

Dialogische Logik und mathematischer Unterrichtsdiskurs

Christian Thiel gewidmet

Günther Görz

Universität Erlangen-Nürnberg, Institut für Informatik

Zusammenfassung

Der traditionellen Auffassung der Logik als Theorie des rationalen Argumentierens hat Paul Lorenzen mit der Begründung der dialogischen Logik ein pragmatisches Fundament verliehen. Mit ihrem allgemeineren Ansatz bei beweisdefiniten anstelle von wahrheitswertdefiniten Aussagen und durch die Rückführung auf schematisiertes Handeln in Dialogspielen gelingt ein umfassendes Verständnis der Semantik. Die übliche modelltheoretische Interpretationssemantik lässt sich dabei ohne weiteres rekonstruieren, doch kommt der Modelltheorie insgesamt eine erläuternde, nicht eine fundierende Rolle zu. Dieses und ihr Ausgangspunkt bei der Beweisdefinitheit machen die dialogische Logik für Argumentieren und Beweisen im Mathematikunterricht interessant; eine Ähnlichkeit mit der Kolmogorovschen Aufgabenlogik liegt auf der Hand. Viele Beweise der Schulmathematik (im „context of justification“) lassen sich ohne weiteres dialogisch rekonstruieren — der Gang der Argumentation kann teilweise direkt in Dialogspielzüge überführt werden. Anhand eines ausführlichen Beispiels wird deutlich, welchen Beitrag ein am rationalen Argumentieren orientierter Aufbau der Logik für den Mathematikunterricht leisten kann. Abschliessend wird das in Erlangen entwickelte Programm *DiaLogic* vorgestellt, mit dem Dialogspiele entweder interaktiv oder vollautomatisch in der Art eines Tableaubeweislers durchgeführt werden können.

1 Vorbemerkungen zum Anfang in der Logik

„Was haben die Gebiete Astronomie, Ökonomie, Volkswirtschaftslehre, Rechtswissenschaften, Mathematik, Medizin, Physik und Soziologie gemein? Sicherlich nicht viel, was ihren Untersuchungsgegenstand angeht — und auch nicht besonders viel, was ihre Methodologie betrifft. Sie haben jedoch dieses miteinander und mit vielen anderen Gebieten gemein: Sie beruhen auf einem gewissen Standard von Rationalität. In jedem dieser Gebiete geht man davon aus, dass sich zwischen rationalen Argumenten, die auf angenommenen Prinzipien oder Fakten beruhen, und wilder Spekulation und Fehlschlüssen, die nicht aus den Annahmen folgen, unterscheiden lässt. ... Die *ganze* Wissenschaft beruht so gesehen auf Logik, nämlich auf der Fähigkeit von Personen, meistens korrekt zu argumentieren, oder wenn dies einmal nicht gelingt, auf der Fähigkeit anderer, Lücken in Argumenten aufzuzeigen. Obwohl es kaum besonders viel gibt, worin alle übereinstimmen, gibt es anscheinend doch eine allgemeine Verständigung darüber, worauf man gerechtfertigter Weise aus gegebenen Informationen schließen darf...“ Mit diesen Sätzen beginnt eines der neueren Logiklehrbücher, *Sprache, Beweis und Logik* von Jon Barwise und John Etchemendy.

Die Logik befasst sich also mit den Regeln der rationalen, zweckgerichteten Argumentation und somit insbesondere mit den Regeln des korrekten Schliessens. Schon Aristoteles charakterisiert

dieses Regelwissen als ein Wissen, das eine vom jeweiligen Thema der Argumentation abgelöste Form hat. Die Logik gehört zum zentralen Bestand jeder Theorie des rationalen Argumentierens, indem sie nur zu einem geringen Teil den Ablauf inhaltlicher Argumentationen behandelt und lediglich elementare und logisch zusammengesetzte Aussagen zulässt. Genauer besehen ist mit Christian Thiel [16] zu unterscheiden zwischen sog. „Sachlogiken“, d.h. Theorien von Schlussweisen, deren Gültigkeit vom Inhalt der verwendeten Aussagen (und damit von einem Sachgebiet) abhängt und der Theorie der Schlüsse, die nicht in dieser Weise abhängig sind, welche also die allgemeinen Argumentationsregeln untersucht. Für letztere bietet die Mathematik ein idealtypisches Anwendungsfeld — beim Beweisen steht die Deduktion nach allgemeinen Regeln im Mittelpunkt.

Traditionell — gegeben durch die Editoren des Aristotelischen Organons — wird die Logik eingeteilt in die Lehre vom

- Begriff (Prädikation und Abstraktion, Definitionslehre),
- Urteil (Elementarsätze und zusammengesetzte Aussagen), und
- Schluss (formale Logik i.e.S.).

Die moderne formale Logik, als deren Begründer Gottlob Frege gilt, hat verschiedene Zugänge zu den logischen Regeln erarbeitet. Hauptsächlich wird unterschieden zwischen der (primär) syntaktisch charakterisierten *Beweistheorie*, die ein evident bzw. semantisch motiviertes System logischer „Axiome“ und Deduktionsregeln zugrundelegt, und der semantisch charakterisierten Wahrheitstheorie, in deren Mittelpunkt der Wahrheits- bzw. Folgerungsbegriff steht und für die in ihrer klassischen Form als „Meta-Prinzip“ die Wahrheitswert-Definitheit aller Aussagen gilt.

Im Folgenden wollen wir einen alternativen Zugang zur Logik skizzieren, der auf das rationale Argumentieren abhebt und aufzeigen, wie sie auf dialogischer Basis pragmatisch begründet werden kann.¹ Dieser Ansatz wurde vor allem durch Paul Lorenzen und Kuno Lorenz (u.a. [14]) ausgearbeitet; die damit vorgestellte Fundierung erlaubt einen präzisen schrittweisen Aufbau der Logik, wobei durch die Dialogsemantik Probleme der üblichen axiomatischen Vorgehensweise vermieden werden können.²

Dabei geht man von idealisierten Argumentationssituationen zwischen zwei Partnern aus, in denen Dialoge um die Geltung einer Behauptung nach allgemein anerkannten Redenormen in geregelter Weise geführt werden. Die Analogie zu Zweipersonenspielen liegt nahe, wobei die Spielzüge hier sprachliche Handlungen aus einem standardisierten Inventar von Ausdrucksmöglichkeiten sind. Die Untersuchung von „Games in Logic“ hat erst jüngst wieder große Popularität erlangt, wie neuere Arbeiten von Hintikka, van Benthem u.a.³ zeigen. Sie können in gewissem Sinn als eine Wiederaufnahme der scholastischen Obligationes-Dispute (vgl. [11], Kap. 16) verstanden werden, in denen bereits zwischen Rahmenbedingungen und Regeln unterschieden wird. Der pragmatische Charakter dieser Grundlegung der Logik wird dadurch unterstrichen, dass das Argumentieren konsequent als Ausführen von *Redehandlungen* bzw. Sprechakten gesehen wird: Kommunizieren ist Handeln. In der modernen Sprechakttheorie, die von Austin begründet und von Searle u.a. weiterentwickelt wurde, wird dies verallgemeinert, indem sie für jede sprachliche Äußerung drei Facetten konstatiert: Aussage oder Proposition (*was?* — „Lokution“), Redeabsicht oder Intention (*wozu?* — „Illokution“) und beabsichtigte Wirkung (*warum?* — „Perlokution“).

Die Rahmenbedingungen für derartige Dialogspiele hat Christian Thiel in folgender Weise charakterisiert [16]: „Die Ausgangsnorm, aus der die Normen der Dialogführung schrittweise be-

¹Dass die in der ursprünglichen Form vorgelegte Ausarbeitung im Hinblick auf eine pragmatische Begründung weiterer Differenzierungen bedarf, worauf u.a. Gethmann [7] Kap. 1.2 hingewiesen hat, kann an dieser Stelle nicht weiter vertieft werden.

²Als derzeit aktuellste und umfassende Einführung in die dialogische Logik wird das Lehrbuch von Inhetveen [10] empfohlen.

³s. z.B. [2, 3, 4, 9]. Einen Überblick über neueste Entwicklungen in der dialogischen Logik gibt [15].

gründet werden, besteht in einer „Fairnessbedingung“, genauer: in der Forderung nach *Gleichberechtigung* der beiden Partner, ohne die das Ergebnis eines Dialogs nicht für beide Partner als verbindlich angesehen werden kann, und ohne die es also, da die beiden ja stellvertretend für *beliebige* Partner stehen, keinen Anspruch auf Allgemeingültigkeit hätte. Zur Gleichberechtigung gehört auch, daß auf *jedes* Argument eines *jeden* der beiden Partner Rücksicht genommen wird, was sich dadurch erfüllen läßt, daß die beiden Dialogpartner ihre Argumente *abwechselnd* vorbringen, nach dem Argument des einen also stets der andere „am Zug“ ist. Beide erhalten dadurch auch das *Recht*, jedes Argument des anderen in Frage zu stellen — es „anzugreifen“ — und seine „Verteidigung“ zu verlangen. Solche Angriffsrechte sollen im Verlauf eines Dialogs nicht erlöschen können. Andererseits soll die Einlösung von Verteidigungspflichten eines der Partner durch die Wahrnehmung seiner Angriffsrechte nicht *aufgehoben*, sondern nur *aufgeschoben* werden dürfen. . . Und zwar genau so lange, bis der die Verteidigungspflicht schaffende Angriff des Dialogpartners seinerseits gegen einen etwaigen Angriff verteidigt worden ist. Eine etwas subtilere Überlegung ergibt dann, daß diese Bedingung nur erfüllt ist, wenn sich die Dialogpartner zur gemeinsamen Norm machen, daß an jeder Stelle ihres Dialogs das *zuletzt* angegriffene Argument stets *zuerst* verteidigt werden soll. Und schließlich muß etwas über das *Ziel* des Dialogs vereinbart werden. Da der Sinn der Argumentation in der Entscheidung darüber besteht, ob eine bestimmte vorgegebene Behauptung Anspruch auf allgemeine Zustimmung hat oder nicht, übernimmt es einer der Partner, diese Behauptung als These zu vertreten. Er fungiert dann im Dialog als *Proponent*, sein Partner als *Opponent* bezüglich der These. Die geforderte Entscheidung liefert der Dialog nur dann, wenn sich die Partner auf eine Regel einigen, die bestimmt, in welchem Fall die These als erfolgreich vertreten gelten soll und in welchem Falle nicht, in der Terminologie der Logik: wann der Proponent den geführten Dialog gewonnen, der Opponent also verloren haben soll und wann nicht. . . Wir werden aber nur dann von einer Behauptung sagen wollen, dass es eine schlüssige Argumentation für sie gebe, wenn sie als These in einem Dialog gegen *jeden* Opponenten verteidigt werden kann. . . Der Proponent wird also einer Strategie bedürfen, die ihm in einem Dialog um die fragliche These zu *jedem* Argument des Opponenten eine Fortsetzung des Dialogs erlaubt, die schliesslich für ihn zum Gewinn führt.“

2 Dialogische Logik

Mit diesem Aufriss des dialogischen Zugangs zur Logik soll deutlich werden, dass die formale Logik im Rahmen einer Theorie des schlüssigen Argumentierens begründet werden kann, ohne dass man auf die Plausibilität gewisser erster Sätze — seien es logische oder ontologische — zurückgreifen muss. An ihre Stelle tritt die pragmatische Einsicht in die Unumgänglichkeit gewisser erster Normen, die für jede rationale Argumentation gelten. Kurz gesagt geht es also um idealisierte Dialogsituationen, in denen um die Geltung einer Behauptung argumentiert wird, indem

- zwei Partner abwechselnd *Argumente* einführen,
- deren Form durch bestimmte *Partikelregeln* und
- deren Zulässigkeit durch bestimmte *Rahmenregeln* bestimmt ist.
- Dialoge werden als Zweipersonen (Gewinn-/Verlust-) Spiel über eine (logisch zusammengesetzte) Proposition, die *These*, durchgeführt und liefern damit ein Entscheidungsverfahren.

Die logischen (Verknüpfungs-) Zeichen werden pragmatisch durch Regeln zur Durchführung standardisierter idealer Sprachspiele eingeführt. Die Dialogschritte bestehen aus Behauptungen von Aussagen, die (quantoren-)logisch aus Elementaraussagen zusammengesetzt sind, und zwar zum einen aus Angriffen auf Behauptungen (Bezweifeln), zum anderen aus Verteidigungen gegen Angriffe. Die Partikelregeln normieren die Behandlung zusammengesetzter Aussagen im

Dialog, d.h. die Formen von Angriffen und Verteidigungen; die Rahmenregeln normieren Dialogverläufe, sie bestimmen die Zulässigkeit eines Dialogschritts in einer Dialogsituation. An dieser Stelle sei besonders hervorgehoben, dass der dialogische Ansatz am Zweiwertigkeitsprinzip festhält, aber in einer abgeschwächten Form, denn es gibt *nach endlich vielen Schritten immer genau einen Gewinner und einen Verlierer*, weshalb hier auch von *Dialogdefinitheit* gesprochen wird. Damit ist keine Entscheidung für eine bestimmte Logik vorweggenommen, denn dieser Aufbau der Logik, der von einer gemeinsamen Geltungskontrolle von Behauptungen ausgeht, kommt zunächst ganz ohne Wahrheitswerte aus.

2.1 Partikelregeln

Am Anfang stehen Elementaraussagen, durch die einem Gegenstand E ein Prädikator p zu- oder abgesprochen wird, ausgedrückt durch Primformeln der Form „ E ist p “: $E\epsilon p$ bzw. $p(E)$ (allgemein $E_1, \dots, E_n\epsilon p$) oder „ E ist nicht p “: $E\epsilon' p$. Zur Stabilisierung des Gebrauchs von Prädikatoren werden Normen eingeführt, die mehrere Prädikatoren zugleich betreffen, sog. *Prädikatorenregeln*:

- „Gehe von ' $x\epsilon p$ ' über zu ' $x\epsilon q$ '!“, in symbolischer Schreibweise $x\epsilon p \Rightarrow x\epsilon q$ (z.B. für p : Mensch, q : Lebewesen), oder
- „Gehe von ' $x\epsilon p$ ' über zu ' $x\epsilon' q$ '!“, symbolisch $x\epsilon p \Rightarrow x\epsilon' q$ (z.B. für p : Mensch, q : Pferd).

Seien nun zwei (affirmative) Prädikatorenregeln, $p(x) \Rightarrow q_1(x)$ und $p(x) \Rightarrow q_2(x)$ gegeben. Dann liegt es nahe, beide konjunktiv zu verknüpfen, so dass sich die kombinierte Prädikatorenregel $p(x) \Rightarrow q_1(x) \wedge q_2(x)$ ergibt. Damit dieser Schritt zulässig wird, muß eine Norm eingeführt werden, wie mit der Partikel „und“ zusammengesetzte Aussagen der Form $A \wedge B$ zu verteidigen sind, wenn sie bezweifelt bzw. angegriffen werden.

Habe also der *Proponent* \mathcal{P} $A \wedge B$ behauptet. Der *Opponent* \mathcal{O} bezweifelt die Aussage (greift die Aussage an), woraufhin der Proponent die Aussage zu verteidigen hat. Der Angriff besteht darin, dass \mathcal{O} von \mathcal{P} fordert, dass \mathcal{P} eine von \mathcal{O} ausgewählte Teilaussage aufnimmt und verteidigt. Nachdem \mathcal{P} dies getan hat, ist das ' \wedge ' in der neuen Situation verschwunden. Wenn also die Teilaussagen verteidigt wurden, wurde die Konjunktion verteidigt. Dies kann man tabellarisch in folgender Weise darstellen:

Es gibt zwei **Angriffs- / Verteidigungsregeln**:

Aussage	Angriff	Verteidigung
$A \wedge B$	$l?$	A
$A \wedge B$	$r?$	B

Bei einer Verknüpfung mit „oder“ hingegen steht es — aufgrund der ihr eigenen Unterbestimmtheit — dem Proponenten zu, auszuwählen, welche der beiden Teilaussagen er verteidigen will. Im Fall einer bedingten Aussage (Subjunktion) $A \rightarrow B$ ist der Proponent nur dann in der Pflicht, die Konklusion zu verteidigen, wenn der Opponent die Prämisse erfolgreich angreifen kann. Negierte Aussagen $\neg A$ kann der Opponent mit A angreifen, und \mathcal{P} hat keinen Verteidigungszug — er kann nur die Behauptung des Opponenten widerlegen. In beiden Fällen muss \mathcal{O} eine eigene Behauptung vorbringen, um einen wirksamen Angriff zu formulieren. Bei der Subjunktion darf \mathcal{P} die erforderliche Verteidigung so lange zurückstellen, bis ein von ihm vorgetragener Gegenangriff von \mathcal{O} abgewehrt ist.

Die Verwendung der Quantoren „für alle“ und „für ein“ setzt parametrisierte Aussagen und damit die Einführung von logischen Variablen x, y, \dots voraus, die für Elemente eines Variabilitätsbereichs stehen. Ein Allsatz mit der parametrisierten Aussage $A(x)$ wird dadurch angegriffen, dass der Opponent ein beliebiges Element n für x auswählen kann, worauf sich der Proponent mit $A(n)$ verteidigen muss. Im Fall einer Existenzaussage bleibt dem Opponenten nur

ein unspezifischer Angriff, denn die Auswahl des Elements n ist dem Proponenten vorbehalten. Damit ergibt sich für die Partikelregeln die folgende Übersicht:

Behauptung	Angriff \mathcal{O}	Verteidigung \mathcal{P}
$A \wedge B$	$l?$ $r?$	A B
$A \vee B$?	A B
$A \rightarrow B$?, A	B
$\neg A$?, A	—
$\bigwedge_x A(x)$	$n?$	$A(n)$
$\bigvee_x A(x)$?	$A(n)$

Wir sehen also, dass der Proponent nicht in jedem Fall verpflichtet ist, eine zusammengesetzte Behauptung in *allen* ihren Teilen zu verteidigen. Vielmehr kann \mathcal{P} die Begründung von Teilbehauptungen davon abhängig machen, dass \mathcal{O} zuvor andere Teilbehauptungen nachweist.

2.2 Allgemeine (effektive) Dialog-Rahmenregel

Zur Ausführung von Dialogspielen ist nun zu normieren, wie Dialoge über mehrfach zusammengesetzte Aussagen zu führen sind, d.h., wer wann einen Zug ausführen darf, was zulässige Angriffs- und Verteidigungszüge sind und wann eine Gewinnsituation vorliegt. Als allgemeine effektive Dialog-Rahmenregel wird festgelegt:

- \mathcal{P} darf **einen** vorhergehenden Zug von \mathcal{O} angreifen oder sich gegen \mathcal{O} s letzten Angriff verteidigen.
- \mathcal{O} darf den **letzten** Zug von \mathcal{P} angreifen oder sich gegen \mathcal{P} s letzten Angriff verteidigen.
- \mathcal{P} hat gewonnen, wenn er eine angegriffene Primformel verteidigt hat oder wenn \mathcal{O} eine angegriffene Primformel nicht verteidigen kann.

An dieser Stelle können wir auf eine detaillierte Begründung der Rahmenregel nicht weiter eingehen, die wichtigsten Gründe wurden aber bereits im einleitenden Abschnitt genannt.⁴ Die geringfügige Asymmetrie bei den Rechten für \mathcal{P} und \mathcal{O} ist in der Behandlung der (evtl. iterierten) Subjunktionen $A \rightarrow B$ begründet: Greift \mathcal{O} mit A an, so soll \mathcal{P} erst A angreifen und nach gelungener Verteidigung von A durch \mathcal{O} dann erst B verteidigen können. Das ist die bekannte Schlussregel „modus ponens“ auf der meta-dialogischen Ebene.

Damit sind wir in der Lage, effektive Dialogspiele zu durchzuführen. Als ein gerade auch für den Mathematikunterricht instruktives Beispiel greifen wir auf einen Satz der „Aufgabenlogik“ des russischen Mathematikers A.N. Kolmogorov (1932) zurück. Sie kann eher als Logik von Problemlösungen denn Logik von Aussagen angesehen werden und liefert wegen ihres konstruktiven Charakters besonders gute Beispiele für die effektive Logik. In ihr wird $A \rightarrow B$ als Zurückführung der Aufgabe B auf eine Lösung der Aufgabe A interpretiert, und $\neg A$ als Herleitung eines Widerspruchs aus der Annahme, es liege eine Lösung der Aufgabe A vor (nach [17]). Dann gilt z.B.

$$(A \rightarrow B) \rightarrow (\neg B \rightarrow \neg A),$$

weil zum einen das Antecedens („Prämisse“), die Lösung von B auf eine Lösung von A zurückzuführen, selbst eine Aufgabe ist. Auch das Consequens („Konklusion“), die Herleitung eines Widerspruchs aus der angenommenen Lösung von A auf die Herleitung eines Widerspruchs aus einer angenommenen Lösung von B zurückzuführen, ist eine Aufgabe. So stellt [17]

⁴Zu Details siehe z.B. [12], S. 260ff.

OPPONENT	PROPONENT
(2): ?, $a \rightarrow b$	(1): THESIS: $a \rightarrow b \rightarrow (\sim b \rightarrow \sim a)$
(4): ?, $\sim b$	(3): $\sim b \rightarrow \sim a$ [[2]]
(6): ?, a	(5): $\sim a$ [[4]]
(8): b	(7): ?2, a
(10): %%%	(9): ?4, b

Abbildung 1: Beweis eines Satzes der Kolmogorovschen Aufgabenlogik durch DiaLogic.

fest, dass sich die Lösung der zweiten Aufgabe auf eine der ersten zurückführen lässt: „Wenn aus einer Lösung von A eine Lösung von B , und aus der Annahme des Vorliegens einer Lösung von B ein Widerspruch konstruiert werden kann, so ist die Aneinanderreihung dieser beiden Konstruktionen zugleich die Konstruktion eines Widerspruchs aus der Annahme des Vorliegens einer Lösung von A — und also ein „Beweis“ von $\neg A$.“

Die Allgemeingültigkeit dieser These soll nun anhand eines effektiven junktorenlogischen („aus-sagenlogischen“) Dialogspiels gezeigt werden:

\mathcal{O}	\mathcal{P}
(1)	$(a \rightarrow b) \rightarrow (\neg b \rightarrow \neg a)$
(2) ?, $a \rightarrow b$	$\neg b \rightarrow \neg a$
(3) ?, $\neg b$	$\neg a$
(4) ?, a	?
(5) [a]	?2, a
(6) b , ?	[a]
(7) b	?
(8) [b], ?	?3, b
(9) ?	[b]

In Zeile (2) und (3) dieses Dialogs hätte der Proponent \mathcal{P} auch das unmittelbar vorausgegangene Argument des Opponenten \mathcal{O} angreifen können, jedoch nur dadurch, dass er eine Elementaraussage vorbringt. Er hätte aber verloren, wenn er diese nicht inhaltlich begründen kann. Er wählt daher die Verteidigung und ist auf der sicheren Seite. In Zeile (5) ist \mathcal{O} genötigt, a inhaltlich zu begründen (angedeutet durch die eckigen Klammern) — sonst hat er verloren. Nun kann aber \mathcal{P} mit a das Argument in Zeile (2) angreifen. \mathcal{O} kann daraufhin einen Aufschub des Vorbringens der Verteidigung b zugunsten des Zweifels an a bewirken (6). Diesem Zweifel kann \mathcal{P} leicht begegnen, indem er [a], also die Begründung von a , die \mathcal{O} in Zeile (5) bereits geliefert hat, übernimmt. Hier kann aber \mathcal{O} der Subjunktionsregel gemäß b vorbringen (7), was \mathcal{P} bezweifelt und damit \mathcal{O} zu einer inhaltlichen Begründung von b nötigt (8). \mathcal{P} greift nun mit b Zeile (3) an, in der \mathcal{O} ja mit $\neg b$ angegriffen hatte. Den Angriff des Opponenten (9) kann \mathcal{P} durch Übernahme von [b] kontern und hat damit gewonnen. Das Beispiel zeigt zugleich, dass \mathcal{P} das Spiel in jedem Fall zu seinen Gunsten entscheiden kann, auch wenn \mathcal{O} die beiden Elementaraussagen a und b begründen kann. Die These gilt also unabhängig vom Inhalt, d.h. vom Wahrheitswert von a und b , sie ist, wie man sagt, *allgemeingültig* bzw. *formal* oder *logisch wahr*. Sätze mit dieser Eigenschaft können als gültige Aussageschemata gelesen werden und sie werden auch „Tautologien“ genannt.

Den auf die formalen Aspekte reduzierten Beweis, wie er automatisch von DiaLogic, der Erlanger Implementierung der dialogischen Logik (s. Abschnitt 4) erzeugt wird, zeigt Abb. 1.

Mit der Definition, dass eine Behauptung dann (*material*) *wahr* heißen soll, wenn sie gegen jeden denkbaren Opponenten verteidigt werden kann, haben wir erstmals einen *Wahrheitsbegriff*,

und zwar auf diskursiver Basis, eingeführt. Materiale Wahrheit ist damit an die Existenz einer Gewinnstrategie für \mathcal{P} gebunden, mit deren Befolgung er gegen jeden Zweifler gewinnen kann. Sie ist es, die benötigt wird, um für das logische Schließen wahre Prämissen identifizieren zu können. Um von der materialen zur *formalen Logik* zu gelangen, müssen wir noch einen Schritt weiter gehen: Zusätzlich zum Bestehen einer Gewinnstrategie wird von dieser gefordert, dass \mathcal{P} nur solche Elementaraussagen behauptet, die \mathcal{O} schon zuvor behauptet hat. Damit entfällt die Überprüfung der Elementarsätze auf ihre Wahrheit, und damit ist die Gewinnstrategie von ihnen inhaltlich unabhängig.

Aus der Tradition bekannte logische Prinzipien wie das „Prinzip des ausgeschlossenen Widerspruchs“ begegnen uns an dieser Stelle wieder als allgemeingültige Aussageschemata. Damit ist dieses Prinzip nicht länger ein „Grundgesetz des Denkens“ oder ein „Axiom“, sondern schlicht ein wahrer Satz, wie im folgenden Dialog gezeigt wird:

	\mathcal{O}		\mathcal{P}
(1)			$\neg(a \wedge \neg a)$
(2)	$?, a \wedge \neg a$		1?
(3)	a		?
(4)	$[a]$		r? 2
(5)	$\neg a$		$?, a$

Auf das „tertium non datur“ $a \vee \neg a$ trifft dies jedoch nicht zu: Der Proponent hat bei der Verteidigung die Wahl zwischen a und $\neg a$. Im ersten Fall verliert er, wenn er a nicht beweisen kann, und gewinnt, wenn a wahr ist. Wählt er $\neg a$, so verliert er, wenn der Opponent die Behauptung a beweisen kann, d.h., er gewinnt nur, wenn a falsch ist. Es gibt also keine konstruktive Gewinnstrategie — allerdings eine klassische!⁵

2.3 Ein quantorenlogisches Dialogspiel mit der effektiven Rahmenregel

Das folgende Beispiel möge die Verwendung der Regel für den Allquantor illustrieren. Er verallgemeinert ja die Festlegung dessen, was als „Beweis“ einer Konjunktion gelten soll, auf beliebig, ja unendlich viele Aussagen dadurch, dass der Proponent verpflichtet wird, eine Aussage $a(x)$ für jedes vom Opponenten gewählte n zu beweisen. Verfügt \mathcal{P} über eine Strategie, in der er \mathcal{O} zwingt, $a(n)$ zuvor zu behaupten, so dass er dieses dann übernehmen kann, ist die allquantifizierte Aussage allgemeingültig.

	\mathcal{O}		\mathcal{P}
(1)			$\neg\neg \bigwedge_x a(x) \rightarrow \bigwedge_x \neg\neg a(x)$
(2)	$\neg\neg \bigwedge_x a(x)$		$\bigwedge_x \neg\neg a(x)$
(3)	$? n$		$\neg\neg a(n)$
(4)	$\neg a(n)$		$\neg \bigwedge_x a(x)$
(5)	$\bigwedge_x a(x)$		$? n$
(6)	$a(n)$		$a(n)$

In Zeile (2) antwortet \mathcal{O} gemäß der Regel für \rightarrow mit dem Antecedens und \mathcal{P} dann mit dem Consequens. In Zeile (3) greift \mathcal{O} dann \mathcal{P} (2) mit n an und \mathcal{P} antwortet durch Einsetzen von n . Zeile (4) zeigt den Angriff von \mathcal{O} auf \mathcal{P} (3) und \mathcal{P} antwortet mit einem Gegenangriff auf \mathcal{O} (2). \mathcal{O} kontert in (5), was \mathcal{P} mit einem Angriff mit genau demselben n beantwortet, das \mathcal{O} in (3) benutzt hatte. Damit ist \mathcal{O} gezwungen (5), mit $a(n)$ zu antworten, was \mathcal{P} übernimmt und damit gewinnt. Wir sehen also, dass \mathcal{P} nur seine zuletzt aufgestellte Behauptung verteidigen muss, \mathcal{O} aber alle bisher aufgestellten Behauptungen.

⁵Zur Ausführung mit dem DiaLogic-Programm s. Abschnitt 4.6

den. Legen wir darauf Wert, Schlussregeln zu finden, die nie auf Existenzaussagen oder Adjunkte als Konklusionen führen, ohne daß diese nicht schon effektiv in den Prämissen enthalten sind, d.h. wenn wir keine Widerspruchsbeweise ohne explizite Konstruktion zulassen wollen, benötigen wir die konstruktive Logik. An dieser Stelle sei noch einmal an Kolmogorovs Deutung als Aufgabenlogik erinnert. Dies ist nicht nur von didaktischem Interesse, sondern auch für viele praktische Anwendungen der Mathematik und Informatik von besonderer Relevanz. Oft werden in diesem Zusammenhang mehrwertige Logiksysteme in Spiel gebracht, was jedoch auf einem Mißverständnis beruht: Die mangelnde Kenntnis eines Wahrheitswerts ist kein dritter Wahrheitswert.

2.5 Gewinnstrategien

Beim systematischen Aufbau der Logik haben wir mit inhaltlichen („materialen“) Dialogen begonnen, bei denen der Erfolg letztlich davon abhängt, dass man für die Primaussagen Rechtfertigungen angeben kann. Nun lenken wir unser Interesse auf solche Aussageformen, deren Wahrheit davon nicht abhängt, für die also die Möglichkeit des Gewinns für jede Aussage dieser Form garantiert werden kann. M.a.W., sie sollen allein aufgrund ihrer Form gegen jeden Opponenten verteidigbar sein und werden daher *logisch wahr* oder *allgemeingültig* genannt. Wir suchen demnach Gewinnstrategien für geschlossene Formeln, d.h. solche, in denen keine Objektvariable frei vorkommt.

Dazu ist festzuhalten:

1. Behauptet \mathcal{O} eine abgeschlossene Primformel, so darf \mathcal{P} dies nicht bezweifeln (sie könnte im speziellen Fall von \mathcal{O} beweisbar sein).
2. \mathcal{P} hat gewonnen, wenn er eine abgeschlossene Primformel zu verteidigen hat, die \mathcal{O} schon vorher behauptet hat (von jeder anderen Primaussage könnte \mathcal{P} im speziellen Fall den Beweis gerade nicht kennen).

Eine Aussage ist (material) *wahr*, wenn es für den Proponenten einen Weg gibt, einen Dialog derart zu führen, daß er gegen jeden Opponenten gewinnt, er also über eine **Gewinnstrategie** verfügt. Wenn es eine Gewinnstrategie für den Opponenten gibt, ist die Aussage falsch.

Eine Aussage ist *formal wahr*, wenn es eine Gewinnstrategie für den Proponenten gibt, derart, dass er jede Primformel, die er zu verteidigen hat, vom Opponenten übernehmen kann. In diesem Fall sprechen wir von einer formalen Gewinnstrategie.

Damit können wir folgende Regel für formale Dialoge formulieren: In Dialogen über formale Wahrheit

- dürfen die Primformeln des Opponenten nicht angegriffen werden, und
- darf der Proponent nur solche Primformeln behaupten, die der Opponent zuvor behauptet hat.

Wenn der Proponent eine Gewinnstrategie für den Dialog über eine Anfangsposition $(A_1, \dots, A_n) \mid B$ hat, spricht man von einem *logischen Schluss (Implikation)* von einer Folge von Prämissen A_1, \dots, A_n auf eine Konklusion B . Implikationen sind Relationen über Aussageschemata (Aussageformen). Das Implikationszeichen \prec ist kein neuer Partikel, sondern ein meta-logisches Zeichen, um über Aussageformen zu sprechen — und *nicht*, um Aussageformen zu kombinieren.

Es gilt der folgende Satz: Eine Implikation $A_1, A_2, \dots, A_n \prec B$ ist gültig gdw. die Aussageform $A_1 \wedge A_2 \dots \wedge A_n \rightarrow B$ allgemeingültig ist.

Eine Gewinnstrategie kann als spezieller *Dialogbaum* dargestellt werden:

- er enthält alle Wahlmöglichkeiten des Opponenten,
- er enthält jeweils eine Wahlmöglichkeit des Proponenten, und
- alle Zweige des Dialogbaums werden vom Proponenten gewonnen.

Was also sind die Regeln für Gewinnstrategien? Wie schon durch den Hinweis auf Dialogbäume nahegelegt, ist zur Beantwortung dieser Frage ein geringfügiger Perspektivwechsel von Vorteil: Anstelle der interaktiven formalen Spiele zwischen Proponent und Opponent betrachten wir Entwicklungsmöglichkeiten von „Spielpositionen“.

2.6 Formale Dialoge und Tableaux

Mit dem Übergang von interaktiv geführten formalen Dialogspielen zu Dialogbäumen, in der gewohnten Einteilung in Proponenten- und Opponentenseite auch als „Tableau“ dargestellt, führen wir einen Entwicklungskalkül ein. Unter einem Kalkül wird ein Verfahren zur Herstellung von Formeln verstanden, das nur mit schematischen Regeln arbeitet. Die Regeln dieses Entwicklungskalküls sind genau diejenigen, die wir suchen.

An dieser Stelle sei ein — stark verkürzter — historischer Exkurs in die von Frege begründete Beweistheorie der modernen formalen Logik gestattet: David Hilbert hatte einen Kalkül entwickelt, mit dem von einigen wenigen ersten logisch wahren Sätzen („Axiomen“) ausgehend mithilfe einer einzigen Schlussregel, des *modus ponens* $A, A \rightarrow B \vdash B$, alle logisch wahren Sätze abgeleitet werden können. Ist nun als Beweisziel ein Satz gegeben, so besteht die große Kunst darin, eine Folge von geschickten Einsetzungen von Aussageformen in die Axiome (und die inzwischen abgeleiteten Sätze) zu finden, so dass am Ende dieser Folge von Ableitungsschritten der zu beweisende Satz steht, sofern er logisch wahr ist. Um eine näher am mathematischen Beweisen liegende Vorgehensweise zu ermöglichen, hatte dann Gerhard Gentzen Kalküle des „natürlichen Schliessens“ vorgelegt, die stattdessen für jeden logischen Partikel Schlussregeln (sog. Einführungs- und Beseitigungsregeln) bieten. Das Problem, geeignete Einsetzungen zu finden, bestand jedoch nach wie vor; erst mit den Tableauekalkülen von Beth, die die Beweisrichtung umkehren, wurde hierfür eine elegante Lösung gefunden. Dabei beginnt man mit dem zu beweisenden Satz und zerlegt ihn schrittweise in Teilformeln, bis man in allen Zweigen bei Primformeln angelangt ist. Die einzelnen Schritte werden lizenziert durch in umgekehrter Richtung angewendete Regeln vom Gentzenschen Typ. Ohne auf Einzelheiten eingehen zu können,⁹ halten wir fest, dass der Tableauekalkül ein Entscheidungsverfahren ist, das das Problem der Erfüllbarkeit löst. Ist eine Formel erfüllbar, führt das Verfahren konstruktiv zu einem Modell der Formel. Die Grundidee besteht darin, das Modell schrittweise aufzubauen, indem man die Formel untersucht und sie „top-down“ zerlegt. Das Verfahren betrachtet exhaustiv alle Möglichkeiten, so daß es eventuell beweisen kann, daß für unerfüllbare Formeln kein Modell gefunden werden konnte.

Somit kann das Bethsche Tableauverfahren als eine Variante des dialogischen Beweisens in der Form eines Entwicklungskalküls gesehen werden: Die gesuchten Regeln für Gewinnstrategien entsprechen den Regeln des Gentzenschen Sequenzenkalküls (genauer: Gentzens G3) — in umgekehrter Richtung gelesen! Im Detail sind dies die folgenden *Entwicklungsschritte für Gewinnstrategien* (vgl. die obige Einführung der Partikelregeln!):

Bezeichne Σ die von Opponenten bisher gesetzten Aussagen; kommt eine Aussage A in Σ vor, so sei dies durch $\Sigma[A]$ verdeutlicht.

⁹s. dazu und zu den klassischen Korrektheits- und Vollständigkeitsbeweisen z.B. [13], Kap. I.3.

$$\begin{array}{ccc}
\Sigma \left\| \begin{array}{l} A \wedge B \\ A \mid B \end{array} \right. & \Sigma[A \wedge B] \left\| \begin{array}{l} C \\ A \end{array} \right. & \Sigma[A \wedge B] \left\| \begin{array}{l} C \\ B \end{array} \right. \\
\Sigma \left\| \begin{array}{l} \bigwedge_x A(x) \\ A(n) \text{ f.a. } n \end{array} \right. & & \Sigma[\bigwedge_x A(x)] \left\| \begin{array}{l} C \\ A(n) \end{array} \right. \\
\Sigma \left\| \begin{array}{l} A \vee B \\ A \end{array} \right. & \Sigma \left\| \begin{array}{l} A \vee B \\ B \end{array} \right. & \Sigma[A \vee B] \left\| \begin{array}{l} C \\ A \mid B \end{array} \right. \\
\Sigma \left\| \begin{array}{l} \bigvee_x A(x) \\ A(n) \end{array} \right. & & \Sigma[\bigvee_x A(x)] \left\| \begin{array}{l} C \\ A(n) \text{ f.a. } n \end{array} \right. \\
\Sigma \left\| \begin{array}{l} \neg A \\ A \mid \perp \end{array} \right. & & \Sigma[\neg A] \left\| \begin{array}{l} C \\ A \end{array} \right. \\
\Sigma \left\| \begin{array}{l} A \rightarrow B \\ A \mid B \end{array} \right. & & \Sigma[A \rightarrow B] \left\| \begin{array}{l} C \\ \mid B \end{array} \right. \left\| \begin{array}{l} C \\ A \mid \end{array} \right.
\end{array}$$

Ein Zweig, der eine Primformel a als Links- **und** Rechtsformel enthält, heißt formal-abgeschlossen. Logische Wahrheit ergibt sich damit aus dem formalen Abschluss.

Als Beispiel sei die formale Gewinnstrategie für die klassische Schlussregel des modus ponens angegeben, also für das Schema des logischen Schlusses $A, A \rightarrow B \prec B$, in der Form eines Hypothesendialogs:

$$\begin{array}{ccc}
(0) & A & \left\| \right. \\
(1) & A \rightarrow B & \left\| \right. B \\
\hline
(2) & ? & \left\| \right. \dots \\
(3) & B & \left\| \right. [A] ?_1 \\
(4) & & \left\| \right. [B] 2
\end{array}$$

Im Entwicklungskalkül würde dabei Zeile (2) entfallen.

Abschliessend wollen wir noch kurz auf das Verhältnis der dialogischen Begründung der Logik zur traditionellen Vorgehensweise eingehen. In Letzterer wird ein logischer Ableitungsbegriff mit Hilfe von Ableitungsregeln und Axiomen definiert. Um ihn zu rechtfertigen, zieht man den innerhalb einer Interpretationssemantik definierten Folgerungsbegriff heran und zeigt, dass Ableitbarkeitsbegriff und Folgerungsbegriff äquivalent sind (Korrektheit und Vollständigkeit). Allerdings besteht das Problem, dass Interpretationssemantik im Stile Tarskis das Grundlegungsproblem der Logik nicht lösen kann, weil sie auf der Metaebene voraussetzt, was erst begründet werden soll — sowohl bei Partikeln als auch bei der Folgerungsbeziehung. Ein solcher Zirkel kann dadurch vermieden werden, dass man die Namenstheorie der Bedeutung durch eine Gebrauchstheorie ersetzt. Insbesondere gilt auch für die Junktoren und Quantoren, dass ihre Bedeutung ihr Gebrauch in der Sprache ist, eben so, wie er durch die dialogischen Verwendungsregeln expliziert wird. Wie Hinst [8] gezeigt hat, können die für metalogische Untersuchungen relevanten Anteile einer Interpretationssemantik als Strukturtheorie rekonstruiert werden. Diese Sichtweise wird u.a. von dem bedeutenden Logiker Arthur Prior geteilt, der gleichermassen zwar eine modelltheoretische Fundierung der Logik ablehnte, aber dennoch der Modelltheorie einen wichtigen Platz bei der Untersuchung logischer Kalküle einräumt.

3 Dialogische Logik und Beweisen im Mathematikunterricht

Dass die Dialogische Logik nicht nur ein wichtiger Beitrag zur Grundlegung der Logik, sondern auch von eminenter didaktischer Bedeutung für die Rekonstruktion mathematischer Argumentationen ist, wurde von mehreren Autoren betont. Bei der dialogischen Beweisführung dürfen keine impliziten Annahmen gemacht werden, vielmehr sind alle Annahmen explizit als Hypothesen einzuführen. Sie erzwingt eine einheitliche, präzise Notation; die saubere Symbolisierung hilft, sämtliche Zwischenschritte im Dialog klar zu machen. Insbesondere werden alle Fallunterscheidungen klar herausgestellt. In der mathematischen Prosa nicht selten nur informell dargestellte Beweisschritte müssen vollständig ausgearbeitet werden. Somit erzwingt die dialogische Rekonstruktion mathematischer Beweise ihre Vervollständigung. Als ein besonders instruktives Beispiel möge die von Rüdiger Inhetveen durchgeführte dialogische Ausarbeitung eines Lehrbuchbeweises aus der Analysis dienen, der von Richard Courant stammt. Sie benennt klar und deutlich diejenigen Schritte, die zur Erklärung des Beweises im Unterricht auszuführen sind. Da der Text (aus [10], S. 213–215) für sich selbst spricht, erübrigt sich an dieser Stelle jeder weitere Kommentar. Er ist vollständig im Anhang mit Genehmigung des Autors abgedruckt.

4 Die Erlanger DiaLogic-Implementation

Schon lange war ein computergestütztes Beweissystem für die Dialogische Logik ein Desiderat. Noch auf Lorenzens Anregung hin wurde eine erste Implementation in Austin angefertigt, eine weitere durch Gerrit Haas in Erlangen. Beide waren weder vollständig noch portabel. Daher haben wir in den 1990er Jahren am Erlanger Informatikinstitut eine Diplomarbeit ausgeschrieben, mit dem Ziel, ein Beweissystem für konstruktive und klassische Beweise unter Einschluss der Modallogik zu schaffen. Die Implementation wurde von Jürgen Ehrensberger ausgeführt [5] und in der Folge mehrfach von Claus Zinn und dem Autor überarbeitet (s.a. [6]).

Das Beweissystem DiaLogic kann in zwei Modi betrieben werden: Im interaktiven Modus, in dem das System die Rolle eines Dialogpartners übernimmt, können Dialogspiele durchgeführt werden, die das praktische Erlernen der dialogischen Logik ermöglichen. Über eine allgemeine Einübung in das kritische Argumentieren hinausgehend kann damit das konstruktive Beweisen an mathematischen Beispielen geübt werden. Im automatischen Modus arbeitet das System wie ein Tableau-Beweiser, der Beweise selbständig durchführt; damit kann es zur Überprüfung von Beweisen dienen. Beweise können konstruktiv und klassisch in der Junktorenlogik und der Quantorenlogik („Prädikatenlogik 1. Stufe“) durchgeführt werden; eine Erweiterung auf die Modallogik ist möglich.¹⁰

Die Implementation von logischen Deduktionssystemen, d.h. Beweisprozeduren, gehört zum Fachgebiet der „Computational Logics“ — wofür es bedauerlicherweise keine griffige deutsche Bezeichnung gibt. Dabei werden üblicherweise drei Abstraktionsebenen unterschieden: Zuerst geht es um die Definition der formalen (logischen) Sprache, also ihrer Syntax und Semantik, wodurch ihre Ausdruckskraft festgelegt wird. Auf der zweiten Ebene wird das Inferenzproblem behandelt, also die logischen Schlüsse, ihre Entscheidbarkeit und die Komplexität des Problems untersucht. Zuletzt — und dies ist der Gegenstand der folgenden Ausführungen — sind dann

¹⁰Das DiaLogic-System ist frei erhältlich und kann über die URL <http://www8.informatik.uni-erlangen.de/IMMD8/staff/Goerz/DiaLogic/index.html> heruntergeladen werden. Es ist in Scheme, einem Dialekt der Programmiersprache LISP, implementiert; zum Einsatz der graphischen Schnittstelle ist Eric Gallios Scheme-System STK (Version 4.0.1) erforderlich, das in Versionen für Windows, Mac OS X, und Linux verfügbar ist. Wir empfehlen dringend die Benutzung der Berkeley-Variante UCB Scheme, version 4.0.1-ucb1.16, die frei erhältlich ist über die URL <http://inst.eecs.berkeley.edu/~scheme/>. Sie enthält alles, was benötigt wird — einschliesslich der portablen Scheme-Programmbibliothek SLIB (!) — und ist sehr einfach zu installieren. Neben der graphischen Schnittstelle kann DiaLogic auch in jedem anderen standardkonformen Scheme-System in einem zeilenorientierten Modus betrieben werden.

spezielle Problemlösungsprozeduren zu erarbeiten, ihre Korrektheit und Vollständigkeit zu untersuchen sowie die Frage, inwieweit sie asymptotisch die Komplexität des Inferenzproblems erreichen.

Was ist nun die Aufgabe eines Beweissystems für die Dialogische Logik? Sein Ziel besteht im Auffinden von Gewinnstrategien — automatisch und unter Kontrolle des Benutzers. Dabei ist aber die Einschränkung zu beachten, dass nur endliche Dialogbäume betrachtet werden können. Unendlich lange Dialoge entstehen durch Wiederholung von Angriffen und Verteidigungen, somit ist die Zahl der Wiederholungen zu beschränken. Unendliche Verzweigungen entstehen bei der Wahl von Termen, so dass zu ihrer Vermeidung ein Verfahren zur Termwahl vorzusehen ist.

Was die *Rahmenregeln* betrifft, ist das oben Gesagte in folgender Weise zu präzisieren:

- **Beginn eines Dialogs:**
Zu Beginn setzt der Opponent beliebig viele Hypothesen, der Proponent setzt eine These. Anschließend ziehen beide Spieler abwechselnd, wobei der Opponent beginnt.
- **Sprechakte:**
Jede Äußerung eines Spielers nach dem Setzen der These ist entweder ein Angriff auf eine zuvor gesetzte Behauptung des Dialogpartners oder eine Verteidigung auf einen Angriff des Dialogpartners.
- **Züge des Opponenten:**
Der Opponent darf stets nur auf die unmittelbar vorhergehende Äußerung des Proponenten reagieren.
- **Beantworten von Angriffen:**
Jeder Angriff auf eine Behauptung darf vom Proponenten höchstens einmal beantwortet werden.
- **Offene Angriffe:**
Angriffe, zu denen noch keine Verteidigung vorgebracht wurde, heißen offene Angriffe. Möchte sich der Proponent gegen einen offenen Angriff verteidigen, muß er den zuletzt gesetzten Angriff zuerst beantworten.
- **Primformeln:**
Primformeln dürfen nicht angegriffen werden. Der Opponent darf Primformeln stets behaupten. Der Proponent darf eine Primformel nur behaupten, wenn der Opponent sie zuvor behauptet hat.
- **Gewinn des Dialogs:**
Ein Spieler hat einen Dialog gewonnen, wenn sein Dialogpartner keine Zugmöglichkeit mehr hat.

Zunächst müssen nun die Entwurfsentscheidungen für das DiaLogic-System erklärt werden, die seine Besonderheiten ausmachen: Zum einen geht es um das genannte Verfahren zur Termwahl. Andererseits haben wir entschieden, wie in Programmiersprachen üblich, ein Typsystem einzuführen: Die Elemente der logischen Sprache gehören bestimmten Sorten an, für die eine Symbolverwaltung realisiert werden muss; insbesondere sind müssen alle vom Benutzer eingeführten sprachlichen Objekte deklariert werden. Es folgen einige Hinweise zur Notation von Formeln sowie zur Steuerung der Dialogspiele.

4.1 Termwahl und Unifikation

Termwahl ist erforderlich bei Angriffen auf Allaussagen und Verteidigungen von Existenzaussagen. Für die Termwahl des Proponenten folgen wir der Maxime, dass sie so lange verzögert werden soll, bis nur noch endlich viele Wahlmöglichkeiten existieren. Dies wird dadurch realisiert,

dass man keinen konkreten Term wählt, sondern stattdessen ein *Termsymbol*, und das Termsymbol an einen konkreten Term bei der Übernahme von Primformeln bindet.¹¹ Für die Termwahl des Opponenten soll gelten, dass stets der für den Proponenten schwierigste Fall gewählt wird: Es wird ein neues (eigenständiges) Termsymbol erzeugt, das nicht gebunden wird.

Beispiel:

$$\begin{array}{l}
 \mathcal{O} \\
 (1) \\
 (2) \quad ? \wedge_x P(x) \\
 (3) \quad \$1 ? \\
 (4) \quad P(\$2)
 \end{array}
 \parallel
 \begin{array}{l}
 \mathcal{P} \\
 \wedge_x P(x) \rightarrow \wedge_x P(f(x)) \\
 \wedge_x P(f(x)) \\
 \$2 ? \\
 \dots
 \end{array}$$

Primformelübernahme des Opponenten: $P(\$2)$ und des Proponenten: $P(f(\$1))$.

Nun sind die beiden Terme gleich zu machen, was durch das *Unifikationsverfahren* erfolgt: Mit $\$2 := f(\$1)$ werden die Terme gleich. Allgemein geht es darum, eine Substitution, den sog. „Unifikator“ θ zu finden, der die Terme p, q gleich macht:

$$\text{UNIFY}(p, q) = \theta, \text{ so dass } \text{SUBST}(\theta, p) = \text{SUBST}(\theta, q)$$

Die Unifikation ist also ein Test von (Syntax-)Bäumen auf Gleichheit; unter einer Substitution ist eine Assoziation von Termen mit Variablen zu verstehen. Ein Unifikator ist eine Substitution, die, auf zwei Ausdrücke angewandt, diese gleich macht. Auf Alan Robinson geht ein Algorithmus zurück, der den allgemeinsten Unifikator bestimmt.

Der vollständige Dialog ist somit:

$$\begin{array}{l}
 \mathcal{O} \\
 (1) \\
 (2) \quad ? \wedge_x P(x) \\
 (3) \quad \$1 ? \\
 (4) \quad P(f(\$1)) \\
 (5) \quad \%
 \end{array}
 \parallel
 \begin{array}{l}
 \mathcal{P} \\
 \wedge_x P(x) \rightarrow \wedge_x P(f(x)) \\
 \wedge_x P(f(x)) \\
 f(\$1) ? \\
 P(f(\$1))
 \end{array}$$

4.2 Verwendung von Sorten

Sorten (Typen) werden eingeführt, um die Überprüfung des korrekten Aufbaus von Termen und Formeln sowie der Variabilitätsbereiche der Quantoren effizient zu realisieren. Dazu ist zunächst erforderlich, die Sorten selbst durch eine eigene Deklaration einzuführen:

- `declare sort sortname`¹²

Weiterhin kann bei Bedarf eine Sortenhierarchie festgelegt und auch eine Universalsorte als ihr oberstes Element eingeführt werden:

- `sort sortname1 >= sortname2`
`universal sortname`

¹¹Zu einer ausführlichen Begründung für diese Vorgehensweise siehe [1], Kap. 5.2.2.

¹²DiaLogic kann auch so konfiguriert werden, dass anstelle der englischen Wortsymbole („declare“ etc.) deutsche benutzt werden („deklariere“ etc.).

Beispiel:

```
declare sort s1
declare sort s2
declare sort s3

sort s1 >= s2
sort s2 >= s3
sort s3 >= s1
```

4.3 Symbolverwaltung: Deklarationen

Alle Symbole, die in Thesen und Hypothesen verwendet werden, müssen deklariert werden. Die Deklaration von Symbolen für Terme und Primformeln ist in folgender Weise vorzunehmen:

- Variablensymbole
`declare variable x nat`
- Konstantensymbole
- Funktionssymbole
`declare function f (nat nat) -> nat prefix f`
`declare function + (nat nat) -> nat infix 2 3 +`
- Prädikatsymbole
`declare predicate P (nat) prefix P`

Weiterhin können Konstantenmengen, z.B. natürliche, ganze, rationale, reelle, oder komplexe Zahlen, sowie auch logische Symbole (Junktor-, Quantorsymbole) deklariert werden.

4.4 Formeln

Formeln treten im Dialog als Behauptungen auf; sie sind zusammengesetzt aus deklarierten Symbolen und werden erzeugt bei der Festlegung der Hypothesen und der These.

Beispiele:

```
hypothesis TND A OR NOT A
thesis ALL x (P(x) SUB P(f(x)))
```

4.5 Dialogspiel-Steuerung

Nach der Festlegung der Hypothesen und der These kann das Dialogspiel gestartet werden. Das System versucht, eine Gewinnstrategie (automatisch oder Benutzerspiel) zu finden, die dann — auch graphisch — ausgegeben werden kann. Der Modus wird in der graphischen Benutzungsschnittstelle über eine Menüauswahl eingestellt. Der Spielverlauf kann am Ende auch graphisch im PostScript-Format ausgegeben werden.

4.6 Steuerung des Dialogspiels

Zur Steuerung des Dialogspiels steht eine Reihe von Optionen zur Verfügung, die in der graphischen Benutzungsschnittstelle über ein Menü wählbar sind. Wird „Benutzerspiel“ gewählt,

arbeitet das System interaktiv, ansonsten automatisch — d.h., es übernimmt die Proponenten- und Opponentenrolle. Im ersteren Fall übernimmt der Benutzer die Rolle des Proponenten. Es kann gewählt werden, ob die Beweise mit den effektiven (konstruktiven) oder klassischen Rahmenregeln geführt werden sollen, es kann eine Angriffs- und Verteidigungsschranke (Anzahl der Schritte, s.o.) festgelegt werden — was oft für das Auffinden einer Gewinnstrategie entscheidend ist—, und es ist auch möglich, auf die Modallogik (konstruktiv oder klassisch) umzustellen.

Das folgende Beispiel für ein Dialogspiel (im zeilenorientierten Modus, der in der graphischen Benutzungsschnittstelle auch als Zusatzoption eingestellt werden kann) zeigt, dass das „tertium non datur“ konstruktiv nicht gilt. Werden aber die klassischen Rahmenregeln geladen und die Verteidigungsschranke entsprechend eingestellt, ist der Beweis erfolgreich:

```
>> declare prop-variable a a
==> predicate symbol: a declared
>> thesis a OR NOT a
==> thesis defined
>> start dialogue
==> PROOF FAILURE
>> load frame-rules klass
...
>> start dialogue
==> PROOF FAILURE
>> set defence-limit 2
==> defence-limit := 2
>> start dialogue
==> PROOF SUCCESS
>> write strategy
PRO (1): THESIS a || ~a
OPP(2): ?,
PRO(3): ~a [[2]]
OPP(4): ?, a
PRO(5): a [[2]]
OPP(6): %%%
```

Zur Demonstration der graphischen Benutzungsschnittstelle sei als zweites Beispiel das Dialogspiel aus Abschnitt 2.3 gezeigt (Abb. 2).

Bisher haben wir immer nur sehr einfache Beispiele betrachtet, was die Ansicht nahelegen mag, dass es sich bei DiaLogic nur um ein Spielsystem, wenn auch von didaktischem Wert, handelt. In der Tat können aber damit auch durchaus anspruchsvolle Beweise durchgeführt werden, die selbst für große professionelle automatische Beweiser eine beachtliche Herausforderung darstellen. Ein großes Beispiel, das mit DiaLogic bearbeitet wurde, ist der Satz, dass die Wurzel aus 2 irrational ist. Aus Platzgründen verzichten wir hier auf eine Darstellung dieses Beispiels, zumal es in [6] ausführlich vorgestellt wurde.¹³ Stattdessen sei abschliessend noch gezeigt, dass das DiaLogic-System auch geeignet ist, eine Brücke zur Einführung der Logikprogrammierung — üblicherweise mit der Programmiersprache PROLOG — zu bauen.

4.7 Logikprogrammierung mit DiaLogic

Dass die Programmiersprache PROLOG in ihrem „reinen“ Grundbestand („Pure PROLOG“) nur eine Teilsprache der Logik erster Stufe, die sog. Horn-Logik implementiert, sei als bekannt vor-

¹³Dieser Artikel sowie die Diplomarbeit von Ehrensberger [5] sind im pdf-Format in der Distribution des DiaLogic-Systems im Unterverzeichnis „docs“ enthalten.

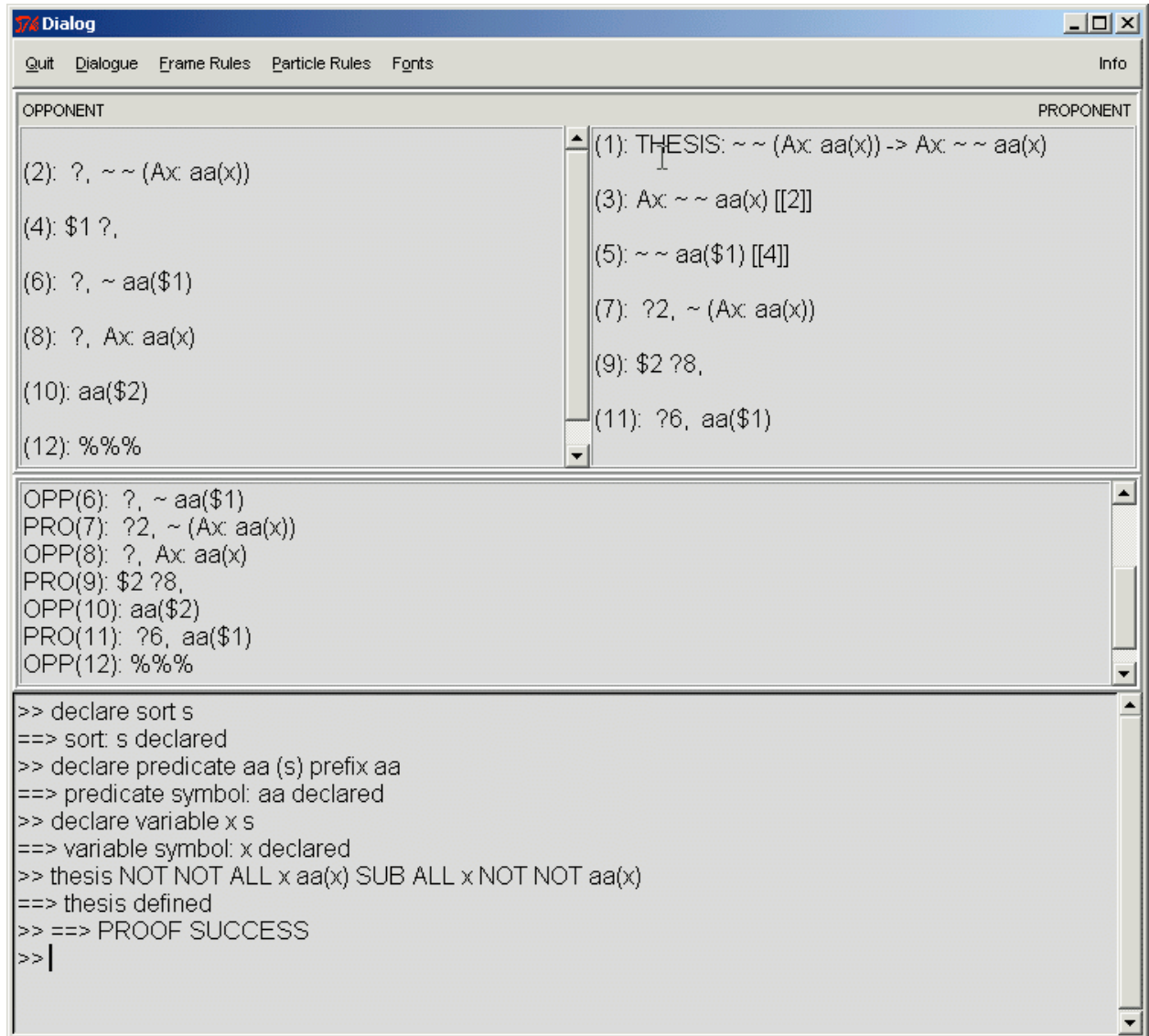


Abbildung 2: Quantorenlogisches Beispiel mit der effektiven Rahmenregel in DiaLogic.

ausgesetzt. Zudem besitzt sie mit der Implementation der Negation als Scheitern eines Beweises eine etwas andere Semantik — de facto ist PROLOG ein nichtmonotones Inferenzsystem. Aus der Sicht der Ausdruckskraft ist die DiaLogic zugrundeliegende Sprache mächtiger, da sie ja die ganze Logik erster Stufe — konstruktiv (effektiv) als auch klassisch — umfasst. Allerdings sind die PROLOG-Interpreter hochgradig optimiert, um eine schnelle und effiziente Programmausführung zu ermöglichen. Dennoch, so unsere Behauptung, eröffnet die dialogische Ausführung von Logikprogrammen eine neue Sichtweise: Beweise, die der PROLOG-Interpreter automatisch ausführt, können interaktiv als Dialogspiele schrittweise durchgeführt werden!

Zur Illustration dieser Behauptung sei die „Axiomatisierung“ der Addition mithilfe der Nachfolgerfunktion in „Pure PROLOG“ gewählt. Sie ist durch das folgende Programm gegeben:

```
add(0, Y, Y).
add(succ(X), Y, succ(Z)) :- add(X, Y, Z).
```

Dabei ist das die Nachfolgerfunktion repräsentierende succ ein uninterpretiertes Funktionssymbol, succ(0) kann nicht zu dem Term 1 ausgewertet werden!¹⁴

Die Aufgabe 0+1 wird in der Anfrage

```
?- add(0, succ(0), Result).
```

formuliert und liefert die Antwort:

```
Result = succ(0)
yes
```

Als zweites Beispiel sei die der Aufgabe 2+2 entsprechende Anfrage formuliert:

```
?- add(succ(succ(0)), succ(succ(0)), Result).
```

Sie resultiert in

```
Result = succ(succ(succ(succ(0))))
yes
```

Die Umsetzung dieses Logikprogramms in DiaLogic einschliesslich der Aufgabe 2+2 zeigt Abbildung 3. In der unteren Hälfte sind die Eingaben protokolliert, wobei die Eingabeaufforderung durch das Zeichen >> angezeigt wird. Die erste Eingabe ist die Deklaration der Sorte nat; sie wird in der Folgezeile bestätigt. Es folgen weitere Deklarationen für die verwendeten Symbole, die in PROLOG nicht notwendig sind. Den beiden PROLOG-Programmzeilen entsprechen die beiden Hypothesen, der Anfrage die These. Das Programm wird über das „Dialogue“-Menü gestartet, worauf das Dialogtableau in der oberen Hälfte der Abbildung ausgefüllt wird. Die konstruierte Lösung wird in Schritt (25) angezeigt.

5 „Intuitionismus und Informatik“

F.L. Bauer hat in der Rubrik „Historische Notizen“ im Augustheft 1999 der Zeitschrift Informatik-Spektrum unter dieser Überschrift ein Plädoyer zugunsten der „intuitionistischen“ Logik und Mathematik im Rahmen der Informatik veröffentlicht. In einer historischen Skizze der Diskussion über die Grundlagen der Mathematik rekurriert er auf den Brouwerschen Intuitionismus, auf den in der Tat die konstruktive Methode zurückgeht, aber er ignoriert dabei die moderne Entwicklung des Konstruktivismus. So kann es nicht verwundern, dass er die Problematik des „tertium non datur“, wie oben ausgeführt, nicht nennt. Stattdessen bemüht er — im Hinblick auf das Weglassen von Axiomen — eine fragwürdige Analogie zwischen der konstruktiven Logik und der nichteuklidischen Geometrie. Dennoch ist ihm darin Recht zu geben, dass die Informatik im Prinzip eine konstruktive Disziplin ist und daher die ihr zugrundeliegende Logik zuerst einmal eine konstruktive sein sollte. Bemerkenswert ist sein Hinweis: „Es wird Zeit, daß sich

¹⁴Die erste Zeile besagt in Infix-Schreibweise $0+Y=Y$, die zweite $X+Y=Z \rightarrow \text{succ}(X)+Y=\text{succ}(Z)$.



Abbildung 3: „Axiomatisierung“ der Addition mithilfe der Nachfolgerfunktion in DiaLogic.

die Informatik des Intuitionismus [d.h. Konstruktivismus; der Autor] annimmt. Dies kann heute emotionslos geschehen. . . “. Die Frage, ob aus Beweisen von Sätzen Konstruktionen der Objekte, d.h. Algorithmen gewonnen werden können, kann ohne Zweifel positiv beantwortet werden, wenn man sich auf konstruktive Beweise beschränkt.

Zugleich und darüber hinausgehend wäre eine logische Grundbildung¹⁵ in diesem Sinne eine sehr wünschenswerte Vorgabe für ein wissenschaftliches Studium nicht nur der Mathematik und Informatik, sondern auch aller anderen Fachrichtungen.

6 Anhang: Lehrbuchbeispiel aus der Analysis

Dieses Beispiel wurde von Rüdiger Inhetveen ausgearbeitet ([10], S. 213–215) und wurde von ihm dankenswerterweise zur Verfügung gestellt.¹⁶

In einem Lehrbuch der Differential- und Integralrechnung heißt es:

„Wenn eine unendliche Folge von Zahlen $a_1, a_2, a_3, \dots, a_n, \dots$ gegeben ist und wenn es eine weitere Zahl g gibt, derart, daß in jedes noch so kleine um g abgegrenzte Intervall alle Zahlen a_n mit Ausnahme von höchstens endlich vielen (d.h. von einem gewissen Wert des Index n an alle) hineinfallen, so sagen wir: die Zahl g ist der Grenzwert der Zahlenfolge $a_1, a_2, a_3, \dots, a_n, \dots$, oder: die Zahlenfolge a_1, a_2, \dots konvergiert gegen g ; in Zeichen:

$$\lim_{n \rightarrow \infty} a_n = g.$$

Aus diesen sieben Zeilen läßt sich zunächst durch den Gebrauch der logischen Notation leicht eine einzige machen (ein eigener Schritt zur Vereinbarung passender Abkürzungen entfällt hier natürlich), nämlich:

$$\bigvee_g \lim_{n \rightarrow \infty} a_n = g \quad \Leftrightarrow \quad \bigvee_g \bigwedge_{\varepsilon > 0} \bigvee_N \bigwedge_{n > N} |a_n - g| < \varepsilon$$

Unser zitierter Autor fährt dann nach wenigen Zeilen so fort:

„Es ist nützlich, sich zu vergegenwärtigen, daß jede konvergente Zahlenfolge beschränkt ist; d.h., zu jeder Zahlenfolge a_1, a_2, a_3, \dots , für welche ein Grenzwert ξ existiert, gibt es eine von n unabhängige positive Zahl M , so daß für alle a_n der Folge $|a_n| < M$ gilt. Aus unseren Definitionen folgt dieser Satz leicht: Sicher gibt es nämlich einen Index N , so daß für $n > N$ immer $|a_n - \xi| < 1$ ist. Unter den N Zahlen $|a_1 - \xi|, |a_2 - \xi|, \dots, |a_N - \xi|$ sei A die größte. Dann dürfen wir $M = |\xi| + A + 1$ setzen. Denn es ist sicher nach der Definition von A die Ungleichung $|a_n - \xi| < A + 1$ für $n = 1, 2, \dots, N$ erfüllt, während für $n > N$ gilt $|a_n - \xi| < 1 \leq A + 1$.“

Von der unnötigen Neueinführung des Buchstabens ξ für den eben noch g genannten Grenzwert wollen wir hier einmal absehen und bei dem g bleiben. Schauen wir uns den zitierten Beweis einmal an. Er beginnt mit einer Ungleichung, die sich darauf stützt, daß es „sicher einen Index gibt“.

¹⁵Vgl. die im Jahrgang 2004/2005 der Zeitschrift Informatik-Spektrum in sechs Teilen veröffentlichte Artikelserie „Informatik als Grundbildung“ von Wedekind, Inhetveen und Ortner, insbesondere [18].

¹⁶Abdruck mit expliziter Genehmigung des Autors.

Aus den gegebenen Definitionen ist das nicht direkt ersichtlich. Zwar stimmt die Behauptung, aber woher kommt sie und wieso beginnt der Beweis ausgerechnet mit ihr?

Für den geschulten Mathematiker verbirgt sich dahinter kein Geheimnis, dem Anfänger aber ... kann eine logische Analyse helfen! Er braucht die fragliche Behauptung nur sauber zu symbolisieren und nach den bekannten Regeln einen Dialog zu führen. Die *Voraussetzung* der Behauptung, daß nämlich eine konvergente Folge a_1, a_2, \dots vorliegt, ist gleichbedeutend mit der Aussage

$$\bigvee_g \lim_{n \rightarrow \infty} a_n = g,$$

und dies wiederum heißt (den Einsquantor über g habe ich der Kürze wegen weggelassen):

$$\bigwedge_{\varepsilon > 0} \bigvee_N \bigwedge_{n > N} |a_n - g| < \varepsilon.$$

Unter dieser Voraussetzung soll gelten:

$$\bigvee_{M > 0} \bigwedge_n |a_n| < M.$$

Damit ist die Ausgangsstellung des zu führenden Dialoges klar. Sie lautet

$$\bigwedge_{\varepsilon > 0} \bigvee_N \bigwedge_{n > N} |a_n - g| < \varepsilon. \quad \rightarrow \quad \bigvee_{M > 0} \bigwedge_n |a_n| < M.$$

Für diese Ausgangsstellung ist eine Gewinnstrategie zu finden. Das *kann* natürlich nicht allein mit logischen Mitteln gelingen. Im konkreten Fall müssen wir noch die Dreiecksungleichung zu Hilfe nehmen. Wir könnten sie in einer geeigneten Form als zusätzliche Hypothese des Opponenten in die Anfangsstellung aufnehmen. Hier wollen wir aber der Einfachheit halber den Proponenten dort, wo er die Dreiecksungleichung braucht, einfach damit rechnen lassen. Das ist in der folgenden Strategie in den Zeilen 6 und 8 der Fall. Die Strategie sieht dann so aus:

0	$\bigwedge_{\varepsilon > 0} \bigvee_N \bigwedge_{n > N} a_n - g < \varepsilon.$	$\bigvee_{M > 0} \bigwedge_n a_n < M.$
1	?(0)	...
2	$\bigvee_N \bigwedge_{n > N} a_n - g < \varepsilon_0.$	$\varepsilon_0 ?(0)$
3	$\bigwedge_{n > N_0} a_n - g < \varepsilon_0.$?(2)
4		$A := \max_{1 \leq \nu \leq N_0} a_\nu - g ;$ $M := A + g + \varepsilon_0$ $\bigwedge_n a_n < M. \quad (\uparrow 1)$
5	$n_1 ?(4)$	$ a_{n_1} < M$
6	?(5)	1. Fall: $1 \leq n_1 \leq N_0 :$ $ a_{n_1} \leq a_{n_1} - g + g \leq A + g =$ $= M - \varepsilon_0 < M$
7	$ a_{n_1} - g < \varepsilon_0$	2. Fall: $n_1 > N_0 :$ $n_1 ?(3)$
8		$ a_{n_1} \leq a_{n_1} - g + g < \varepsilon_0 + g \leq$ $\leq A + g + \varepsilon_0 = M$

Wie hängt nun dieser Dialog mit dem zitierten Beweis zusammen? Unser Autor beginnt mit Zeile 2: Er setzt (das darf er natürlich: er greift ja als Proponent einen Allquantor an) $\varepsilon_0 = 1$. Schon an dieser Stelle wird natürlich die Einsicht, daß am Ende eine Strategie vorliegt (d.h. daß ε_0 hier schematisch auftritt) ein wenig erschwert: Es ist ja nicht gerade üblich, die Zahl 1 schematisch zu verwenden. Dann beschreibt er den entsprechenden Opponenteneintrag in dieser Zeile mit den Worten: „Sicher gibt es nämlich einen Index N , so daß für $n > N$ immer $|a_n - \xi| < 1$ ist.“ Auch das ist eher irreführend als hilfreich, denn in Wirklichkeit reformuliert er eine Opponentenbehauptung, und das heißt für das weitere Vorgehen, daß er — der Proponent/Verfasser — diesen Existenzsatz als Voraussetzung benutzen darf. (Sicherlich soll der fragliche Existenzsatz daneben auch noch wahr sein, denn die Strategie des Proponenten soll ja nicht davon „leben“, daß der Opponent wegen dieser Behauptung aufgeben muß.) Im Text unseres Autors folgen jetzt die Definitionen, die der Proponent in Zeile 4 eingetragen hat (wobei M zwar *definiert*, aber nicht mehr *explizit verwendet* wird!). Danach geht er zu Zeile 6 bzw. 8 des Proponenten über, benutzt aber *nicht* die Dreiecksungleichung, sondern zwei sehr verkürzte Argumente, die man ihrerseits nur versteht, wenn man *doch* auf die Dreiecksungleichung zurückgreift, und ... hört dann einfach auf.

Die für alle n zu zeigende Ungleichung $|a_n| < M$ taucht im Beweis gar nicht auf. Die Frage, warum es genügt $\varepsilon_0 = 1$ zu betrachten, wird, wie gesagt, mit Stillschweigen übergangen. Und warum der Beweis an der zitierten Stelle fertig ist, das herauszufinden bleibt ebenfalls dem künftigen Leser überlassen. Der Dialog zeigt es aber deutlich: in diesen Zeilen stehen die (wegen der Fallunterscheidung zwei) Verteidigungen, die der Proponent auf den Angriff aus Zeile 5 noch schuldig ist, um eine Gewinnstellung zu erreichen. Wenn der Leser weder „geschulter Mathematiker“ ist, noch Kenntnisse in der Dialogführung besitzt, dann hat er vermutlich ein kleines Problem. Mit solchen Kenntnissen ist es jedoch gar nicht so schwierig, den Beweis als insgesamt doch korrekt zu rekonstruieren.

Literatur

- [1] Beckstein, C.: *Zur Logik der Logikprogrammierung. Ein konstruktiver Ansatz*, Nr. 199 in Informatik-Fachberichte, Springer-Verlag, Berlin and New York, 1988.
- [2] Benthem, J. v.: *Games in Logic*, in Hoepelman, J. P. (Hrsgb.): *Representation and Reasoning. Proceedings of the Stuttgart Workshop on Discourse Representation, Dialogue Tableaux, and Logic Programming*, Max Niemeyer Verlag, Tübingen, 1988, S. 3–15.
- [3] Benthem, J. v.: *Logical Constuction Games*, in *Fourth Georgian Autumn School in Logic, Language and Information*, Borjomi, December 2001, S. 1–12.
- [4] Benthem, J. v.: *What Logic Games are Trying to Tell Us*, Stanford University, Stanford, CA, December 2002.
- [5] Ehrensberger, J.: *Ein System für dialogische Logik*, Erlangen, 1996, Diplomarbeit.
- [6] Ehrensberger, J.; Zinn, C.: *DiaLog: A System for Dialogue Logic*, in *14th Conference on Automated Deduction, Townsville, North Queensland, Australia, July 13–17, 1997, Proceedings*, Bd. 1249 von *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, 1997, S. 446–460.
- [7] Gethmann, C. F.: *Protologik. Untersuchungen zur formalen Pragmatik von Begründungsdiskursen*, Theorie, Suhrkamp, Frankfurt am Main, 1979.
- [8] Hinst, P.: *Fundamentalsemantische Grundlegung der Logik und strukturtheoretische Rekonstruktion der Interpretationssemantik*, in Mittelstraß, J.; Riedel, M. (Hrsgb.): *Vernünftiges Denken. Studien zur praktischen Philosophie und Wissenschaftstheorie*, Walter de Gruyter, Berlin and New York, 1978, S. 52–70.
- [9] Hodges, W.: *Logic and Games*, in *Stanford Encyclopedia of Philosophy*, 2001.
- [10] Inhetveen, R.: *Logik – Eine dialog-orientierte Einführung*, Edition am Gutenbergplatz (Teubner), Leipzig, 2003.
- [11] Kretzmann, N.; Kenny, A.; Pinborg, J. (Hrsgb.): *The Cambridge History of Later Medieval Philosophy. From the Rediscovery of Aristotle to the Disintegration of Scholasticism, 1100–1600*, Cambridge University Press, Cambridge, 1982.

- [12] Lorenz, K.: *Basic Objectives of Dialogue Logic in Historical Perspective*, *Synthese*, Bd. 127, Nr. 1–2, 2001, S. 225–263.
- [13] Lorenzen, P.: *Lehrbuch der konstruktiven Wissenschaftstheorie*, Bibliographisches Institut, B.I. Wissenschaftsverlag, Mannheim and Wien and Zürich, 1987.
- [14] Lorenzen, P.; Lorenz, K.: *Dialogische Logik*, Wissenschaftliche Buchgesellschaft, Wiesbaden, 1978.
- [15] Rahman, S.; Keiff, L.: *On How to Be a Dialogician. A Short Overview on Recent Developments on Dialogues and Games*, in *Logic, Thought and Action*, Bd. 2 von *Logic, Epistemology, and the Unity of Science*, Springer, Berlin, 2005, S. 359–408.
- [16] Thiel, C.: *Wozu Geschichte der Logik?*, *Philosophisches Jahrbuch*, Bd. 79, 1. Halbband, 1972, S. 77–87.
- [17] Thiel, C.: *Über Ursprung und Problemlage des argumentationstheoretischen Aufbaus der Logik*, in Gethmann, C. F. (Hrsgb.): *Theorie des wissenschaftlichen Argumentierens*, Suhrkamp, Frankfurt am Main, 1980, S. 117–189.
- [18] Wedekind, H.; Inhetveen, R.; Ortner, E.: *Informatik als Grundbildung. Teil VI: Logik und Geltungssicherung*, in *Informatik Spektrum*, Springer-Verlag, Berlin und Heidelberg, Februar 2005, S. 48–52.