Automath : ein Projekt zur Kontrolle vom Mathematik

V8

INNSBRUCKER MATHEMATIKERTAG 1974

AUTOMATH - ein Projekt zur Kontrolle von Mathematik [1)]

von N. G. de Bruijn

Dieser Vortrag soll das Projekt AUTOMATH beschreiben, ohne über die Natur der Sprache AUTOMATH selbst zu sehr ins Konkrete zu gehen. Wir verweisen auf die am Ende angeführten Arbeiten [1] und [2], wo man Einzelheiten über die Definition der Sprache finden kann; hier wollen wir uns hauptsächlich auf die Motivation konzentrieren.
Ein klärendes Wort gleich zu Beginn: AUTOMATH ist eine mathematische Sprache, nicht eine Programmiersprache. Freilich haben die beiden Arten von Sprachen vieles gemeinsam, und gewiß kann jede von Ideen aus dem Gebiet der anderen profitieren.
Das Projekt AUTOMATH wurde 1966 geplant. Die Idee war die, ein System zu entwickeln, das gestattet, ganze mathematische Theorien in so präziser Weise niederzuschreiben, daß die Verifikation der Richtigkeit automatisch durchgeführt werden kann, und dennoch Schritt für Schritt den Kontakt mit der herkömmlichen mathematischen Darstellungsweise zu wahren. Ähnliches mag vielleicht Leibniz vorgeschwebt sein, nahm aber damals nicht Gestalt an , da man weder Interesse an forma-

---

[1)] Der vorliegende Text ist, abgesehen von ganz geringfügigen Änderungen, eine Übersetzung (besorgt von Bernhard Roider und durchgesehen vom Autor) des englischsprachigen Manuskripts zu einem vom Vortragenden im Rahmen des Symposiums APLASM (Orsay, Dezember 1973) gehaltenen Referat. Der englische Originaltext wird im Symposiumsbericht gedruckt erscheinen.

len Sprachen noch Erfahrung mit ihnen hatte.
Die Idee ist die, eine solche Sprache zu schaffen, daß alles, was wir in ihr schreiben, als korrekte Mathematik interpretierbar ist, solange wir syntaktisch korrekt schreiben (einschließlich korrekter Verweise auf schon Gesagtes). Dies kann bis zum Schreiben einer riesigen mathematischen Enzyklopädie gehen, zu der jeder, Mensch oder Maschine, beitragen kann, was er will, wobei jeder Beitrag, der nicht aus syntaktischen Gründen zurückgewiesen wurde, von anderen ohne Bedenken verwendet werden kann. Die Idee einer Art formalisierter Enzyklopädie wurde schon 1900 von Peano gefaßt und teilweise verwirklicht, aber das war noch weit von dem entfernt, was man automatisch lesbar nennen könnte.
Das Nachprüfen syntaktischer Richtigkeit kann einem Computer überlassen werden. Da nur nachgeprüft wird, ob der Text nach den Regeln geschrieben worden ist, müssen wir zugeben, daß das Nachprüfen eine ebenso menschliche Tätigkeit darstellt wie das Schreiben. Dennoch denken wir an einen Computer, um Normen festzusetzen: was ein Computer nicht kann, kann man nicht automatisch nennen. Außerdem haben Computer einige praktische Vorteile gegenüber Menschen. Sie nehmen jedes Detail ernst und werden nie aus Langeweile nachlässig. Der schreibende Mensch neigt dazu, gelegentlich in den Details abzuwechseln und zu glauben, dies habe anderswo keine Auswirkungen; der Computer kennt hier keinen Pardon.
Die Geschwindigkeit des Computers ist kaum ein Problem, da wir von ihm nicht viel mehr Arbeitsleistung als von einem schreibenden Menschen erwarten. Die tatsächlichen Probleme liegen bei den heutigen Computersystemen in der Speicherplatzorganisation.Der Mathematiker hat ein unterteiltes Gedächtnis-rasche und langsame Teile des Gehirns, das Papier, auf dem er schreibt, seine eigenen jüngsten Aufzeichnungen, die Bücher auf seinem Schreibtisch, die Institutsbibliothek, und schließlich andere Bibliotheken,

auf die er zurückgreifen muß, wenn seine Institutsbibliothek nicht ausreicht. Ähnlich enthält das Gedächtnis eines Computersystems Flip-Flops den Kernspeicher, Trommel, Platten, Band usw. Mensch oder Maschine, in beiden Fällen steht der Benützer vor der Entscheidung, was wo zu speichern ist. Beim Computer ist es durchaus möglich, daß technische Verbesserungen der Speicher mit raschem Zugriff diese Speicherungsschwierigkeiten aus der Welt schaffen werden.

Von vornherein bestand eines der Ziele des Projekts AUTOMATH darin, etwas Universelles zu schaffen. Dies ist ein Nachteil gegenüber Systemen, die nur kleine Teile der Mathematik anzupacken versuchen, wie etwa Aussagenlogik, Prädikatenlogik usw. Die Universalitätsforderung hatte zufolge, daß bezüglich des automatischen Beweisens von Sätzen keine Ansprüche gestellt werden konnten. Diese Thematik ist so schwierig, daß ein Erfolg nur in Situationen möglich ist, wo Probleme und Methoden einem ganz beschränkten Gebiet angehören und wo Sprache und syntaktische Analyse genau auf die zu erwartende Situation zugeschnitten sind.

AUTOMATH ist eine Sprache, in der wir Bücher schreiben können, die aus Folgen von Zeilen bestehen. Die syntaktische Richtigkeit einer Zeile hängt von den vorangehenden Zeilen ab. Vorderhand interessieren wir uns hauptsächlich für Bücher, die der üblichen mathematischen Darstellung fast Zeile für Zeile folgen und keine Gedanken ausdrücken, die dem menschlichen Mathematiker nicht kämen.

Wir müssen uns klar sein, daß keine Sprache die gesamte mathematische Aktivität umfassen kann. Sprache und Bezeichnungsweise mögen das Gestaltwerden von Ideen beeinflussen, doch zu verlangen, daß Ideen nur in einer strengen Sprache Gestalt annehmen dürfen, hieße die Mathematik töten. Insbesondere können geometrische und physikalische Intuition in einem starrgefügten formalen

Gefüge kaum Platz finden. Auf der rein sprachlichen Seite erscheint es schwierig, erklärende, natürliche Sprache durch etwas Formaleres zu ersetzen. Psychologisch besteht mathematisches Verstehen meist in mehr (doch manchmal in weniger) als dem Nachprüfen der Richtigkeit: Es kann in seelischer Beruhigtheit bestehen, die eine mathematische Situation mit bereits vertrauten Situationen im Einklang sieht. Ein Teil dieser Art des Denkens vollzieht sich, wie man glaubt, unterbewußt. Selbst wenn wir nicht auf vollständiger Formalisierung bestehen sondern nur Schritt für Schritt verläßlich Mathematik verlangen, töten wir Teile der Mathematik, zumindest in ihren Frühstadien. Wichtige Teile der Mathematik wurden auf der Basis grundlegender Irrtümer oder zumindest sehr ernster Lücken erforscht. Ohne Kenntnis von den Kostbarkeiten am jenseitigen Ufer hätte man nie die Kraft (oder die Methode) gefunden, den Irrtum zu überwinden oder die Lücke zu schließen. In manchen Fällen war es für die Mathematik ein großes Glück, daß man nicht über das intellektuelle Rüstzeug verfügte, die Existenz eines Irrtums oder einer Lücke überhaupt zu bemerken, bevor man mit der jenseits liegenden Materie reiche Erfahrungen hatte.
Wir wollen die Herstellung vollkommen formalisierter Mathematik als eine Art Fließband zu beschreiben versuchen. Wenn wir ein Buch in **AUTOMATH** als Endziel ansehen,so ergeben sich folgende Phasen:

(I) mathematische Ideen
(II) formale Definitionen und Beweise
(III) sehr präzise und ausführliche Darstellung derselben
(IV) ein Buch in einer Zwischensprache
(V) ein Buch in AUTOMATH.

Wir haben (IV) eingefügt, weil AUTOMATH - eben wegen seiner Universalität - nicht leicht zu schreiben ist.

Der überwiegende Teil des mathematischen Stoffes betrifft nur einen jeweils kleinen Teil der Mathematik mit bewährten Traditionen, die Dinge kurz anzuschreiben und auszudrücken. Deshalb haben wir Bücher der Type (IV), sozusagen in einer problemorientierten Sprache, eingeschlossen.
Welche Art von Personal brauchen wir am Fließband? Um (I) herzustellen, brauchen wir den großen Mathematiker. (Hier meinen wir keine besondere Klasse von Mathematikern - jeder Mathematiker kann dann und wann groß sein). Um von (I) nach (II) zu kommen, brauchen wir den guten Mathematiker, der das Gebiet und seine Technik beherrscht. Die Phasen (I) und (II) haben selbstverständlich mit AUTOMATH nichts zu tun; sie sind das Gebiet der herkömmlichen mathematischen Praxis.
Um das halbfertige Produkt von (II) nach (III) überzuführen, brauchen wir einen ausgebildeten Mathematiker. Auch er muß den Gegenstand immerhin kennen, wenigstens sollte er die stenographischen Traditionen des Gegenstands beherrschen.
Die Überführung von (III) nach (IV), von (IV) nach (V), sowie die Endkontrolle von (V), kann billigen Arbeitskräften überlassen werden. Viel davon, jedenfalls die Kontrolle von (V),kann ganz billiger Arbeitskraft in Gestalt eines Coputers überlassen werden.
Es gibt viele Dinge, die eine universelle Sprache wie AUTOMATH erreichen könnte. Einige davon bilden, für sich allein betrachtet, keine ausreichende Motivation für das Projekt AUTOMATH, aber ihre Gesamtheit erscheint bedeutungsvoll genug, um eine Anstrengung zu unternehmen. Wir wollen die Ziele in zwei Gruppen einteilen: Nachprüfen und Verstehen.
Das erste, woran man beim Wort "nachprüfen" denkt, ist wohl das Nachprüfen langwieriger Beweise, wobei die Kette nur so stark ist wie ihr schwächstes Glied, und wobei oft genug die Verläßlichkeit des Beweises nicht

durch Intuition oder praktische Erfahrung gestützt wird. Insbesondere wird man diese Situation bei komplizierten kombinatorischen Problemen zu erwarten haben, wo zahlreiche Fälle und Unter-Fälle geprüft werden müssen. Unter diesem Stichwort finden wir auch Probleme aus der Semantik der Coputerprogramme. Die Anzahl der nötigen elementaren Schritte und der bürokratische Arbeitsaufwand können so groß sein, daß menschliche Methoden sehr unzuverlässig werden. Gerade hier müssen wir auch an die Probleme des Teamworks und der Zusammenarbeit zwischen Mensch und Maschine denken. Beide erfordern ein festgefügtes Kommunikationssystem. Es erscheint der Mühe wert, auf diesem Gebiet zu arbeiten, da ungeheure Summen für Computersoftware ausgegeben werden und es durchaus von Interesse ist, zu wissen, was verläßlich ist und was nicht.
Werfen wir jetzt einen Blick auf die Ziele, die unter das Schlagwort "Verstehen" fallen. Zunächst läßt sich sagen, daß die bloße Tatsache, eine feste, wohldefinierte Sprache für die Mathematik zu besitzen, für sich allein schon ein Vorteil ist. Sie erlaubt uns, die mathematische Diskussion einzuteilen in (I) das Formulieren in der Sprache, (II) Diskussion über die Art und Weise des Formulierens, (III) Verknüpfung von in der Sprache Formuliertem mit Dingen einer anderen Welt, wie etwa üblicher Mathematik, physikalischer Wirklichkeit usw. Wir könnten (II) als "Metasprache" und (III) als "Interpretation" bezeichnen.
Die meisten Mathematiker haben von den Grundlagen ihrer eigenen Mathematik keine klare Vorstellung. Dies mag zum Teil ein Fehler der Logiker sein, die vor lauter interessanten technischen Fragen in ihrem eigenen Gebiet ihre ursprüngliche Mission, ein Fundament für andere zu errichten, vernachlässigt haben. Viele Mathematiker haben eine vage Vorstellung, daß die Prädikatenlogik zusammen mit der Mengenlehre eine vollständige Basis für

ihre eigene Aktivität bildet, aber wenn sie sich diese Gebiete ansehen, stellen sie zu ihrer Überraschung fest, daß Logik und Mengenlehre ebenfalls in mathematischer Aktivität bestehen! Statt einer Grundlage für die mathematische Schablone Axiome-Definitionen-Schlußregeln-Beweise-Lehrsätze finden sie allenthalben dieselbe Schablone wieder. Was fehlt, ist eine gute Sprache. In der Tat, in AUTOMATH wird all das ganz klar. Die Sprache enthält kaum etwas, das Logik genannt werden könnte, und sobald wir die Sprache haben und die Dinge (syntaktisch) richtig sagen, gehört die Frage, was Axiome, Schlußregeln, Definitionen, Annahmen, Lehrsätze usw. sind, nur mehr der Metasprache und der Interpretation an. Es ist ohne jeden Einfluß auf die Resultate eines Buches in AUTOMATH , ob wir etwas eine Definition, einen Lehrsatz oder sonstwie nennen; es ist einfach, so wie es steht, richtig.

Ein anderes Ziel in Richtung "Verstehen" ist die Komplexitätsanalyse. Manche Dinge sind schwieriger als andere, und eine vollständige formale Darstellung vermag dies zu durchleuchten. Es ist möglich, Teile der Mathematik nach ihrer "Tiefe" zu klassifizieren. Die Mathematik des XIX. Jahrhunderts war gewiß tiefer als die des XVIII. Etwas stilisiert kann man sagen, daß man im XVIII. Jahrhundert über Funktionen sprechen konnte, die man explizit konstruiert hatte, aber nicht sagen konnte, "f sei eine Funktion", da das Wort"Funktion" metasprachlich war. Ebenso stilisiert könnte man sagen, daß sich die Mathematik des XVIII. Jahrhunderts in PAL, derjenigen Teilsprache von AUTOMATH, die wir durch Weglassen des λ-Kalküls erhalten, ausdrücken läßt.

In diesen Zusammenhang paßt auch die Bemerkung, daß AUTOMATH die historische Reihenfolge umkehrt. Schon in PAL werden etwa "Beweise" genau gleich behandelt wie "Zahlen", während selbst in der zweiten Hälfte des zwanzigsten Jahrhunderts die Mehrzahl der Mathematiker

"Beweis"als einen metasprachlichen Begriff und eine "Zahl" als ein "Objekt" empfinden. Die Vorstellungen, was ein Objekt ist und was nicht, sind meist vage. Der Unterschied zwischen Objekten und Nicht-Objekten läuft anscheinend parallel mit der Unterscheidung zwischen Sprache und Metasprache; man empfindet ein Objekt als etwas, das man mit einem Symbol bezeichnen kann. Viele Leute glauben, daß man besser über Mengen als über Prädikate spricht. Anstatt zu sagen, daß x das Prädikat P erfüllt, formen sie lieber die Menge aller Dinge, die dieses Prädikat erfüllen, und sagen dann, daß x zu dieser Menge gehört. Meist liegt der Grund dafür in der Angst vor Prädikaten, die nicht für Objekte gehalten werden.

Um auf das "Verstehen" zurückzukommen: Im Englischen sagt man häufig, Mathematik werde durch Terrormethoden (intimitation) unterrichtet und durch sklavische Nachahmung (imitation) erlernt - freier übersetzt: Mathematikunterricht ist Einschüchtern und Eintrichtern. Der einzige Weg, herauszufinden, wieviel Wahrheit darin enthalten ist, besteht darin, alles in einer sehr strengen Sprache zu kodifizieren.

Unter dem Stichwort "Verstehen" kann man schließlich den Einfluß einreihen, den jede neue Bezeichnungsweise, sofern sie ein gewisses Maß an Ausdruckskraft besitzt, auf die Entwicklung der Mathematik ausübt, mag er nun beabsichtigt sein oder auch nicht.

Abgesehen vom "Nachprüfen" und "Verstehen" hat die Tatsache, daß Maschinen die Mathematik, die wir produzieren, verarbeiten können, weitere Vorteile. Beispielsweise können wir uns vorstellen, daß wir einer Maschine ein Buch über analytische Zahlentheorie geben mit dem Auftrag "Mich interessiert nur der Primzahlsatz. Drucke alles, was für diesen Satz benötigt wird, und lasse alles andere weg." (Manche Leute behaupten, Edmund Landau sei eine solche Maschine gewesen; er schrieb seine Bücher so.) Oder wir können verlangen: "Drucke Satz 325 und alle Definitionen, die benötigt werden, um das, was er aussagt, lesen zu können, von Adam und Eva an." In diesem Falle werden auch die Beweise weggelassen.

Um anzudeuten, wie mathematische Überlegungen in einem Buch in AUTOMATH ausgedrückt werden, müssen wir auf die Sprache kurz eingehen. Zunächst wollen wir festhalten, daß die Bücher in ineinandergeschachtelte "Blöcke" von Zeilen eingeteilt sind. Die erste Zeile eines Blocks hat eine besondere Form. Ihre Interpretation ist die, daß wir entweder eine Variable, die innerhalb des Blocks benützt werden kann, oder eine im gesamten Block gültige Annahme einführen.

Alle Zeilen haben folgende Form:

"Im Kontext A ist der Name B definiert als C und vom Typ D." Hier ist B ein neuer, in früheren Zeilen nicht vorkommender, Bezeichner. C und D sind Ausdrücke, die aus alten Bezeichnern mithilfe einiger Konnektivs wie Klammern, Parenthesen, Beistriche usw. aufgebaut sind. Manche Zeilen (die Blockeröffner) haben nur einen Strich (—) anstelle von C (Interpretation: eine Variable wird eingeführt, indem man sie benennt und ihren Typ angibt). In jeder Zeile ist A eine Kette von bereits eingeführten blockeröffnenden Bezeichnern. Die A-Teile der Zeilen dienen dazu, die Blockstruktur des Buches anzugeben, indem sie für jede Zeile angeben, zu welchen Blöcken sie gehört.

Manchmal ist C nicht ein Ausdruck sondern das besondere Symbol "PN". Die Zeilen mit C-Teil PN dienen dazu, primitive Begriffe (englisch: primitive notions) einzuführen, welche nicht definiert werden. Sie erhalten nur Namen und Typ, keine Definition, doch sie können von da an benützt werden. Eine PN-Zeile ist kein Blockeröffner; sie steht einfach irgendwo innerhalb eines Blockes.

Wir müssen die Möglichkeit erwähnen, daß der D-Teil

einer Zeile kein Ausdruck ist, sondern nur das Symbol "type". Solche Zeilen führen einen neuen Typ ein, entweder durch Definition oder als primitiv oder als Variable.
Dies umreißt ganz grob den Aufbau der Sprache PAL, die wir früher erwähnt haben. Die Sprachen der AUTOMATH-Familie ergeben sich aus PAL, wenn wir eine bestimmte Art von λ-Kalkül mit Typen hinzufügen. Wir gehen darauf nicht ein.
Einige Worte über die Interpretation. Zunächst ist die Kontextanzeige (der A-Teil) etwas, was üblicherweise in der Mathematik nicht explizit angegeben wird. Teile davon können etwa aus der Unterteilung in Kapitel und Abschnitte erschlossen werden, andere Teile lassen sich durch sorgfältiges Lesen des vorangehenden Textes aufspüren. Der B-Teil hat die übliche Interpretation als Name für ein neues Objekt, das wir bilden oder postulieren. Die Interpretation der C- und D- Teile wird adäquat beschrieben, indem man sagt, daß B durch C definiert wird und vom Typ D ist. Wir wollen das Symbol Ɛ für diese Typzuweisung einführen: CƐD. In der natürlichen Sprache sagen wir etwa: "3 ist eine Zahl", doch da das Wörtchen "ist" für vielerlei verwendet wird, ziehen wir "3 Ɛ Zahl" vor.
Einige der Typen, die wir verwenden werden, haben mengenartige Interpretationen. Anstelle von "3 Ɛ Zahl" könnte man an $3 \in S$ denken, wobei S die Menge aller Zahlen ist, aber wir müssen uns hüten, Ɛ und $\in$ zu verwechseln. In AUTOMATH ist der Typ eines Dinges C (d.h. das D mit CƐD) eindeutig bestimmt und kann mittels eines Algorithmus berechnet werden. Bei $3 \in S$ ist dies nicht so, da S jede 3 enthaltende Menge sein kann.
Neben den Typen mit mengenartiger Interpretation gibt es mehrere andere. Die wichtigsten sind die aussagenartigen Typen. Dieser Interpretation nach entspricht der D-Teil einer Aussage und der C-Teil ihrem Beweis.

Wir können gewisse Operationen an Beweisen vornehmen: wenn sie von Variablen abhängen, können wir Ausdrücke für diese Variablen in genau der gleichen Weise wie im Falle von Objekten, die von Variablen abhängen, substituieren. Dies hat die Wirkung, daß ein (durch Substitution) modifizierter Beweis als Beweis für die entsprechend modifizierte Aussage angenommen wird. Man beachte, daß der B-Teil der Zeile ein Name für den Beweis C, nicht für die Aussage D ist. Die ganze Zeile kann ein Lehrsatz genannt werden; spätere Anwendungen dieses Lehrsatzes erfolgen durch Verweise auf B. Man beachte, daß die Mehrzahl der Lehrsätze nur Sprossen einer Leiter zu einer wichtigen Lehrsatzzeile sein werden, der ein Mathematiker diesen Namen zubilligen würde; er würde die anderen Zeilen nicht einmal als Hilfssätze bezeichnen.

Es gibt auch Blockeröffner mit aussagenartiger Interpretation. Diese sagen gleichsam: "x sei ein Beweis für die Aussage D." Das heißt, diese Zeilen führen im gesamten Block gültige Annahmen ein. Und es können Zeilen auftreten, wo der C-Teil PN ist. Diese dienen dazu, die Wahrheit der Aussage D als Axiom einzuführen. Wir haben also für drei Typen von aussagenartigen Zeilen vorgesorgt: Lehrsätze, Annahmen und Axiome.

Wir können, wenn wir wollen, neue Typen schaffen, und wir können auch Interpretationen auswählen. Wenn wir zum Beispiel eine mathematische Theorie der ebenen geometrischen Konstruktionen mit Lineal und Zirkel schaffen wollen, brauchen wir uns nicht der Mühe zu unterziehen, die Konstruktionen als Mengen zu verschlüsseln (gemäß der dogmatischen Vorstellung, daß alles eine Menge ist; in [3] findet man Kritik dazu), sondern wir können direkt einen Typ "Konstruktion" einführen.

Wir erwähnen einen weiteren Fall. Für jede Menge Ω führen wir einen Typ "program (Ω)" ein. Die Interpretation von $C \in$ program (Ω) ist die, daß C ein auf dem Zustandsraum Ω operierendes Programm ist. Mittels PN-Zeilen führen wir primitive Programme und primitive Methoden zur Konstruktion größerer Programme aus kleineren Komponenten ein. Mit anderen Worten, wir beschreiben die Syntax einer Programmiersprache in demselben Buch, in dem wir die Logik und Mathematik behandeln (es gibt keinen wesentlichen Unterschied zwischen den beiden letzteren). Wir können dann in demselben Buch Axiome über die Semantik der Primitiva der Programmiersprache entwickeln. Und wir können noch im gleichen Buch logische Lehrsätze (abgeleitete Schlußregeln), mathematische Lehrsätze, semantische Lehrsätze, spezielle Programme und semantische Ergebnisse über diese Programme herleiten. Die verschiedenen Teile können untereinander verflochten sein. Zum Beispiel können eine mathematische Behandlung des größten gemeinsamen Teilers (g.g.T.) in zahlentheoretischem Rahmen, eine Beschreibung eines Computerprogrammes zur Auffindung des g.g.T. und ein Beweis, daß die Ausführung des Computerprogramms abbricht und den Wert der zahlentheoretischen Funktion g.g.T. liefert, zusammenkommen. (Bezüglich expliziter und ausführlicher Vorschläge zur semantischen Behandlung ALGOL-artiger Sprachen vgl.[4]). Es würde überhaupt nichts schaden, wollte man Syntax und Semantik zweier verschiedener Programme in einem Buch niederschreiben und in dem selben Buch beweisen, daß das Programm P_1 in der Sprache Q_1 dieselben semantischen Auswirkungen hat wie das Programm P_2 in der Sprache Q_2. Beweise dieser Art können lang und ermüdend und dennoch bedeutungsvoll sein - und sie sind bisweilen typische Fälle, wo automatische Verifikation angebracht ist.

Wenn ein Buch dieser Art mit der Außenwelt in Beziehung gesetzt werden soll, ergibt sich Anlaß zu viel Interpretation. Solange wir über keinen zusätzlichen Formalismus zur Behandlung der Interpretation verfügen, haben wir uns zu "überzeugen", daß die Primitiva (seien sie logischer, mathematischer, syntaktischer oder semantischer Natur) ausdrücken, was sie in der Außenwelt bedeuten sollen. Und wir müssen uns "überzeugen", daß die Interpretationen der Primitiva Interpretationen weiteren Materials induzieren und daß Interpretationen der Endresultate ohne Rücksicht auf die Teile des Buchs, die zwischen den Primitiva und den Endresultaten liegen, gewinnbar sind. Und wir glauben, daß die Endinterpretationen mathematisch richtig sind.

Diese Situation ist bei der Interpretation von Computersprachen komplexer als bei herkömmlicher Mathematik, aber nicht wesentlich davon verschieden. Interpretation muß immer auf verhältnismäßig intuitiver Grundlage ruhen, solange die "Außenwelt" nicht vollständig formalisiert ist.

Wir beenden diese Ausführungen mit einer kurzen Beschreibung der Projektgruppe AUTOMATH am Institut für Mathematik der technischen Universität Eindhoven in Holland. Die Gruppe befindet sich seit 1967 in langsamem Aufbau; Anfang 1974 bestand sie aus vier vollbeschäftigten Mathematikern (Logiker und Computerwissenschaftler mitverstanden), drei teilbeschäftigten Mathematikern (darunter der Autor dieser Arbeit, der das Projekt leitet), einem Programmierer und einer teilbeschäftigten Kartenlocherin. Hier ist ein kurzer Katalog der bisherigen Aktivitäten.

(I) Sprachüberprüfer wurden hergestellt und stehen nun im Rahmen eines Zeitzuteilungssystems (time sharing) im Dialogbetrieb zur Verfügung. Der Text kann Zeile für Zeile in die Maschine eingegeben werden, die innerhalb höchstens einiger Sekunden antwortet. Wenn der Überprüfer eine Zeile zurückweist, gibt er eine vollständige Fehlerbeschreibung, die im allgemeinen die damit betraute Person zur Verbesserung des Textes befähigt (möglicherweise nach telefonischer Rückfrage bei dem Mathematiker, der den Text erstellt hat). Bis September 1973 war die Rechenmaschine eine Electrologica X8, seither eine Burroughs 6700. In beiden Fällen erfordern die zur Verfügung stehenden Multiprogramming-Systeme die Verwendung der Programmiersprache ALGOL 60.

(II) Die theoretische Arbeit über Sprachen der AUTOMATH-Familie konzentrierte sich auf Probleme, die mit [starker] Normalisierung und dem Church-Rosser-Theorem zusammenhängen. Fast alle Ziele sind erreicht. Einen detaillierten Bericht über eine der Sprachen, nämlich AUT-SL, findet man in [8]. Es gibt gewisse Überschneidungen mit Arbeiten anderer ([5],[6],[7]), die unabhängig damit begonnen haben, Logik in der Sprechweise des λ-Kalküls mit Typen zu interpretieren, und zwar ungefähr zu der Zeit, als mit dem Projekt AUTOMATH begonnen wurde.

(III) Gleichsam als Testfall wurde die Aufgabe in Angriff genommen, einen besonders minuziösen mathematischen Text in die Sprache AUTOMATH zu übersetzen. Die Wahl fiel auf die "Grundlagen der Analysis" von Landau. Die Übersetzung, die L.S. van Benthem Jutting vornimmt, ist etwa zur Hälfte fertiggestellt. Es wurde nicht versucht, den Text zum Zwecke der leichteren Übersetzung in die Sprache AUTOMATH umzustellen, sondern es wurde dem Landauschen Texte so genau als möglich gefolgt (sodaß sich alle Nachteile und keine Vorteile ergeben). Wir hoffen, daß die gewonnene Erfahrung bei der Entscheidung über die für den allgemeineren Gebrauch günstigste Zwischensprache sehr nützlich sein wird. Mehrere Möglichkeiten werden derzeit untersucht.

Das Projekt AUTOMATH hängt sehr wesentlich von der finanziellen Unterstützung durch die Niederländische Organisation zur Förderung der reinen Forschung (Z.W.O.) ab.

Literatur

[1] de Bruijn, N.G. "The mathematical language AUTOMATH, its usage, and some of its extensions" (Die mathematische Sprache AUTOMATH, ihr Gebrauch und einige ihrer Erweiterungen), Symposium on Automatic Demonstration (Versailles, Dezember 1968), Lecture Notes in Mathematics, Band 125, Seite 29-61, Springer-Verlag (1970).

[2] de Bruijn, N.G., "AUTOMATH, a language for mathematics" (AUTOMATH, eine Sprache für die Mathematik), Mitschrift (hergestellt von B. Fawcett) einer Vorlesungsreihe im Séminaire de Mathématiques Supérieures, Université de Montreal,1971.

[3] de Bruijn,N.G., "Set theory with type restrictions" (Mengenlehre mit Typen-Einschränkungen), International Colloquium on Infinite and Finite Sets, Keszthely, Ungarn, 1973.

[4] de Bruijn, N.G.,"A system for handling syntax and semantics of computer programs in terms of the mathematical language AUTOMATH" (Ein System zur Behandlung von Syntax und Semantik von Computerprogrammen, ausgedrückt in der mathematischen Sprache AUTOMATH), Bericht, Institut für Mathematik, Technische Universität, Eindhoven.

[5] Girard, J.Y., "Une extension de l'interprétation de Gödel à l'analyse et son application à l'élimination des coupures dans l'analyse et la théorie des types" (Eine Erweiterung der Gödelschen Interpretation auf die Analysis und ihre Anwendung auf die Elimination der Schnitte in Analysis und Typentheorie), Proc. 2nd Scandinavian Logic Symposium

(Hrsg.: Fenstad), North-Holland Publishing Company, Amsterdam, 1970.

[6] Howard, W.A., "The formulae-as-Types notion of Construction" (Formeln als Typen - Ein Konstruktionsbegriff), hektographiert,1969.

[7] Martin-Löf, P., "An intuitionistic theory of types" (Eine intuitionistische Typentheorie), unveröffentlicht, 1972.

[8] Nederpelt, R.P., "Strong normalization in a typed lambda calculus with lambda structured types", (Starke Normalisierung in einem λ-Kalkül mit λ-strukturierten Typen),Dissertation, Technische Universität Eindhoven, 1973.

Institut für Mathematik
Technische Universität
Eindhoven, Niederlande