Peter Köpke, Bernhard Schröder

Natürlich formal

Die Linguistik profitiert in ihrer Theoriebildung seit den sechziger Jahren des vergangenen Jahrhunderts in einem kaum zu überschätzenden Maße von den Formalwissenschaften, namentlich der Theorie formaler Sprachen, der formalen Logik und der Mengen- und Modelltheorie bei ihrer Theoriebildung in Syntax, Semantik und zunehmend auch in der Pragmatik. Die sprachverarbeitende Künstliche Intelligenz oder Computerlinguistik[1] gehört zu den direkten Anwendern der formallinguistischen Theoriebildung. Als ihre spezifische theoretische Fragestellung tritt die Frage effizienter algorithmischer Umsetzung hinzu.

Umgekehrt haben linguistische Fragestellungen formalwissenschaftliche Forschung angestoßen; zu denken ist an den Bereich der Grammatikformalismen, der intensionalen und der dynamischen Logik. Kann die Formalisierung in der Linguistik darüber hinausgehend auch grundsätzliche Rückwirkungen auf den Begriff der Formalisierung in den Formalwissenschaften haben? Die (computer-)linguistische Entwicklung der letzten beiden Jahrzehnte gibt Anlass zu einem tentativen Ja.

Mathematisches Wissen gilt als Musterbeispiel *exakten* und *expliziten* Wissens. Verwendete Begriffe müssen eindeutig definiert sein, Argumente müssen zwingend und ohne Ausnahme sein. Hierzu hat die Mathematik eine leistungsfähige Formelsprache ausgebildet, die oft als Inbegriff mathematischer Genauigkeit angesehen wird. In dieser Sprache können nicht nur arithmetische Beziehungen zwischen Objekten beschrieben werden, sondern auch Relationen und logische Abhängigkeiten zwischen Eigenschaften.

G. Frege \cite{BerechtigungBegriffsschrift} vergleicht seine Formalsprache, die Begriffsschrift, mit „Werkzeuge[n] für besondere Zwecke", die nicht die Nachteile des Universalwerkzeugs natürliche Sprache aufweise: „Weichheit und Veränderlichkeit". Gemeint sind hier wohl Eigenschaften natürlicher Sprache wie (1) Vagheit, (2) Mehrdeutigkeit, (3) Elliptizität aber auch (4) die Vielfalt konkurrierender semantisch äquivalenter Ausdrucksmöglichkeiten und (5) das Überspringen argumentativer Schritte bzw. die Verwendung unausgesprochener Voraussetzungen im natürlichsprachlichen Diskurs. Semantische Eigenschaften natürlicher Sprachen sind (1-4). (5) ist dem System natürliche Sprache nicht inhärent, sondern nur eine Beobachtung an üblichem Sprachgebrauch. Da die logischen Ausdrücke in formalwissenschaftlichen Kontexten in der Regel normiert verwendet werden (*alle* wird immer ohne Existenzpräsupposition gelesen) und das nichtlogische Vokabular direkt oder indirekt – auf

[1] Die beiden Begriffe *sprachverarbeitende KI* und *Computerlinguistik* werden oft nicht koextensional benutzt. Man mag die beiden Begriffe als durch unterschiedliche Zugangsweisen zu einem gemeinsamen Forschungsinteresse charakterisiert sehen.

diese Unterscheidung wird noch einzugehen sein – explizit oder implizit definiert wird, ist Vagheit kaum ein Problem. Formalisierung zielt aber auf die Probleme (2-5): Formalismen sind (2’) eindeutig, (3’) explizit und (4’) verfügen über eine überschaubare Menge an Ausdrucksmitteln, was die Definition von Ableitungskalkülen vereinfacht, (5’) mit deren Hilfe bestimmbar ist, was ein lückenloses Argument ist.

Die Formalisierung mathematischer Aussagen ist besonders wichtig in Bereichen, die sich der Vorstellung entziehen oder die auf Grund ihrer Allgemeinheit auf verschiedene Vorstellungsbereiche zutreffen. Erst durch geeignete Formalisierung ließen sich komplizierte Situationen, etwa in den Grundlagen der Analysis, klären. Auch die Systematik vieler Bereiche tritt erst nach umfangreicher Formalisierung hervor: die Regeln der Differentialrechnung erscheinen dann als systematische syntaktische Umformungen von Funktionstermen. Syntaktische Operationen treten als neue Rechenarten neben Addition und Multiplikation. A. N. Whitehead \cite[S.~61]{IntrMath} charakterisiert diese Entwicklung folgendermaßen:

> [...] by the aid of symbolism, we can make transitions in reasoning almost mechanically by the eye, which otherwise would call into play the higher faculties of the brain. [...] Civilisation advances by extending the number of important operations which can be performed without thinking about them.

Im Laufe der Entwicklung wurde deutlich, dass es keinen grundlegenden Unterschied zwischen Rechnen und logischem Argumentieren gibt. Schon bei Raimundus Lullus findet sich die Idee einer *ars magna* als universelle, kombinatorische Logik. G. W. Leibniz unterschied in seinen formal-logischen Untersuchungen *ars inveniendi* und *ars iudicandi* und entwickelte die Utopie einer rechnenden Vernunft, mit der sich alle Probleme auf eindeutige, mechanische Art lösen lassen könnten. Das Leibniz'sche Programm des *calculemus* inspirierte manche Entwicklungen in der modernen Informatik und konnte in bescheidenen Teilen durch Computerprogramme realisiert werden.

Am erfolgreichsten war die *ars iudicandi* im Bereich der Mathematik. G. Boole, G. Frege und D. Hilbert schufen mit Aussagenlogik und Quantorenlogik eine formale mathematische Sprache, in der sich mit der mengentheoretischen Begründung der Mathematik alle mathematischen Eigenschaften ausdrücken lassen. Im Gödelschen Vollständigkeitssatz \cite{Gödel} wurde gezeigt, dass die Beweiskalküle von Frege und Hilbert grundsätzlich für alle Zwecke ausreichen: wenn eine Aussage mathematisch bewiesen werden kann, so kann sie innerhalb eines Beweiskalküls aus den zugrunde gelegten Axiomen abgeleitet werden.

Hiermit ist prinzipiell der Weg zur mathematischen *ars iudicandi* gewiesen: ein formales Argument ist korrekt, wenn es eine Folge von formalen Aussagen ist, von denen jede mittels der Regeln des Beweiskalküls aus Axiomen oder früheren Aussagen gebildet werden kann. Die Überprüfung eines Textes auf diese Kriterien kann mit Hilfe einfacher Algorithmen vorgenommen werden.

Die tatsächliche mathematische Fachsprache weicht aber von dem Ideal formaler Texte und formal-logischer Argumentationen erheblich ab. Mathematisches Denken, Argumentieren und Kommunizieren vollzieht sich teilweise in vortheoretischen Begriffen und Analogien. Oft sind Formulierungen absichtlich unvollständig oder sogar formal falsch, um die zugrundeliegenden Ideen besser zu erfassen. Dieses Phänomen – hinter dem weiterhin das Ideal vollkommener Exaktheit steht – findet sich in Abstufungen in allen mathematischen Texten.

Die *Principia Mathematica* von B. Russell und A.N. Whitehead \cite{Principia} stellten den ersten umfassenden Versuch dar, die Mathematik formal-logisch aufzubauen. Russel und Whitehead bestätigten durch ihre Arbeit einerseits die grundsätzliche Möglichkeit einer solchen Unternehmung, andererseits führten aber gerade die *Principia* zu der Erkenntnis, dass eine Durchformalisierung zu umfangreich und mathematisch unfruchtbar sein würde. So schreibt Russell \cite{RussAuto} selbst über das Unterfangen:

> ... never quite recovered from the strain of writing it.

In der von N. Bourbaki \cite{Bourbaki} angestrebten Gesamtdarstellung der Mathematik *Éléments De Mathématique* stehen inhaltliche Vollständigkeit und Exaktheit im Vordergrund, aber einer formalistischen Darstellung wird von vornherein ein Absage erteilt:

> If formalized mathematics were as simple as the game of chess, then once our chosen formalized language had been described there would remain only the task of writing out our proofs in this language, [...] But the matter is far from being as simple as that, and no great experience is necessary to perceive that such a project is absolutely unrealizable: the tiniest proof at the beginnings of the Theory of Sets would already require several hundreds of signs for its complete formalization. [...] formalized mathematics cannot in practice be written down in full, [...] We shall therefore very quickly abandon formalized mathematics, [...]

Diese Sichtweise wurde bis in die 60er und 70er Jahre von den meisten Logikern geteilt. Die Möglichkeit der Formalisierung wurde nur als letztendliches Korrektheitskriterium angesehen. So schreibt Mac Lane \cite[S.~377]{MacLane}:

> As to precision, we have now stated an absolute standard of rigor: A Mathematical proof is rigorous when it is (or could be) written out in the first-order predicate language $L(\in)$ as a sequence of inferences from the axioms ZFC, each inference made according to one of the stated rules. [...] When a proof is in doubt, its repair is usually just a partial approximation to the fully formal version.

Die praktische Seite der formalisierten Mathematik änderte sich allerdings vollkommen mit dem Einsatz von elektronischer Rechenmaschinen. Die undankbare Aufgabe

der Überprüfung von formalen Regeln in Aussagen und Ableitungen lässt sich mit den Textverarbeitungsmöglichkeiten von Computern problemlos ausführen. Im Vordergrund der Entwicklung ab etwa 1960 standen *Beweiser*, deren Aufgabe in der selbständigen Ableitung von vorgegebenen Aussagen aus Axiomen besteht. Es zeigte sich jedoch bald, dass das automatische Beweisen aus Komplexitätsgründen nur in sehr
begrenzten Bereichen möglich ist.

Als Konsequenz wurden Systeme entwickelt, die den Mathematiker bei der Abfassung und Überprüfung formaler Beweise unterstützen, aber nicht versuchen, anspruchsvolle Beweisideen zu generieren. Pionier dieses Ansatzes war N. G. de Bruijn (vergl. \cite {automath}). Das System Automath überprüft mathematische Texte, die in einer lisp-artigen formalen Sprache abgefasst sind, auf logische Richtigkeit. Die allgemeine Anwendbarkeit des Systems wurde von L. S. van Bentham Jutting 1977 \cite{Jutting} durch Übersetzung der *Grundlagen der Analysis* von E.~Landau \cite{Landau} demonstriert. Allerdings arbeitet das System mit einer für den Außenstehenden unverständlichen logischen Sprache.

An diesem Beispiel wird die Wichtigkeit einer „lesbaren" Eingabesprache deutlich. Die formale Richtigkeit soll nicht auf Kosten der mathematischen Klarheit gehen. Neuere Systeme kommen dieser Forderung entgegen. Die polnische Entwicklung Mizar durch A. Trybulec u. a. \cite{mizar} zeichnet sich durch eine formale Sprache aus, die wichtige Ähnlichkeiten mit der gewöhnlichen mathematischen Sprache besitzt. Die Mizar-Sprache verhält sich zur Automath-Sprache wie eine höhere Programmiersprache zu einer Maschinen-nahen Sprache. Die Mizar-Sprache ist auf Grund einer ansatzweise natürlichsprachlichen Grammatik und geschickter Konventionen „lesbar".

Ein Schlüsselbegriff des Mizar-Systems ist der des „offensichtlichen Schlusses". Das System enthält einen automatischen Beweiser, der eine Reihe von „offensichtlichen" Umformungen und Einsetzungen akzeptiert, wie sie in einem gewöhnlichen Beweis ebenfalls unbegründet bleiben. Auf diese Weise wird implizites Wissen in Beweisen erkannt und verarbeitet. Im Mizar-System sind bisher umfangreiche Formalisierungen mathematischer Theorien durchgeführt worden. Gegenwärtig umfasst die elektronische Bibliothek des Mizar-Projekts etwa 2000 Definition und 20000 Theoreme aus allen grundlegenden mathematischen Bereichen [http://www.mizar.org/library/]. Es wird zur Zeit an der Übersetzung der anspruchsvollen Monographie \cite{lattices} gearbeitet.

Inzwischen existieren neben Automath und Mizar zahlreiche Beweis-Prüfsysteme, die sich nach verwendeter Logik, Anwendungsbereich, Benutzer-Komfort, eingebautem automatischen Beweiser und Leistungsfähigkeit erheblich unterscheiden.

Doch zeigt eine nähere Betrachtung der unter (2-5) oben genannten Probleme bei der Beurteilung von Argumenten in natürlicher Sprache, dass die Entwicklung bei Mizar und Verwandten keineswegs stehen bleiben muss. Formale überprüfbare Beweissprachen können zumindest der Sprache, die in mathematischen und formalwissenschaftlichen Lehrbüchern verwendet wird, noch weit ähnlicher sein als derzeitige „lesbare" Beweissprachen es sind. Der Mizar-Vision einer formal geprüften Forschungs- und Lehrbibliothek mathematischen Wissens käme man so näher.

Zu den einzelnen Punkten:
Ad (2): Lexikalische Ambiguitäten, die nicht durch die Feststellung einer bestimmten Diskursdomäne oder durch Betrachtungen des engeren Satzkontextes, aufzulösen wären, sind in Texten über formale Gegenstände eher selten. Größere Schwierigkeiten können ambige Anaphern und Skopusambiguitäten bereiten. Mehr als in anderen Textsorten wird jedoch gerade in formalwissenschaftlichen Texten von Techniken Gebrauch gemacht, die auf Disambiguierung zielen (Benennung von Diskursreferenten, Verwendung von Konstruktionen wie *für alle x gilt, dass*, bei denen der Skopus explizit gemacht wird). Ansonsten scheinen in den weitaus meisten Fällen Standardannahmen über den Skopus gerechtfertigt. Ein Beweiseditor, der vor unauflösbaren Ambiguitäten warnt und den Autor über heuristisch aufgelöste informiert könnte für größtmögliche Transparenz und Klarheit sorgen.
Ad (3): Wenn wir zum Bereich der Auslassungen auch das *Bridging* rechnen (Beispiel: *Gegeben seien zwei nicht parallele Geraden. Der Schnittpunkt ...*), so hängt die erfolgreiche Behandlung dieses Phänomens vom lexikalischen und Hintergrundwissen in der jeweiligen formalwissenschaftlichen Teildomäne ab. Die Fachsprachen in den Teildomänen sind jedoch überschaubar, so dass viele dieser Phänomene erfassbar scheinen. Sehr viel mehr außersprachliche Intelligenz erfordern Auslassungen, deren intendierte Auflösung Analogien involviert (Beispiel: *im Fall 2 ebenso*).
Ad (4): Die Ausdrucksvielfalt natürlicher Sprachen gegenüber Formalsprachen wird von den Beweisautoren oft genutzt, um eine globale Kohärenz herzustellen. Irrealis-Konstruktionen werden beispielsweise zur Anzeige eines Widerspruchsbeweises verwendet. Auf einzelne Fälle in Fallunterscheidungen kann mit unterschiedlichen Mitteln Bezug genommen werden, darunter auch räumlich oder temporal deiktischen. Die Ausdrucksvielfalt sollte in einer Beweissprache möglichst wenig beschränkt werden, da sie für den geübten Leser wichtige Hilfe beim globalen Diskursverständnis gibt.
Ad (5): Mizar gestattet die implizite Verwendung trivialer Sätze und das Überspringen trivialer Beweisschritte (z.B. die Umkehr der Argumente bei symmetrischen Relationen), die wir auch in natürlichsprachlichen Argumenten für gewöhnlich übergehen. Die Annäherung an übliche Beweistexte hängt davon ab, inwiefern sich diese Beweislücken mit automatischen Beweisern schließen lassen. Was als trivial gilt, dürfte in hohem Maße teildomänenspezifisch sein, eine Typologie der Beweislücken ist vonnöten.

Die voranstehende Diskussion legt nahe, dass es in der Reichweite heutiger computerlinguistischer Techniken liegt, eine Beweissprache, die der üblichen Beweissprache in mathematischen und formalwissenschaftlichen Publikationen weitaus ähnlicher ist, als es heute verbreitete Beweissprachen sind, formal zu interpretieren, um in ihr formulierte Beweise automatisch überprüfbar zu machen. Solche Beweissprachen haben möglicherweise das Potenzial, auch in die Publikationspraxis Einzug zu halten. Damit sind – in Anspielung auf die montaguesche Vision – zwar nicht natürliche Sprachen zu Formalsprachen geworden, aber ein Treffpunkt scheint in Sicht.