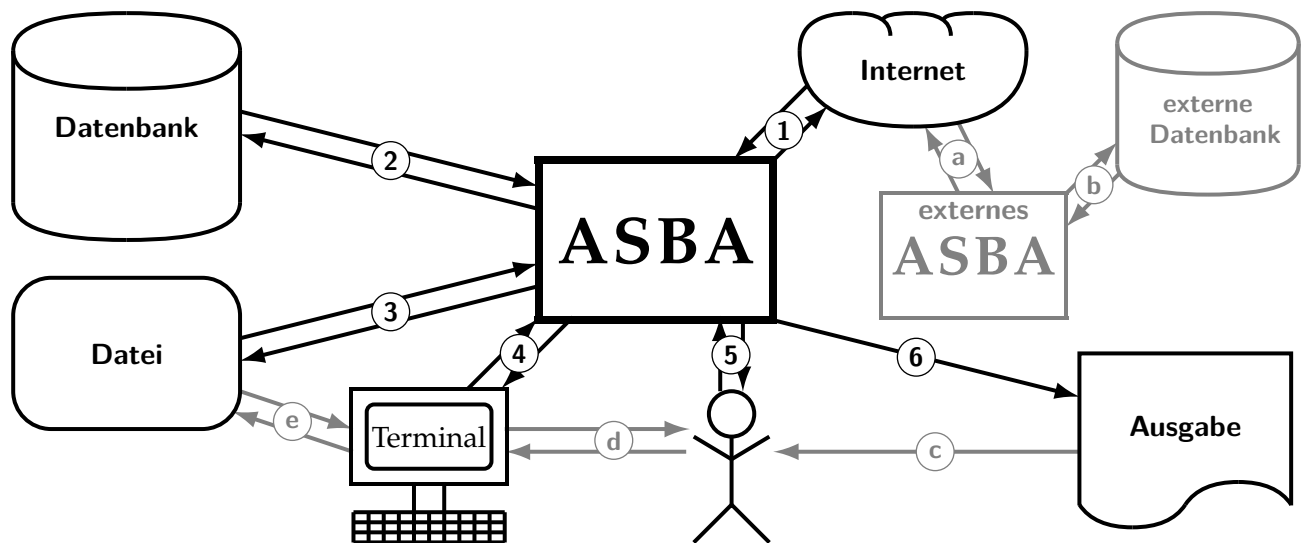


Abbildung 1.1.: Die Umgebung von ASBA

In den in 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, Anzahl der Beweisschritte 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 \rightarrow **externes ASBA** Inhalte der externen Datenbank.

externes ASBA \rightarrow **externe Datenbank** Inhalte der Datenbank.

(c) **Ausgabe** \rightarrow **Anwender** Alle Daten der Ausgabe.

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

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

Die Datenflüsse (a) bis (e) erfolgen außerhalb von ASBA und 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 Beweise.

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 Prädikat in einem bestimmten Fachgebiet.

Fachgebiete Ein Fachgebiet ist ein Teil der Mathematik mit einer zugehörigen Basis von Axiomen, Sätzen, Fachbegriffen 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 logische 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.

⁶⁾ siehe [30]

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

Zeichen oder Buchstaben, aus denen diese Zeichenketten bestehen dürfen, müssen definiert werden. Außerdem Regeln, wie aus diesen Buchstaben Zeichenketten (Formeln) gebildet werden dürfen. Gebraucht werden also:

- *Bausteine*, also Grundelemente, auch (Satz-)Buchstaben genannt, aus denen komplexe Ausdrücke (Zeichenketten) zusammengesetzt werden können.
- *Formationsregeln*, mit denen festgelegt wird, wie man aus den Bausteinen komplexe Ausdrücke erzeugen kann.

Sätze lassen sich als eine Menge von Formeln (Zeichenketten), 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 von 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 2.3 auf Seite 14), andererseits dienen sie auch zum Testen von ASBA. Daher behandelt dieses Kapitel die mathematischen Grundlagen viel ausführlicher, als für die Erstellung von ASBA erforderlich ist. Alle hier aufgeführten Axiome, Sätze und Beweise sollen dazu kodiert und die Beweise von ASBA verifiziert werden.

2.1. Metasprache

Wenn man über eine Sprache spricht, braucht man auch eine Sprache, in der Aussagen über die erstere getroffen werden können. Wenn die zuerst genannte Sprache die der Mathematik ist, nimmt man üblicherweise die natürliche Sprache als Metasprache. Leider ist diese oft ungenau, nicht immer eindeutig und abhängig vom Zusammenhang, in dem sie gesprochen wird¹⁾. Um diese Probleme in den Griff zu bekommen, wird die Metasprache zum Teil formalisiert. Durch diese Formalisierung erinnert sie dann teilweise schon an mathematische Formeln. Die Sprachebenen sollten aber sorgfältig unterschieden werden.

2.1.1. Metasprachliche Ausdrücke

Ein *metasprachlicher Ausdruck* ist eine in normaler Sprache verfasste Aussage, wie z. B. (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. In einem erweiterten Sinne gehören auch Relationen einschließlich ihrer Operanden dazu²⁾, wie z. B. „ $A = A$ “, „ $A \equiv B$ “, „ $A < B$ “, usw.

Während die Beispiele (a) und (b) einfache, nicht mehr zerlegbare metasprachliche Ausdrücke sind, ist Beispiel (c) zusammengesetzt. Für alle drei Aussagen lässt sich feststellen, ob sie richtig sind oder nicht. Das kann man für den zweiten Teil von (c) aber nicht, wenn man nicht weiss worauf sich „das“ bezieht. Natürlich muss auch der Zusammenhang, in dem ein metasprachlicher Ausdruck formuliert wird, bekannt sein, denn z. B. ist die Bedeutung von „Ich“ nur dann bekannt, wenn man weiss von wem die Aussage ist. Auf eine exakte Definition von „metasprachlicher Ausdruck“ wird verzichtet, weil das intuitive Verständnis hier ausreicht. In erster Näherung können aber alle sprachlichen Ausdrücke, die im Prinzip überprüft werden können, als metasprachliche Ausdrücke betrachtet werden.

Zusammengesetzte metasprachliche Ausdrücke wie (c) können zum Teil formalisiert werden. Dies wird mit den folgenden Definitionen erreicht:

¹⁾ 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“.

²⁾ Wird statt des Symbols der Name der zugehörigen Relation verwendet, ist dies unmittelbar einleuchtend. So wird z. B. aus der Formel „ $A < B$ “ die metasprachliche Aussage „A ist kleiner als B“.

$A \Rightarrow B$	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]“.
$A \&\& B$	steht für „[Es gilt] A und B “.
$A \parallel B$	steht für „[Es gilt] A oder B “.

Bei den Schlussregeln³⁾ wird $'|'$ statt $'\&\&'$ bei unterschiedlicher Priorität⁴⁾ verwendet.

$A B$	steht für „ A und B “.
---------	----------------------------

Offensichtlich sind das alles ebenfalls metasprachliche Ausdrücke, jetzt aber teilweise formalisiert. (c) lässt sich dann ausdrücken als „Ich habe ein rotes Auto' && ,das kann 200 km/h schnell fahren.““.

Um Verwechslungen mit den logischen Symbolen zu vermeiden, werden für „und“ und „oder“ die Symbole $'\&\&'$ und $'||'$ verwendet. A und B können als Operanden von $'\Leftrightarrow'$, $'\&\&'$, $'||'$ und $'|'$ ohne Bedeutungsveränderung vertauscht werden. Wird in einer Formel nur einer der Operatoren $'\&\&'$, $'||'$ oder $'|'$ verwendet, können die Operanden beliebig permutiert werden, so dass dann auch eine Klammerung überflüssig ist. – Ein Symbol für „nicht“ wird hier nicht gebraucht.

Metasprachliche Ausdrücke können auch geklammert werden, um die Reihenfolge der Auswertung eindeutig zu machen. $'\Rightarrow'$, $'\Leftarrow'$, $'\Leftrightarrow'$, $'\&\&'$, $'||'$ und $'|'$ heißen *metasprachliche Operatoren*. Ihre Prioritäten werden im Unterabschnitt 2.4.2 auf Seite 18 zusammen mit anderen Operatoren definiert.

Sollen zwei metasprachliche Ausdrücke miteinander verglichen werden, muss klar sein auf welche Art; ob z. B. als Zeichenfolgen – mit oder ohne Wertung der Zwischenräume –, als Wahrheitswerte oder auf sonstige Art. Wenn die Art des Vergleichs implizit oder explizit klar ist und sich die beiden Ausdrücke auf diese Art vergleichen lassen, heißen sie *vergleichbar*.

2.1.2. Mit Gleichheit verwandte Symbole

2.1.2.1. Allgemeine Voraussetzungen

In diesem und allen weiteren Abschnitten wird vorausgesetzt:

- Wenn mehrere der im Folgenden definierten Operatoren $'\Leftrightarrow'$, $'\Rightarrow'$, $'\Leftarrow'$, $'\neq'$ und $'\equiv'$ verwendet werden, dann im selben Zusammenhang⁵⁾.
- Soweit verwendet sind die *interessierenden Eigenschaften* für $'\Rightarrow'$ und $'\equiv'$ bekannt. Dabei muss jede interessierende Eigenschaft für $'\equiv'$ auch eine für $'\Rightarrow'$ sein.
- Soweit verwendet sind die jeweiligen Operanden von $'\Rightarrow'$ und $'\equiv'$ vergleichbar.

2.1.2.2. Definition der mit Gleichheit verwandten Symbole

Unter den Voraussetzungen von Paragraph 2.1.2.1 werden die folgenden (metasprachlichen) Operatoren definiert:

= **Gleichheit** „ $A = B$ “ heißt, dass A und B sich in den interessierenden Eigenschaften für $'\Rightarrow'$ nicht unterscheiden.⁶⁾ – „ A ist dasselbe wie B “ oder „ A ist identisch zu B “ – Inwieweit die Begriffe

³⁾ siehe Abschnitt 2.3 auf Seite 14

⁴⁾ siehe Tabelle 2.3 auf Seite 20

⁵⁾ Statt von einem *Zusammenhang* könnte man auch von einer *Umgebung* sprechen. Diese Bezeichnung ist aber auch ein verbreiteter Fachbegriff, so dass auf seine Verwendung verzichtet wird. Die Exaktheit der Begriffe in diesem Dokument soll für Erstellung von ASBA ausreichen; was darüber hinausgeht, ist nicht Inhalt dieses Dokuments.

⁶⁾ Z. B. sind zwei logische Operatoren gleich, wenn sie stets denselben *Wahrheitswert* liefern.

Gleichheit und Identität korrelieren, wird hier nicht erörtert. siehe [29]

\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) oder „ A ist nicht identisch zu B “.

\equiv **Äquivalenz** „ $A \equiv B$ “ heißt, dass A und B sich in den interessierenden Eigenschaften für $'\equiv$ ' nicht unterscheiden. – „ A ist das gleiche wie B “ oder „ A ist so wie B “.

\ncong **Kontravalenz** „ $A \ncong B$ “ heißt, dass A und B sich in mindestens einer der interessierenden Eigenschaften für $'\ncong$ ' unterscheiden. – „ A ist nicht das gleiche wie B “ oder „ A ist nicht so wie B “.

$: \Leftrightarrow$ **Metadefinition** „ $A : \Leftrightarrow B$ “ heißt, dass der Metaausdruck A definitionsgemäß gleich dem Metaausdruck B ist, wobei B auch eine Definition in natürlicher Sprache sein kann. A und B können sich gegenseitig ersetzen. B darf dabei von A weder direkt noch indirekt abhängen, d. h. A darf in B und zugehörigen Definitionen noch nicht vorkommen.

Üblicherweise ist A hier eine Bezeichnung und B eine Aussage, so dass man „ $A : \Leftrightarrow B$ “ auch als „ A steht für B “ lesen kann. Oft wird damit Gleichheit ($'=$ '), Äquivalenz ($'\equiv$ ') oder eine andere Relation definiert.

$: =$ **Definition** „ $A : = B$ “ heißt, dass der Ausdruck A definitionsgemäß gleich dem Ausdruck B ist. Gewissermaßen ist A nur eine andere Schreibweise für B . A und B können sich gegenseitig ersetzen.⁷⁾ B darf dabei von A weder direkt noch indirekt abhängen, d. h. A darf in B und zugehörigen Definitionen noch nicht vorkommen. – Man beachte, dass $' : \Leftrightarrow '$ und $' : = '$ verschiedene Sprachebenen sind.

Üblicherweise ist A hier eine Variable und B eine Formel, so dass man „ $A : = B$ “ auch als „ A steht für B “ lesen kann.

Es sei noch

$$\mathcal{M} := \{ |, \&\&, ||, \Rightarrow, \Leftrightarrow, \Leftarrow, =, \neq, \equiv, \ncong, : \Leftrightarrow, : = \}$$

die Menge der metasprachlichen Operatoren und der mit Gleichheit verwandten Symbole.

2.2. Formale Elemente

Ein *formales Element* kann z. B. eine Menge, Zeichenfolge, Zahl, Formel, usw. sein. Zwei formale Elemente A und B sind *vergleichbar*, wenn beide von derselben Art sind, d. h. wenn z. B. jeweils beide Mengen, Zeichenfolgen, Zahlen oder formale Elemente – die vergleichbare Ergebnisse liefern – sind.

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, ist (b) dann richtig, wenn der Zwischenraum zwischen den einzelnen Zeichen nicht zählt. Wenn er aber zählt, ist (b) falsch.

⁷⁾ 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 \bar{P} und \bar{Q} gleich sind.

Im Zusammenhang mit binären Relationen werden noch einige Verabredungen getroffen. Dazu seien ' \sim ', ' \simeq ', ' \triangleleft ', ' \triangleright ', ' \trianglelefteq ' und ' \trianglerighteq ' Beispielsymbole für Relationen und ' $=$ ' und ' \neq ' die Symbole für Gleichheit und Ungleichheit. Wenn dann nichts anderes gesagt wird gelte stets:

$$\begin{aligned} ((A \sim B) \parallel (A = B)) &\Leftrightarrow (A \simeq B) \\ (A \triangleleft B) &\Leftrightarrow (B \triangleright A) \\ (A \trianglelefteq B) &\Leftrightarrow (B \trianglerighteq A) \end{aligned} \quad (2.1)$$

Mit der Definition einer Relation der einen Seite ist damit automatisch auch die der anderen Seite erfolgt, mit der Ausnahme, dass man „ $A \sim B$ “ so nicht mit Hilfe von „ $A \simeq B$ “ definieren kann. Dies könnte man zwar mit Hilfe des Ansatzes

$$(A \sim B) \Leftrightarrow (A \simeq B) \ \&\& \ (A \neq B) \quad (2.2)$$

versuchen, aber die so definierte Relation ' \sim ' kann, muss aber nicht mit der in (2.1) übereinstimmen. Allerdings lässt sich (2.1) aus (2.2) ableiten und wenn „ $(A = B) \Rightarrow (A \simeq B)$ “ gilt, auch (2.2) aus (2.1). – Auf einen Beweis wird hier verzichtet.

Es sei noch angemerkt, dass wegen (2.2) die Definition von ' \Leftarrow ' in Abschnitt 2.1.1 auf Seite 11 überflüssig ist und wegen der Klammerregeln (siehe Unterabschnitt 2.4.2 auf Seite 18) auch alle Klammern in diesem Abschnitt 2.2. Die Prioritäten der Operatoren ' \triangleleft ', ' \triangleright ', ' \trianglelefteq ' und ' \trianglerighteq ' unterscheiden sich normalerweise nicht; ebenso wenig die der Operatoren ' \sim ' und ' \simeq ', die aber durchaus verschieden von den Prioritäten von ' $=$ ' und ' \neq ' sein können.

Als Beispielsymbol für binäre Operatoren wird ' \circ ' verwendet. Mit ' \circ ' zusammenhängende Verabredungen werden hier nicht getroffen.

2.3. Schlussregeln

Die Regeln zur Formulierung und Prüfung der Beweise müssen fest codiert werden. Sie sind quasi die Axiome von ASBA und sollten daher möglichst wenig voraussetzen. Dazu wird ein *Genzen-Kalkül* verwendet, so wie er in [1] Kapitel 1.4 beschrieben ist (siehe auch [34, 33]).

Ein Beweis in ASBA besteht aus einer Menge von formales Element (die *Voraussetzungen*) und einer Folge von zulässigen Transformationen, mit der neue (Mengen von) formalen Elementen generiert werden, bis eine gewünschte Menge von formalen Elementen (die *Folgerungen*) abgeleitet sind.

In diesem Abschnitt geht es um die zulässigen Transformationen, d. h. die gültigen Schlussregeln. Dazu gehören zunächst die Basisregeln.

2.3.1. Basisregeln

Gemäß [1] Kapitel 1.4 *Ein vollständiger aussagenlogischer Kalkül* werden sechs Basisregeln definiert. Zuvor werden aber noch einige Definitionen gebraucht. Dazu seien n, m, k und l natürliche Zahlen (auch 0), α, α_i, β und β_j formale Elemente, X, X_i, Y und Y_j Mengen von formalen Elementen und

$$\begin{aligned} X &:= X_1 \cup X_2 \cup \dots \cup X_n \cup \{\alpha_1, \alpha_2, \dots, \alpha_m\} \\ Y &:= Y_1 \cup Y_2 \cup \dots \cup Y_k \cup \{\beta_1, \beta_2, \dots, \beta_l\} \end{aligned}$$

X und Y können auch die leere Menge sein. Damit wird definiert:

$\alpha \vdash \beta \quad :\Leftrightarrow \quad \beta$ ist mittels schrittweiser Anwendung *zulässiger Transformationen* (siehe weiter unten) aus α ableitbar. Sprechweise: Aus α ist β *ableitbar* oder *beweisbar*; kurz: „ α ableitbar β “ bzw. „ α beweisbar β “ – Es kann auch ' α ' durch ' X ' und/oder ' β ' durch ' Y ' ersetzt werden.

$X_1, X_2, \dots, X_n, \alpha_1, \alpha_2, \dots, \alpha_m \vdash Y_1, Y_2, \dots, Y_n, \beta_1, \beta_2, \dots, \beta_m \quad :\Leftrightarrow \quad X \vdash Y$

Eine *zulässige Transformation* ist die Anwendung einer *Basisregel* (siehe unten) oder einer davon abgeleiteten sonstigen *Schlussregel*, z. B. aus Unterabschnitt 2.3.3 auf der nächsten Seite.

Seien α und β wieder formale Elemente und X und Y Mengen von formalen Elementen. Für die sechs Basisregeln werden dann nur noch die logischen Operatoren ' \neg ' und ' \wedge ' benötigt. Bei den weiteren Schlussregeln wird noch ' \rightarrow ' gemäß der Definition 2.4 auf Seite 22 verwendet.

Bei den Schlussregeln und der Substitution⁸⁾ soll das Komma stärker binden als ' \vdash ' und ' \leftarrow '. ' \leftarrow ' steht für „und“ bzw. '&&'⁹⁾ und bindet schwächer als ' \vdash ' und ' \leftarrow '.¹⁰⁾

Bei der Auswahl der Basisregeln, der Formulierung und der Bezeichnungen wird auf [1, 34] zurückgegriffen. Wie in [34] steht 'E' für „-Einführung“ und 'B' für „-Beseitigung“ (oder „-Elimination“) von Operatoren.

$\frac{}{\alpha \vdash \alpha} \quad (\text{Anfangsregel}) \quad (\text{AR})$

$\frac{X \vdash \alpha}{Y \vdash \alpha} \quad , \text{ für } Y \supseteq X \quad (\text{Monotonieregel}) \quad (\text{MR})$

$\frac{X \vdash \alpha, \neg \alpha}{X \vdash \beta} \quad (\text{Einführung/Beseitigung der Negation Teil 1}) \quad (\neg 1)$

$\frac{X, \alpha \vdash \beta \mid X, \neg \alpha \vdash \beta}{X \vdash \beta} \quad (\text{Einführung/Beseitigung der Negation Teil 2}) \quad (\neg 2)$

$\frac{X \vdash \alpha, \beta}{X \vdash \alpha \wedge \beta} \quad (\text{Einführung der Konjunktion}) \quad (\wedge E)$

$\frac{X \vdash \alpha \wedge \beta}{X \vdash \alpha, \beta} \quad (\text{Beseitigung der Konjunktion}) \quad (\wedge B)$

Eine Schlussregel steht für die Aussage, dass mit den formalen Elementen¹¹⁾ über dem Strich auch die unter dem Strich gelten. Im Gegensatz zu den weiteren Schlussregeln werden die Basisregeln nicht weiter hinterfragt, d. h. sie gelten als Axiome.

⁸⁾ siehe 2.3.2 auf der nächsten Seite

⁹⁾ siehe Unterabschnitt 2.1.1 auf Seite 11

¹⁰⁾ siehe Fußnote 3 von Tabelle 2.3 auf Seite 20

¹¹⁾ hier: Aussagen in einer formalen Form.

2.3.2. Identitätsregeln

Die zulässigen Transformationen, d. h. die Anwendung der Schlussregeln, erfordern zulässige Substitutionen. Dazu wird dem Gleichheits- oder Identitätszeichen '=' mittels Einführungs- und Beseitigungsregel eine Bedeutung verliehen.¹²⁾ Es seien α , β und γ wieder formale Elemente. Zunächst wird definiert:

$\gamma(\alpha \leftarrow \beta) \quad := \quad$ Das formale Element, dass man erhält, wenn in γ alle oder nur einige Vorkommen von α durch β ersetzt.

Gegebenenfalls muss noch die Auswahl der Ersetzungen angegeben werden.

Nun können die beiden Identitätsregeln definiert werden:

$\frac{}{\alpha = \alpha} \quad (\text{Einführung der Identität}) \quad (=E)$

$\frac{\alpha = \beta \mid \gamma}{\gamma(\alpha \leftarrow \beta)} \quad (\text{Beseitigung der Identität}) \quad (=B)$

Die Identitätsregeln werden hier eingeführt, um die Substitution zu rechtfertigen. Wie die Basisregeln gelten sie als Axiome, würden also eigentlich dazu gehören. Da sie aber nicht weiter verwendet werden, werden sie hier nicht zu den Basisregeln gezählt.

2.3.3. Weitere Schlussregeln

In [1] werden aus den Basisregeln mittels zulässiger Transformationen weitere Schlussregeln abgeleitet.¹³⁾ Man vergleiche auch mit [34].

$\frac{X, \neg\alpha \vdash \alpha}{X \vdash \alpha} \quad (\text{Beseitigung der Negation; Indirekter Beweis}) \quad (\neg 3)$

$\frac{X, \neg\alpha \vdash \beta, \neg\beta}{X \vdash \alpha} \quad (\text{Reductio ad absurdum}) \quad (\neg 4)$

$\frac{X, \alpha \vdash \beta}{X \vdash \alpha \rightarrow \beta} \quad (\text{Einführung der Implikation}) \quad (\rightarrow E)$

$\frac{X \vdash \alpha \rightarrow \beta}{X, \alpha \vdash \beta} \quad (\text{Beseitigung der Implikation}) \quad (\rightarrow B)$

$\frac{X \vdash \alpha \mid X, \alpha \vdash \beta}{X \vdash \beta} \quad (\text{Schnittregel}) \quad (SR)$

$\frac{X \vdash \alpha \mid \alpha \rightarrow \beta}{X \vdash \beta} \quad (\text{Abtrennungsregel (Modus ponens)}) \quad (TR)$

¹²⁾ siehe [34]

¹³⁾ In [1] werden die Identitätsregeln zwar weder aufgeführt noch angewandt, ohne Substitution geht es aber nicht.

2.3.4. Beispiel einer Ableitung

Als Beispiel wird hier die Schnittregel aus den Basisregeln abgeleitet.¹⁴⁾ Dazu werden für die Zeile n definiert:

$$R_n := \begin{cases} A_n & , \text{ wenn in der Spalte „Regel“ „...Annahme“ steht.} \\ A_i \text{ bzw. } \bar{R}_i & , \text{ wenn in der Spalte „Regel“ die Zahl } i \text{ steht.} \\ \text{sonst: die Schlussregel, auf die in der Spalte „Regel“ verwiesen wird} \end{cases}$$

$S_n :=$ die Menge der Substitutionen in der Spalte „Substitutionen“.

$Z_n :=$ die Menge der Zeilennummern in der Spalte „angewendet auf“.

$\bar{R}_n :=$ das Ergebnis der Substitution auf die Regel bzw. Aussage R_n .

$$A_n := \begin{cases} \bar{R}_n & , \text{ wenn in der Spalte „Aussage“ nichts steht.} \\ \text{sonst: das formale Element in der Spalte „Aussage“} \end{cases}$$

$D_n :=$ die Menge der Abhängigkeiten in der Spalte „Abhängigkeiten“.

Zeile (n)	Regel (R_n)	Substitutionen (S_n)	erzeugte Regel (\bar{R}_n)	angewendet auf ... (Z_n)	Aussage (A_n)	Abhängig- keiten (D_n)
1	fixe Annahme				$X \vdash \alpha$	1
2	fixe Annahme				$X, \alpha \vdash \beta$	2
3	Annahme				$X \vdash \beta$	3
4	(MR)	$Y \leftrightarrow X, \neg \alpha$	$\frac{X \vdash \alpha}{X, \neg \alpha \vdash \alpha}$	1	$X, \neg \alpha \vdash \alpha$	1
5	(AR)	$\alpha \leftrightarrow \gamma$	$\frac{}{\gamma \vdash \gamma}$		$\gamma \vdash \gamma$	
6	(MR)	$\alpha \leftrightarrow \gamma$ $X \leftrightarrow \gamma$ $Y \leftrightarrow X, \gamma$	$\frac{\gamma \vdash \gamma}{X, \gamma \vdash \gamma}$	5	$X, \gamma \vdash \gamma$	
7	6	$\gamma \leftrightarrow \neg \alpha$	$X, \neg \alpha \vdash \neg \alpha$			
8	($\neg 1$)	$X \leftrightarrow Z$	$\frac{Z \vdash \alpha, \neg \alpha}{Z \vdash \beta}$			
9	8	$Z \leftrightarrow X, \neg \alpha$	$\frac{X, \neg \alpha \vdash \alpha, \neg \alpha}{X, \neg \alpha \vdash \beta}$	4, 7	$X, \neg \alpha \vdash \beta$	1
10	($\neg 2$)		$\frac{X, \alpha \vdash \beta \mid X, \neg \alpha \vdash \beta}{X \vdash \beta}$	2, 9	$X \vdash \beta$	1, 2

Tabelle 2.1.: Ableitung der Schnittregel aus den Basisregeln

Eine Zeile n in der Tabelle 2.1 hat folgende Bedeutung:

- Nehme die Regel/Aussage R_n und wende darauf die Substitutionen aus S_n an. Das Ergebnis ist \bar{R}_n .
- Wende die Regel/Aussage \bar{R}_n auf die Aussagen A_i mit $i \in Z_n$ an.
- Die Aussage (unter dem Strich von) \bar{R}_n bildet die (erzeugte) Aussage A_n .
- Die Menge D_n der Abhängigkeiten ist gleich der Vereinigung der Mengen D_i mit $i \in Z_n$.¹⁵⁾

¹⁴⁾ Die Form der Tabelle ist angelehnt an [34] Kapitel 2.2.4 Eine Beispielableitung.

¹⁵⁾ Wenn D_n leer ist, dann ist A_n allgemeingültig.

2.4. Aussagenlogik

2.4.1. Konstante und Operatoren

Die Tabelle 2.2 auf der nächsten Seite¹⁶⁾ definiert für die zweiwertige Logik Konstanten- und Operator-symbole über die Wahrheitswerte ihrer Anwendung. So ergeben sich, abhängig von den Wahrheitswerten der Operanden A und B ¹⁷⁾, 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.

Für einige Junktoren, 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 Junktoren sind in der Tabelle grau hinterlegt. Hellgrau hinterlegt sind Zeilen mit weniger gebräuchlichen Junktoren. Die restlichen Operationen 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 die Operatoren derselben Priorität gilt Rechtsklammerung¹⁸⁾. Im Folgenden wird nur noch ein Teil der logischen Operatoren aus der Tabelle 2.2 auf der nächsten Seite und der metasprachlichen Operatoren aus Unterabschnitt 2.1.1 auf Seite 11 berücksichtigt. Diese werden in der Tabelle 2.3 auf Seite 20 mit abnehmender Priorität aufgelistet.

Die Prioritäten der logischen Operatoren wurden aus [1] Kapitel 1.1 Seite 5 entnommen und ergänzt und die der metasprachlichen Operatoren 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 in streng formaler Form vorliegen. Die Formalisierung stützt sich auf [28]; siehe auch [21, 24]. Da Computerprogramme mit der *Polnischen Notation*¹⁹⁾

¹⁶⁾ Die Tabelle basiert auf den Wahrheitstafeln in [27] Kapitel 2.2 und [1] Kapitel 1.1 Seite 3.

¹⁷⁾ Im Gegensatz zu Paragraph 2.4.3.1 auf Seite 20 können A und B hier beliebige Aussagen – auch Formeln – sein.

¹⁸⁾ 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 \rightarrow , [...]“

¹⁹⁾ Bei der *Polnischen Notation* wird eine zweistellige Operation $(A \circ B)$ dargestellt als $\circ AB$. Eine Zwischenstufe ist $\circ(A, B)$, 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.

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
⊤	W	-	-	-	-	-	-	Verum	Wahr	-
⊥	F	-	-	-	-	-	-	Falsum	Falsch	-
	-	W	W	-	-	-	-			-
(...)	-	W	F	-	-	-	-	Klammerung ³	A ist geklammert	6 ⁴
¬	-	F	W	-	-	-	-	Negation	Nicht A	5 ⁵
	-	F	F	-	-	-	-			-
	-	-	-	W	W	W	W	Tautologie		-
∨	-	-	-	W	W	W	F	Disjunktion; Adjunktion; Alternative	A oder B	3
← ⇐ ⊂	-	-	-	W	W	F	W	Replikation; Konversion; konverse Implikation	A folgt aus B	2
⌋	-	-	-	W	W	F	F	Präpendenz	Identität von A	-
→ ⇒ ⊃	-	-	-	W	F	W	W	Implikation; Subjunktion; Konditional	Wenn A so B; Aus A folgt B; A nur dann wenn B	2
⌊	-	-	-	W	F	W	F	Postpendenz	Identität von B	-
↔ ⇔	-	-	-	W	F	F	W	Äquivalenz; Bijunktion; Bikonditional	A genau dann wenn B; A dann und nur dann wenn B	1
∧ & ·	-	-	-	W	F	F	F	Konjunktion	A und B; Sowohl A als auch B	4
↑ ≰	-	-	-	F	W	W	W	NAND; Unverträglichkeit; Sheffer-Funktion	Nicht zugleich A und B	4
+ ∨̇ ⊍ ⊕	-	-	-	F	W	W	F	XOR; Antivalenz; ausschließende Disjunktion	Entweder A oder B	3
↮ ⇎ ≠	-	-	-	"	"	"	"	Kontravalenz		-
⌊	-	-	-	F	W	F	W	Postnonpendenz	Negation von B	-
↛ ⇏ ⊅	-	-	-	F	W	F	F	Postsektion		-
⌊	-	-	-	F	F	W	W	Pränonpendenz	Negation von A	-
↚ ⇍ ⊄	-	-	-	F	F	W	F	Präsektion		-
↓ ∇	-	-	-	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 \top und \perp als Junktoren bzw. Operatoren bezeichnet.

Die Operatoren ' \subset ', ' \supset ', ' \Leftarrow ' und ' $\not\subset$ ' 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 Operationen aufzuführen.

³ Klammerung ist genau genommen kein Operator und wird nicht nur bei logischen, sondern auch bei anderen Ausdrücken verwendet.

⁴ Die Priorität der Klammern ist größer als die aller Operatoren.

⁵ Die Priorität der unären Operatoren muss größer 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 2.2.: Definition von aussagenlogischen Symbolen.

Klammern	()
Unäre logische Operatoren	\neg
Binäre logische Operatoren	$\wedge \quad \cdot \quad \uparrow$ $\vee \quad + \quad \downarrow$ $\leftarrow \quad \rightarrow$ \leftrightarrow
Mit Gleichheit verwandte Symbole; ihre Prioritäten untereinander sind nicht eindeutig und bleiben daher undefiniert.	$= \quad \neq \quad \equiv \quad \neq$
Ableitungsrelation ¹ Substitution ¹ Definition	\vdash \leftarrow $:=$
Metasprachliche Operatoren ²	$\&\&$ \parallel $\Leftarrow \quad \Rightarrow$ \Leftrightarrow \hookleftarrow
Metadefinition	$:\Leftrightarrow$
Strukturelemente der natürlichen Sprache, z. B. Satzzeichen ³	$\cdot \quad , \quad ; \quad \text{usw.}$

¹ siehe Unterabschnitt 2.3.1 auf Seite 14² siehe Unterabschnitt 2.1.1 auf Seite 11 und für ' \hookleftarrow ' Unterabschnitt 2.3.1 auf Seite 14³ Innerhalb von formalen Elementen können Satzzeichen eine andere Priorität haben.
Siehe z. B. Unterabschnitt 2.3.1 auf Seite 14.**Tabelle 2.3.:** Prioritäten von Operatoren in abnehmender Reihenfolge

besser umgehen können und Klammern dort überflüssig sind, werden viele Formeln auch in die Polnische Notation überführt.

2.4.3.1. Bausteine der aussagenlogischen Sprache

Zur Einteilung der aussagenlogischen Junktoren werden die folgenden Mengen definiert:

\mathbb{N}_0	$:=$	Menge der <i>natürlichen Zahlen</i> einschließlich 0
\mathcal{C}	$:= \{\top, \perp\}$, Menge der <i>aussagenlogischen Konstanten</i>
\mathcal{U}	$:= \{\neg\}$, Menge der <i>unären aussagenlogischen Operatoren</i>
\mathcal{B}	$:= \{\wedge, \vee, \rightarrow, \leftrightarrow, \leftarrow, \uparrow, \downarrow, \cdot, +\}$, Menge der <i>binären aussagenlogischen Operatoren</i>

Damit sind alle in der Tabelle 2.2 auf der *vorherigen Seite* verwendeten wesentlichen Konstanten und Operatoren²⁰⁾ erfasst und es können die folgende Mengen definiert werden:

\mathcal{V}	$:= \{p_n \mid n \in \mathbb{N}_0\}$, Menge der <i>atomaren Formeln</i>	(2.3)
\mathcal{J}	$:= \mathcal{C} \cup \mathcal{U} \cup \mathcal{B}$, Menge der <i>Junktoren bzw. Operatoren</i> ; auch <i>logische Signatur</i>	
\mathcal{A}	$:= \mathcal{V} \cup \mathcal{J}$, <i>Alphabet der aussagenlogischen Sprache</i> (für \mathcal{J})	
\mathcal{J}_x	$\subseteq \mathcal{J}$, eine Teilmenge von \mathcal{J} für eine Indexvariable x	
\mathcal{A}_x	$:= \mathcal{V} \cup \mathcal{J}_x$, <i>Alphabet der aussagenlogischen Sprache für \mathcal{J}_x</i>	

²⁰⁾ Jeweils nur die ersten der grau hinterlegten Zeilen sowie ' \cdot '.

Für Elemente aus \mathcal{V} werden hier normalerweise die großen lateinischen Buchstaben A, B, C , usw. verwendet. Die Elemente aus \mathcal{V} (atomare Formeln) 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:

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

$$\begin{aligned} \mathcal{V} &\subset \mathcal{F}_x \\ \mathcal{J}_x \cap \mathcal{C} &\subset \mathcal{F}_x \\ A \in \mathcal{F}_x &\Rightarrow (\circ A) \in \mathcal{F}_x, \text{ für } \circ \in \mathcal{U} \cap \mathcal{J}_x \\ A, B \in \mathcal{F}_x &\Rightarrow (A \circ B) \in \mathcal{F}_x, \text{ für } \circ \in \mathcal{B} \cap \mathcal{J}_x \end{aligned}$$

Nur die auf diese Weise konstruierten Formeln sind Elemente von \mathcal{F}_x .

Für $\mathcal{J} = \mathcal{J}_x$ sei noch $\mathcal{F} := \mathcal{F}_x$.

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

$$\begin{aligned} \mathcal{V} &\subset \mathcal{F}_x^p \\ \mathcal{J}_x \cap \mathcal{C} &\subset \mathcal{F}_x^p \\ A \in \mathcal{F}_x^p &\Rightarrow (\circ A) \in \mathcal{F}_x^p, \text{ für } \circ \in \mathcal{U} \cap \mathcal{J}_x \\ A, B \in \mathcal{F}_x^p &\Rightarrow (A \circ B) \in \mathcal{F}_x^p, \text{ für } \circ \in \mathcal{B} \cap \mathcal{J}_x \end{aligned}$$

Nur die auf diese Weise konstruierten Formeln sind Elemente von \mathcal{F}_x^p .

Für $\mathcal{J} = \mathcal{J}_x$ sei noch $\mathcal{F}^p := \mathcal{F}_x^p$.

Wie man leicht sieht, gilt:

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

Durch Anwendung der Klammerregeln von Paragraph 2.4.3.1 auf der vorherigen Seite lassen sich in der Regel noch viele Klammern der Formeln aus \mathcal{F}_x einsparen. Die Formeln aus \mathcal{F}_x^p sind frei von Klammern. Die Namen der Operatoren finden sich in der Tabelle 2.2 auf Seite 19. Für aussagenlogische Formeln, d. h. von Elementen aus \mathcal{F} bzw. \mathcal{F}^p , werden hier normalerweise die kleinen griechischen Buchstaben α, β, γ , usw. verwendet. Sie können dabei auch atomare Formeln bezeichnen (siehe (2.3)).

2.4.4. Definition aussagenlogische Operatoren durch andere

Im folgenden gelte für zwei aussagenlogische Formeln α und β :

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

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

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

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

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{F}_x bzw. 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}_{and} :

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

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

$$\begin{aligned}
 (\alpha \leftrightarrow \beta) &:= ((\alpha \rightarrow \beta) \wedge (\alpha \leftarrow \beta)) && \leftrightarrow \alpha \beta := \wedge \rightarrow \alpha \beta \leftarrow \alpha \beta \\
 \perp &:= (p_0 \wedge (\neg p_0)) && \perp := \wedge p_0 \neg p_0 \\
 (\alpha \cdot \beta) &:= (\alpha \wedge \beta) && \cdot \alpha \beta := \wedge \alpha \beta \\
 (\alpha \uparrow \beta) &:= (\neg(\alpha \wedge \beta)) && \uparrow \alpha \beta := \neg \wedge \alpha \beta \qquad (2.6)
 \end{aligned}$$

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

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

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

$$\top := (p_0 \vee (\neg p_0)) \qquad \top := \vee p_0 \neg p_0$$

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

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

Nun sind wieder alle Operatoren definiert.

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

$$\begin{aligned}
 (\alpha \vee \beta) &:= ((\neg\alpha) \rightarrow \beta) && \vee \alpha \beta := \rightarrow \neg \alpha \beta \\
 (\alpha \wedge \beta) &:= (\neg((\neg\beta) \leftarrow \alpha)) && \wedge \alpha \beta := \neg \leftarrow \neg \beta \alpha
 \end{aligned}$$

woraufhin dann (2.4) bzw. (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 $\mathcal{J}_{\text{nand}}$ bzw. \mathcal{J}_{nor} schließlich werden die passenden Definitionen 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

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

ausgewählt und es ist (2.6) bzw. (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{F}_x , genau die gesamte Formelmengen \mathcal{F} 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:

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

> > > Aussagenlogik weiter bearbeiten. < < <

2.5. Prädikatenlogik

> > > Prädikatenlogik bearbeiten. < < <

2.6. Mengenlehre

> > > Mengenlehre bearbeiten. < < <

3. Design

Diese 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, 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 in Abschnitt 1.3 auf Seite 6. Die beiden Ziele 1 Daten und 15 Lizenz sind für die Entwicklung von ASBA von sekundärer Bedeutung und werden daher in diesen Abschnitt nicht übernommen. Die anderen Ziele werden noch verfeinert.

> > > **Ziele aus Abschnitt 1.3 in Anforderungen umwandeln.** < < <

1. *Form*: Die Daten liegt in formaler, geprüfter Form vor. (siehe Ziel 2 auf Seite 6)
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 Fachgebietes und seiner untergeordneten Fachgebiete gültig, solange sie nicht umdefiniert werden. Das oberste Fachgebiet ist die ganze Mathematik. (siehe Ziel 3 auf Seite 6)

3. *Prüfung*: Vorhandene Beweise können automatisch geprüft werden. (siehe Ziel 4 auf Seite 6)
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 viele Beweisschritte und welche Axiome und Sätze¹⁾ er verwendet. (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*: Axiome 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 \LaTeX Quellcode (*.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. *T_EXstudio* als Editor für L^AT_EX. → <http://www.texstudio.org/> – Lizenz siehe [7]
T_EXstudio 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 L^AT_EX 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

1.1. 1.1 Fragen → 1.2 Eigenschaften	5
1.2. 1.2 Eigenschaften → 1.3 Ziele	7
1.3. 1.1 Fragen → 1.3 Ziele	7
2.1. Ableitung der Schnittregelaus den Basisregeln	17
2.2. Definition von aussagenlogischen Symbolen.	19
2.3. Prioritäten von Operatoren in abnehmender Reihenfolge	20

Abbildungsverzeichnis

1.1. Die Umgebung von ASBA	8
--------------------------------------	---

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> 02.01.2004²⁾
- [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](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

- (AR), 17, 34
- ASBA, 4–6, 8, 9, 11, 12, 14, 18, 24, 25, 28
- Ableitungsrelation, 20, 34
- Abtrennungsregel, 16, 34
- Anfangsregel, 15, 34
- Ausgabeschema, 6, 8, 9, 24, 27
- Axiom, 4–9, 18, 24–27
- Basisregel, 14–17, 28, 35
- Beweisschritt, 9, 11
- Beweis, 4–9, 18, 24–27
- Boolsche Signatur, 22
- Fachbegriff, 4–6, 9, 24–27
- Fachgebiet, 4–6, 9, 24–27
- Identitätsregel, 16, 34
- (MR), 17, 35
- metasprachlicher Ausdruck, 11, 12, 35
- metasprachliche Aussage, 11
- metasprachlicher Operator, 12, 13, 18, 20, 33, 34
- Metasprache, 11, 35
- Monotonieregel, 15, 34
- Prädikatenlogik, 35
- Prädikat, 9
- (SR), 35
- Satz, 4–6, 8, 9, 18, 24–27
- Schlussregel, 15, 34, 35
- Schnittregel, 16, 17, 28, 34
- (TR), 34
- Wahrheitswert, 12
- ableitbar, 15, 33
- \vdash , 34
- atomare Formel, 20, 21, 33
- beweisbar, 15, 34
- formales Element, 13–17, 20, 35
- gültige Schlussregel, 14, 36
- interessierende Eigenschaft, 12, 13
- logische Signatur, 20, 22, 23, 34
- $\&\&$, 13, 15, 34
- $\|$, 13
- $(\neg 1)$, 17
- $(\neg 2)$, 17
- $|$, 33
- vergleichbar, 12, 13
- zulässige Transformation, 14–16
- Alphabet der logischen Sprache, 20
- Atom, 21
- atomare Formeln, Menge der, 20
- aussagenlogische Formel in Polnischer Notation, 21
- aussagenlogische Formel mit Klammerung, 21
- binären Operatoren, Menge der, 20
- Definition, 13
- Gleichheit, 12
- Junktoren, Menge der, 20
- Konstanten, Menge der, 20
- Kontravalenz, 13
- Metadefinition, 13
- natürlichen Zahlen, Menge der, 20
- Polnische Notation, 18
- Satzbuchstabe, 21
- Teil-Alphabet der aussagenlogischen Sprache, 20
- Ungleichheit, 13
- unären Operatoren, Menge der, 20
- Ziel, 6
- Äquivalenz, 13

Symbolverzeichnis

(\dots) , 19	\Rightarrow , 12, 13, 20
$+$, 20	\parallel , 12, 20
\wedge , 19, 20	\Leftarrow , 12, 13, 20
\leftrightarrow , 19, 20	\neq , 13, 14, 20
\rightarrow , 19, 20	$(\neg 1)$, 15
\cdot , 20	$(\neg 2)$, 15
\uparrow , 19, 20	$(\neg 3)$, 16
\downarrow , 19, 20	$(\neg 4)$, 16
\neg , 19, 20	\neq , 13, 20
\vee , 19, 20	\sim , 14
\leftarrow , 19, 20	\simeq , 14
$+$, 19	\triangleright , 14
$ $, 12	\triangleright , 14
(AR) , 15	\leftarrow , 20
(MR) , 15	$ $, 13
(SR) , 16	
(TR) , 16	
\vdash , 15, 20	
$(\wedge B)$, 15	
$(\wedge E)$, 15	
\mathcal{A}_x , 20	
\mathcal{A} , 20	
\mathcal{B} , 20	
\mathcal{C} , 20	
\mathcal{F}_x^p , 21	
\mathcal{F}_x , 21	
\mathcal{J}_x , 20	
\mathcal{J} , 20	
\mathcal{M} , 13	
\mathcal{U} , 20	
\mathcal{V} , 20	
$:=$, 13, 20	
$(=B)$, 16	
$(=E)$, 16	
\equiv , 13, 20	
$=$, 12–14, 20	
$(\rightarrow B)$, 16	
$(\rightarrow E)$, 16	
\perp , 19	
\triangleleft , 14	
\trianglelefteq , 14	
\top , 19	
$\&\&$, 12, 20	
$:\Leftrightarrow$, 13, 20	
\Leftrightarrow , 12, 13, 20	

Glossar

- \vdash Ableitungsrelation: ... ableitbar ... (siehe *ableitbar*). 15, 20, 31, 32, 34
- \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. 20, 32
- \mathcal{C} Die Menge der aussagenlogischen Konstanten. 20, 32
- \mathcal{F}_x Eine Teilmenge der Menge \mathcal{F} der aussagenlogischen Formeln mit Klammerung. 21, 32
- \mathcal{F}_x^p Eine Teilmenge der Menge \mathcal{F} der aussagenlogischen Formeln in polnischer Notation. 21, 32
- \mathcal{J} Die Menge der aussagenlogischen Operatoren. 20, 32
- \mathcal{J}_x Eine Teilmenge der Menge \mathcal{J} der aussagenlogischen Operatoren. 20, 32
- \mathcal{M} Die Menge der metasprachlichen Operatoren und der mit Gleichheit verwandten Symbole. 13, 32
- \mathcal{U} Die Menge der aussagenlogischen unären Operatoren. 20, 32
- \mathcal{V} Die Menge der aussagenlogischen atomaren Formeln. 20, 32
- $:=$ Ein metasprachlicher Operator: ... definitionsgemäß gleich ... 13, 20, 32
- $=$ Ein metasprachlicher Operator: ... gleich (ist dasselbe wie, ist identisch zu) ... 12–14, 20, 32
- \equiv Ein (Meta-)Operator: ... äquivalent (ist das gleiche wie, ist so wie) zu ... 13, 20, 32
- \perp Eine Aussagenlogische Konstante: Falsch. 19, 32
- \triangleleft Ein Beispielsymbol für eine Relation mit Umkehrrelation \triangleright . 14, 32
- \trianglelefteq Ein Beispielsymbol für eine Relation mit Gleichheit und Umkehrrelation \trianglerighteq . 14, 32
- \top Eine Aussagenlogische Konstante: Wahr. 19, 32
- $:\Leftrightarrow$ Ein metasprachlicher Operator: ... definitionsgemäß gleich (definitionsgemäß genau dann, wenn) ... 13, 20, 32
- \Leftrightarrow Ein metasprachlicher Operator: ... genau dann wenn ... 12, 13, 20, 32
- \Rightarrow Ein metasprachlicher Operator: ... dann auch ... 12, 13, 20, 32
- \parallel Ein metasprachlicher Operator: ... oder ... 12, 20, 31, 32
- \Leftarrow Ein metasprachlicher Operator: ... sofern ... 12, 13, 20, 32
- $\&\&$ Ein metasprachlicher Operator: ... und ... Die Priorität ist höher als die von $'|'$. 12, 15, 20, 31, 32, 34
- \neq Ein metasprachlicher Operator: ... ungleich (nicht dasselbe wie, nicht identisch zu) ... 13, 14, 20, 32
- \neq Ein (Meta-)Operator: ... nicht äquivalent (ist nicht das gleiche wie, ist nicht so wie) ... 13, 20, 32
- \sim Ein Beispielsymbol für eine Relation. 14, 32
- \simeq Ein Beispielsymbol für eine Relation mit Gleichheit. 14, 32
- \triangleright Ein Beispielsymbol für eine Relation mit Umkehrrelation \triangleleft . 14, 32

- \supseteq Ein Beispielsymbol für eine Relation mit Gleichheit und Umkehrrelation \subseteq . 14, 32
- \leftarrow Substitution: ... substituiert durch ... (siehe die Definition in Unterabschnitt 2.3.1 auf Seite 14). 20, 32
- | Ein metasprachlicher Operator: ... und ... Die Priorität ist niedriger als die von '&&'. 13, 31–33
- (=B) Beseitigung von '='. 32
- (=E) Einführung von '='. 32
- (\wedge B) Beseitigung von ' \wedge '. 32
- (\wedge E) Einführung von ' \wedge '. 32
- (\rightarrow B) Beseitigung von ' \rightarrow '. 32
- (\rightarrow E) Einführung von ' \rightarrow '. 32
- (\neg 1) Einführung/Beseitigung von ' \neg ' Teil 1. 17, 32
- (\neg 2) Einführung/Beseitigung von ' \neg ' Teil 2. 17, 32
- (\neg 3) Beweistechnik „Indirekter Beweis“. 32
- (\neg 4) Reductio ad absurdum (indirekter Beweis). 32
- (AR) Anfangsregel. 17, 31, 32, 34
- (MR) Monotonieregel. 17, 31, 32, 35
- (SR) Schnittregel (Modus ponens). 31, 32, 35
- (TR) Abtrennungsregel. 31, 32, 34
- ableitbar** Wenn sich eine Formel β aus einer Formel α mittels zulässiger Transaktionen ableiten lässt, heißt β ableitbar aus α . Sprechweise: „ α ableitbar β “. Eine oder beide Formeln α bzw. β dürfen dabei durch Formelmengen ersetzt werden. (siehe auch \vdash und Ableitungsrelation) – Synonym: beweisbar. 15, 31, 33
- Ableitungsrelation** Die Relation ' \vdash '. 20, 31, 34
- Abtrennungsregel** Eine *Schlussregel* - siehe (TR). 16, 31, 34
- Anfangsregel** Eine *Schlussregel* um beginnen zu können - siehe (AR). 15, 31, 34
- ASBA** Programmsystem, das Axiome, Sätze, Beweise und Auswertungen behandeln kann. 4–6, 8, 9, 11, 12, 14, 18, 24, 25, 28, 31
- atomare Formel** Eine Formel, die sich nicht weiter zerlegen lässt. 20, 21, 31, 33
- Ausgabeschema** Ein Schema, mit dem bestimmte mathematische Objekte ausgegeben werden sollen. 6, 8, 9, 24, 27, 31
- Axiom** Eine Formel, die unbewiesen als wahr angesehen wird. 4–9, 18, 24–27, 31
- Basisregel** Eine *Schlussregel*, die nicht mehr auf andere zurückgeführt wird. Obwohl das auch auf die Identitätsregeln zutrifft, werden diese hier aber nicht dazu gezählt.. 14–17, 28, 31, 35
- Beweis** Eine zulässige Ableitung von Folgerungen aus gegebenen Voraussetzungen. 4–9, 18, 24–27, 31
- beweisbar** Synonym zu *ableitbar*. 15, 31, 34
- Beweisschritt** Eine Vorschrift, wie aus vorgegebenen Aussagen eine weitere folgt. 11, 31
- Boolsche Signatur** Die logische Signatur $\{\neg, \wedge, \vee\}$. 22, 31

Fachbegriff Ein Name für einen mathematischen Begriff. 4–6, 9, 24–27, 31

Fachgebiet Ein Teil der Mathematik mit einer zugehörigen Basis aus Axiomen, Sätzen und spezifischen Fachbegriffen und Darstellungen. 4–6, 9, 24–27, 31

formales Element Ein mathematisches Element in formaler Schreibweise. Bis auf wenige Aussagen kommen darin *metasprachliche Ausdrücke* nicht mehr vor. 13–16, 31, 35

gültige Schlussregel Eine Schlussregel die aus den Basisregeln und den schon als gültig abgeleiteten Schlussregeln abgeleitet werden kann. 31

Identitätsregel Eigentlich eine Basisregel zur Identität. Da sie nur zur Rechtfertigung der Substitution verwendet werden, werden sie hier aber nicht dazu gezählt. 16, 31, 34

interessierende Eigenschaft Solche Eigenschaften von Ausdrücken, die im aktuellen Zusammenhang von Interesse sind. 12, 13, 31

logische Signatur Eine in *Metasprache* verfasste Aussage, die auch zusammengesetzt sein kann. 20, 22, 31, 34

Metasprache Eine Sprache, in der Aussagen über Elemente einer anderen Sprache getroffen werden können. 11, 31, 35

metasprachliche Aussage Eine in *Metasprache* verfasste Aussage, die auch zusammengesetzt sein kann. 11, 31

metasprachlicher Ausdruck Eine in normaler Sprache verfasste Aussage, die auch zusammengesetzt sein kann. 11, 12, 31, 35

metasprachlicher Operator Ein Operator, dessen Operanden *metasprachliche Ausdrücke* sind. 12, 20, 31, 33, 34

Monotonieregel Eine *Schlussregel* - siehe (MR). 15, 31, 34

Prädikat Ein Element der *Prädikatenlogik* (siehe Abschnitt 2.5 auf Seite 23). Z. B. kann man eine „Gruppe“ als ein zweistelliges Prädikat „Gruppe($G, +$)“ definieren, in dem G eine Menge und $+$ eine Operation, d. h. eine (zweistellige) Funktion „ $+ : G \times G \rightarrow G$ “ ist, so dass die Gruppenaxiome erfüllt sind. 9, 31

Prädikatenlogik siehe Abschnitt 2.5 auf Seite 23. 31, 35

Satz Eine mathematische Aussage, dass eine bestimmte Folgerung aus gegebenen Voraussetzungen abgeleitet werden kann. 4–6, 8, 9, 18, 24–27, 31

Schlussregel Eine Regel für eine (zulässige) Umwandlung von Formeln. 15, 31, 34, 35

Schnittregel Eine *Schlussregel* - siehe (SR). 16, 17, 28, 31, 34

vergleichbar Zwei *metasprachliche Ausdrücke* bzw. *formale Elemente* heißen – auf eine bestimmte Art – vergleichbar, wenn sie auf diese Art (z. B. als Zeichenketten oder als vergleichbare Ergebnisse von Formeln) verglichen werden können. Die Art muss implizit bekannt oder explizit angegeben sein. Meistens genügt es zu wissen, was für metasprachliche Ausdrücke bzw. formale Elemente es sind. Sie müssen dann nur von derselben Art sein. 12, 13, 31

Wahrheitswert Wahrheitswerte sind die Werte „wahr“ und „falsch“, oft auch als „true“ und „false“ oder einfach '1' und '0' bezeichnet. 12, 31

zulässige Transformation Eine zulässige Umformung oder Erzeugung einer Formel aus einer vorgegebenen Menge von Formeln, d. h. die Anwendung einer gültigen Schlussregel. 14, 15, 31