Begriffs-Hierarchien

und λ-Kalkül

Eine Untersuchung über Sprache, Wirklichkeit und Mathematik

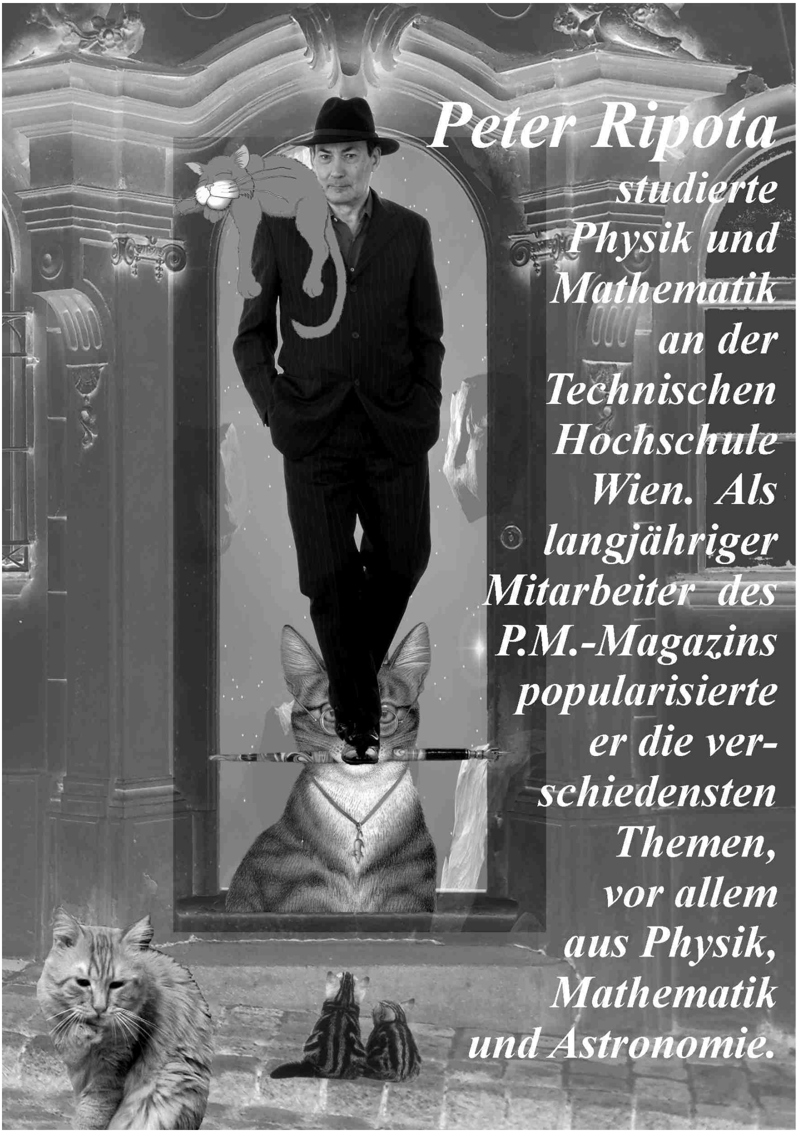
von Peter Ripota

Begriffs-Hierarchien
und
Lambda-Kalkül

λ

Eine Untersuchung über Sprache, Wirklichkeit und Mathematik

von ***Peter Ripota***


Peter Ripota
studierte Physik und Mathematik an der Technischen Hochschule Wien. Als langjähriger Mitarbeiter des P.M.-Magazins popularisierte er die ver-schiedensten Themen, vor allem aus Physik, Mathematik und Astronomie.

Bibliografische Information der Deutschen Nationalbibliothek:
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

Herstellung und Verlag: Books on Demand GmbH, Norderstedt
ISBN 978-3-7526-5366-3

e-mail: tango@peter-ripota.de
Webseite: http://www.peter-ripota.de

Begriffshierarchien und λ-Kalkül

Von Dipl-Ing. Peter Ripota

Furtwangen-Freiburg 1978, überarbeitet 2006 & 2020

Inhalt

0. Randbemerkung

Als ich meine Doktorarbeit schrieb, ahnte ich nichts von den Schwierigkeiten, daraus auch einen akademisch verwertbaren Titel zu machen. Ich lebte in Freiburg, arbeitete in Furtwangen, meine Doktorväter wirkten in Hamburg und Bremen. Zur Verteidigung und zur Bekämpfung aller Intrigen hätte ich nach Hamburg ziehen müssen (für einen Gebirgsbewohner ein grässlicher Gedanke), und außerdem wäre der Erfolg zweifelhaft gewesen. Denn meine Doktorväter kümmerten sich nicht um die Arbeit, sondern ließen sie von anderen, nicht-kompetenten Professoren beurteilen, mit dem erwarteten vernichtenden Ergebnis. So blieb es bei der Arbeit.

Aber auch nach über 40 Jahren finde ich die darin niedergelegten Gedanken noch immer bemerkenswert, auch wenn ich selbst nicht mehr alles verstehe. Besonders das simple BASIC-Programm zur Erzeugung neuer Konzepte halte ich für eine nützliche, wenngleich ungenutzte Innovation. So stelle ich meine damalige Arbeit der Öffentlichkeit vor; wer immer daran Spaß hat, möge sie sich zu Gemüte führen.

Peter Ripota

1. Zusammenfassung

Diese Arbeit befasst sich mit der **Definition von Begriffen** vom Standpunkt der Informatik, der Linguistik und der Wissenschaftstheorie. Ausgangspunkt, Leitfaden und Methodik sind die kombinatorischen Konzepte und Verfahren des λ-Kalküls (Theorie der Abstraktion und funktionalen Applikation).

Begriffsfelder werden hauptsächlich durch Anwendung dreier "Aufbau-Operatoren" geschaffen. Der **Abstraktionsoperator λ** erzeugt abstrakte Begriffe, die, als Funktionen aufgefasst, durch Anwendung auf bestimmte reale oder formale Objekte neue (konkretere) Begriffe schaffen können. Der **Synthese-Operator S** erzeugt Mengen (ungeordnete Zusammenfassungen von Elementen), die in unserer Arbeit auch "Strukturen" genannt werden; und der **Vektor-Operator V** erzeugt Felder (geordnete Zusammenfassungen von Elementen). An zahlreichen Beispielen wird die Anwendung und Brauchbarkeit dieser Operatoren bei der Darstellung von Begriffssystemen unterschiedlicher Wissensgebiete gezeigt und praktisch erprobt.

Durch die Unterscheidung zwischen **Nominaldefinitionen** (Definition von Begriffen durch andere Begriffe) und **Attributivdefinitionen** (Definition von Begriffen durch Eigenschaften) gelingt die Beantwortung der Frage, welche mathematischen Strukturen durch Begriffe geschaffen werden. Es zeigt sich, dass attributiv definierte Begriffe auf natürliche Weise zu einem *Verband* zusammengeschlossen ("lattisiert") werden können, während dies bei Nominaldefinitionen im allgemeinen zu uninterpretierbaren (sinnlosen) Resultaten führt. Auf *Begriffsverbände* können die Gesetze und Verfahren der Verbandstheorie angewandt werden; außerdem ist ein Bezug zu den so genannten extensionalen Modellen des λ-Kalküls (SCOTT) möglich. Schließlich erlaubt die Verbandstruktur eine mechanische Ableitung neuer Begriffe (unter Zugrundelegung von Einschränkungen bzw. Folgerungen), die in einem Programm, dem **Konzeptgenerator**, realisiert werden können.

Durch Verallgemeinerung des Begriffs "Begriff" gelangt man zum Konzept des **Informationselement**s eines Tripels aus Prädikat (Attribut, Eigenschaftsklasse), Objekt (Gegenstand, Begriff) und zugehörigen Wert (Ausprägung, Eigenschaft). Informationselemente sind die Grundlagen zur Beschreibung der Wirklichkeit. Aus Informationselementen werden zunächst **Zustände** aufgebaut, d.h. es wird die Menge aller für ein zu beschreibendes System relevanten Tripel Prädikat-Objekt-Wert gebildet (statische, isolierte Beschreibung der Realität). Nimmt man auch die Beziehungen zwischen den Individuen hinzu, dann erhält man eine Beschreibung durch **Situationen**.

Zustände und Situationen können sich ändern; Eine **Überführungsfunktion** beschreibt diese Änderungen; aus Zuständen entstehen dadurch **Prozesse**, aus Situationen **Aktionen**. An einigen Beispielen wird gezeigt, wie einige grundlegende Aktionen (z.B. "Erfüllung") durch hinreichend abstrakte Definition viele für die Praxis bedeutsame Konzepte implizit enthalten und durch entsprechende Reduktion explizit erzeugen.

Ein wichtiger Platz wird der **graphischen Darstellung** von Begriffsfeldern eingeräumt. Dabei wird von Überlegungen ausgegangen, die der Verfasser in früheren Arbeiten zur Aufstellung von **Logogrammen** entwickelte. Die Repräsentation **linguistischer Relationen** geht insbesondere von Grundbegriffen der Automatentheorie aus. Den linguistischen Relationen ist ein eigenes Kapitel gewidmet, wobei gezeigt wird, wie die gängigen, in semantischen Netzen häufig verwendeten Relationen in unserer Theorie auf sehr einheitliche Weise dargestellt werden (z.B. "type - token", "is", "isa", usw.).

Die Arbeit zeigt auch Maße zur numerischen Charakterisierung von Begriffen und Begriffsstrukturen sowie Relationen zwischen Begriffen, die auch didaktisch (zur Aufstellung von Lehrprogrammen) verwertet werden können. In verschiedenen wissenschaftstheoretischen Exkursen wird gezeigt, wie die Konzepte "Intension" und "Extension" auf Begriffsverbänden ebenso einfach wie natürlich exakt definiert werden können.

2. Einleitung

In JONATHAN SWIFTs satirischem Roman, "Gullivers Reisen" (1726) wird unter anderem die Akademie von Lagoda geschildert. Darin wird dem staunenden Weitreisenden eine Erfindung vorgeführt, mit deren Hilfe wissenschaftliche Theoreme mechanisch produziert werden können. Es heißt da im fünften Kapitel des dritten Teils: *Der erste Professor, den ich sah, befand sich in einem großen Zimmer und war von vierzig Schülern umgeben. Nach der gewöhnlichen Begrüßung bemerkte er, dass ich ernstlich einen Rahmen* ((engl. 'frame'. MINSKY (1974) verwendet den gleichen Ausdruck zur Kennzeichnung kleinster Konzeptionseinheiten, aus denen unsere Vorstellung von der Welt aufgebaut werden soll)) *betrachtete, welcher den größten Teil des Zimmers ausfüllte, und sagte: ich wundere mich vielleicht, dass er sich mit einem Projekt beschäftige, die spekulativen Wissenschaften durch praktische und mechanische Operationen zu verbessern. Die Welt werde aber bald die Nützlichkeit dieses Verfahrens bemerken. Er schmeichle sich mit dem Gedanken, dass eine höhere und edlere Idee noch nie aus dem Gehirn eines Menschen entsprungen sei. Ein jeder wisse, wie viel Mühe die gewöhnliche Erlernung der Künste und Wissenschaften erfordere; er sei überzeugt, durch seine Erfindung werde die ungebildetste Person bei mäßigen Kosten und bei nur einiger körperlicher Anstrengung Bücher über Philosophie, Poesie, Staatskunst, Gesetze, Mathematik und Theologie ohne die geringste Hilfe von Geist oder Studium schreiben können.*

Nach dieser bombastischen Ankündigung wird die Erfindung wie folgt beschrieben: *Der Rahmen war zwanzig Quadratfuß groß und befand sich in der Mitte des Zimmers. Die Oberfläche bestand aus einzelnen Holzstücken von der Dicke eines Würfels, von denen jedoch einzelne größer als die anderen waren. Die waren sämtlich durch dünne Drähte miteinander verknüpft. Diese Holzstücke waren an jeder Fläche mit überklebtem Papier bedeckt, und auf diesen Papieren waren alle Worte in der Landessprache ... ohne alle Ordnung aufgeschrieben. Der Professor bat mich achtzugeben, da er nun seine Maschine in Bewegung setzen wolle. Jeder Zögling nahm auf seinen Befehl einen eisernen Griff zur Hand, von denen vierzig am Bande des Rahmens befestigt waren. Durch eine plötzliche Umdrehung wurde die ganze Anordnung der Wörter verändert. Alsdann befahl er sechsunddreißig der jungen Leute, die verschiedenen Zeilen langsam zu lesen, und wann sie drei oder vier Wörter herausgefunden hatten, die einen Satz bilden konnten, diktierten sie dieselben den vier anderen, welche sie niederschrieben. Diese Arbeit wurde drei- oder viermal wiederholt. Die Maschine war aber so eingerichtet, dass die Wörter bei jeder Umdrehung einen neuen Platz einnahmen, sobald der Holzwürfel sich von oben nach unten drehte.*

Nach der exakten Beschreibung dieses Algorithmus zur Aufzählung aller möglichen Wortkombinationen (ohne Wiederholung) weist der gute Professor am Ende der Vorstellung noch einmal auf die Bedeutung dieser genialen Erfindung hin:

Aus diesem reichen Material werde er der Welt ein vollständiges System aller Wissenschaften und Künste geben; ein Verfahren, das er jedoch *verbessern und schneller beendigen könne, wenn das Publikum ein Kapital zusammenbringen wolle, um fünfhundert solcher Rahmen in Lagado zu errichten, und wenn man die Unternehmer veranlassen werde, ihre verschiedenen Sammlungen zu einer gemeinsamen zu vereinigen.*

Wie man sieht, gab es damals die gleichen Finanzierungsprobleme wie heute.

Besonders erfolgreich wurde die Methode in der modernen Poesie angewandt. Ausgehend von den Produktionen der Dadaisten (BRETON, ELUARD, ARP) entwickelte sich besonders in Frankreich eine 'poesie combinatoire', deren bekanntester Vertreter QUENEAU ist. Einen guten Übersichtsartikel mit zahlreichen Beispielen liefert GARDNER 1977.

Die Idee einer kombinatorisch konstruierten Erkenntnissammlung taucht auch in der Erzählung von der "Universalbibliothek" des deutschen Science-Fiction-Schriftstellers KURD LASSWITZ wieder auf (1902). Auch populärwissenschaftliche Schriftsteller haben sich damit beschäftigt, teils um die Mächtigkeit kombinatorischer Verfahren in der Mathematik zu zeigen, teils um die Absurdität eines solchen Gedankens bloßzulegen (wie etwa beim flötenblasenden Affen Sir ARTHUR EDDINGTONs (1958)).

In dieser Arbeit wird - unter anderem - die Idee der Akademie von Lagoda realisiert. Es werden zwar keine Sätze erzeugt (obwohl dies auch möglich wäre), sondern es werden Begriffe mit Hilfe eines Programms aus Elementen generiert. Die konzeptionelle Grundlage dazu liefert eine Theorie der Begriffshierarchien, der Begriffserzeugung, -umwandlung und -definition, welche ein kombinatorisches mathematisches System, nämlich den <u>Lambda-Kalkül</u> oder die ihre äquivalente <u>Theorie der Kombinatoren</u> verwendet. Warum gerade diese Theorie?

Stellen wir uns vor, wir hätten einen Begriff (ein Konzept) B, der eine abstrakte Wirklichkeit repräsentiert. Aus diesem Begriff sind eine Reihe anderer Begriffe B_1 bis B_n ableitbar, während B selbst (sofern er nicht logisches Element ist) seinerseits aus den Begriffen A_1 bis A_m abgeleitet werden kann. Die Möglichkeiten der Ableitbarkeit sind vielfältig.

Es gibt die *logische Deduktion*, die vor allen bei mathematischen Konzepten häufig angewandt wird. Es gibt die Möglichkeit, einen Begriff zu *konkretisieren* (zu spezialisieren), oder ihn zu *verallgemeinern* (zu abstrahieren). Beispielsweise kann der Begriff "Atom" zu einem konkreten Atom (z.B. "Helium-Atom") spezialisiert werden; er kann andrerseits

Bestandteil der Definition anderer Begriffe sein (z.B. Molekül, Bindung, Element, usw.).

Sucht man nun nach einer mathematischen Theorie, welche die Operationen der Abstraktion und der Konkretion formalisiert, dann findet man sie in der Theorie der "Funktionalen Applikation", eben im λ-Kalkül. Diese Arbeit ist Teil eines umfassenden Projekts, welches sich mit der Aufstellung fachsystematischer Netze (HAEFNER 1974) beschäftigt. Solche Netze entstehen durch inhaltliche Einschränkung und methodische Erweiterung semantischer Netze (siehe dazu BRUNNSTEIN 1975). Bei der Aufstellung fachsystematischer Netze geht es darum, die Begriffe eines Wissensgebietes und die zwischen den Begriffen herrschenden Relationen als verschiedene Ebenen zu repräsentieren; wenn wir dann etwa von der Kausalebene sprechen, meinen wir alle Kausalbeziehungen zwischen den Begriffen des Wissensgebietes.

Eine sehr wichtige Ebene ist die Definitionsebene, da in ihr die semantischen und linguistischen, die deduktiven und logischen Beziehungen der Begriffe untereinander erfasst werden. Mit der Problematik dieser Ebene beschäftigt sich die vorliegende Arbeit, allerdings nur insoweit, als Definitionen mit Hilfe des Formalismus und der Ideen des λ-Kalküls möglich sind. Bezüglich alternativer Definitionsverfahren siehe z.B. BIERWISCH und KIEFER 1969. Zunächst zeigen wir in einem kurzen Überblick, in welcher Weise der λ-Kalkül bisher auf Probleme der Datenverarbeitung und speziell auf linguistische Probleme angewandt wurde.

Nach Vorarbeiten durch SCHÖNFINKEL (1924), der eine variablenfreie Darstellung arithmetischer Gesetze suchte (und durch seine 'Kombinatoren' auch erreichte), begann vor allem durch CHURCH (1941) und CURRY (1958) eine systematische Untersuchung der Beziehungen zwischen freien und gebundenen Variablen, der Einsetzung und Abstraktion durch den von CHURCH eingeführten Abstraktionsoperator Lambda (λ). Versuche, den Kalkül als Grundlage der gesamten Mathematik und insbesondere der Logik anzupreisen, schlugen zunächst fehl, da durch das Auftauchen logischer Widersprüche gezeigt werden konnte, dass der Kalkül zu umfangreich war, einen zu großen Teil der Wirklichkeit diskriminationslos umfasste. Erst die Einführung von Typen (die das ursprüngliche, 'reine' Konzept des Kalküls zerstörte) konnte diesem Problem entgehen.

Die Anwendung des Kalküls in der Datenverarbeitung begann sehr früh und erwies sich als außerordentlich fruchtbar. MCCARTHY schuf mit seiner Programmiersprache LISP (1961) eine programmiertechnische Realisierung des Kalküls, wobei sich die Sprache als ungemein vielseitig verwendbar erwies und vor allem die Pionierleistungen der Datenverarbeitung (Sprachanalyse, künstliche Intelligenz) umfasste. Bald setzten auch theoretische Anwendungen ein. LANDIN (1966) übersetzte einen Teil von ALGOL in den λ-Kalkül und entwickelte eine Interpretationsmaschine dafür. SCOTT (1972 und viele andere Veröffentlichungen) schuf durch seine 'extensionalen Modelle des λ-Kalküls' eine Brücke zur gewohnten Datenverarbeitung, die in den üblichen Programmiersprachen mit einer Vielzahl von Typen hantiert. (Selbst LISP ist nicht ganz typenlos.) NEES (1973) wandte den Kalkül auf die Definition von Funktionen zur Beschreibung kognitiver Prozesse an, und BREKLE (1969, 1970) benutzte ihn in der Linguistik zur Erzeugung neuer Substantiva. Da unsere Arbeit von hier ihren Ausgangspunkt nimmt, soll auf diese beiden Autoren etwas ausführlicher eingegangen werden.

NEES benutzt den Kalkül zur Beschreibung zweier Funktionen eines Systems, die er **I** (für 'Internum') und **E** (für 'Externum') nennt. Dabei ist es seine Absicht, zu zeigen, dass der λ-Kalkül in der Gestalt der Notation, die man vor allem LANDIN verdankt, ein leistungsfähiges Werkzeug zur Beschreibung von kognitiven Systemen ... ist." NEES spezialisiert die sehr allgemeinen Systemgleichungen für I und E zunächst auf ein System zum Auf-, Um- und Abbau von Streckenkomplexen mit Hilfe eines graphischen Displaygerätes (I Maschine, E = Benutzer), und später auf ein Muster-Erkennungsprogramm zur Analyse hexagonaler Strukturen mit einer Anwendung auf die biologische Kybernetik (kontinuierliche Zentrierung eines Flecks). - Die Bedeutung der Arbeit liegt weniger im einer kalkülspezifischen Applikation als in dem Versuch, die Vorzüge und die Vielseitigkeit dieses Kalküls auch bei der Anwendung auf Probleme des 'Lebendigen' zu zeigen, denn bis jetzt war der λ-Kalkül meist ein streng mathematisches Mittel zum Beweis tiefliegender mathematischer Theoreme, oder ein bequemes Darstellungsmittel für spezielle Programmiersprachen.

BREKLE benutzt den λ-Kalkül zur Definition linguistischer Kategorien. Er stellt Funktionen mehrerer Variabler auf. Je nachdem, welche frei und welche gebunden sind, ergeben sich verschiedene linguistische Kategorien, von einfachen Nominalkompositionen bis zu komplexen satzsemantischen Strukturformeln. Bedeutet z.B.

ATT(x,E): das Objekt x hat die Eigenschaft E, dann gilt z.B. mit

E:= periodische Veränderung und f := λx.ATT(x,E):
(f JAHR) Jahreszeiten
(f MEERESHÖHE) = Gezeiten
(f LUFT) = Ton
(f LEBEN) = Biorhythmen

usw. Die runden Klammern bedeuten die Anwendung der Funktion f auf die nachfolgenden Parameter. Der erste Ausdruck ergibt den Term ATT(JAHR, periodische Veränderung), und das kann man, wie gezeigt, mit "Jahreszeiten" interpretieren. Ähnliches gilt für die anderen Beispiele.

Nimmt man jetzt umgekehrt x als fest (freie Variable) mit der Bedeutung *x := Mensch* und E als variabel (gebunden, ersetzbar) und nennt die Funktion jetzt g, dann ergehen sich durch Applikationen die folgenden zusammengesetzten Substantiva:
g := λE.ATT(x,E)
(g VERRÜCKT) = Irrer
(g WEIBLICH) = Frau
(g TOT) = Leiche
(g JUNG) = Kind

Dieses System lässt sich beliebig ausdehnen, beispielsweise auf Aktionen. So bedeute der Term x U y , dass sich x mit y unterhält. Daraus ergeben sich folgende Fälle:
λxy. xUy alle Gespräche (Unterhaltungen) zwischen zwei Menschen
(λxy.xUy PETER) alle Gespräche Peters mit jemand anderen
(λxy.xUy PETER PAUL) die Unterhaltung zwischen Peter und Paul

In BREKLE 1970 systematisiert der Autor seine Erkenntnisse bezüglich der Nominalkompositionen des Englischen. Dabei orientiert er sich strikt an der Realität, d.h. er versucht, das Vorhandene mit seiner Methode möglichst vollständig zu erfassen. Die eigentlichen Vorzüge dieses Verfahrens - die Schaffung neuer Begriffe - werden bei ihm nur als Möglichkeit angedeutet: "Das ... skizzierte Modell würde es jedoch ohne weiteres erlauben, Bedingungen zur Erzeugung weiterer - hier möglicherweise nicht erfasster - Typen aufzunehmen." Eine Untersuchung über die Abhängigkeiten und Strukturen der so abgeleiteten Begriffskomplexe findet bei BREKLE nicht statt. Ansonsten aber ist seine Arbeit einer der wichtigsten Ausgangspunkte der eigenen Untersuchungen.

Der BREKLEsche 'Aufbau der Welt' durch satzsemantische Strukturen macht keinen Gebrauch von der Theorie der Typen. Durch

Berücksichtigung der Typen (Klassen) der verwendeten Variablen und Funktionen gelangt man in einem systematischen Aufbau zu einer unendlichen Mannigfaltigkeit immer komplexerer semantischer Kategorien, die sich im einfachsten Fall auf nur zwei Grundklassen zurückführen lassen. CURRY und FEYS (1958) führen ein solches Programm teilweise durch, und zwar in zwei Beispielen:

Im ersten Beispiel gehen sie von den mathematischen Typen "Individuum" (oder Zahl, Argument, Objekt) und "Aussage" (oder Wahrheitswert) aus. Rein mathematisch ergeben sich aus der einfachen Gleichung $y = f(x)$ insgesamt vier verschiedene Funktionen, je nachdem, welche Klasse durch *f* in welche Klasse übergeführt wird.

Der Übergang I → I ergibt gewöhnliche Funktionen eines Individuums;

der Übergang I → A ergibt Eigenschaften oder Klassen;

der Übergang A → I ergibt Extraktoren (z.B. den Jota-Operator); und der Übergang A → A ergibt Wahrheitswertfunktionen oder logische Funktoren.

Nimmt man nun diese Funktionen als Argumente, dann gelangt man auf einer höheren Stufe zu einer weiteren Reihe von Kategorien, und diese Stufenfolge kann man beliebig weit fortsetzen (vorausgesetzt, es ergehen sich sinnvolle Kategorien, d.h. solche, die eine Entsprechung in der Realität haben oder haben könnten, die also interpretierbar sind)

Im zweiten Beispiel verwenden die Autoren die grammatikalischen Kategorien "**Nominalkomplex**" und "**Satz**" als Ausgangselemente und leiten daraus durch wiederholte Funktionalbildung eine Vielzahl grammatikalischer Kategorien ab. Im Unterschied zu BREKLE, der sich bei seinem Aufbau auf die Bedeutung freier Variabler stützt (und daher keine Systematik hat bzw. braucht), verwenden CURRY und FEYS nur gebundene (bedeutungslose) Variable, deren Typen durch kontrollierte Kombination eine systematische Komplexbildung zulassen.

In beiden Fällen fehlt aber eine Untersuchung über die wechselseitigen Abhängigkeiten der so generierten Kategorien, Komplexe und Strukturen. Begriffe stehen schließlich in irgendeiner Relation zueinander. Werden sie aus dem λ-Kalkül erzeugt, müssen diese Beziehungen auch mit dem λ-Kalkül konsistent bzw. aus ihm ableitbar sein. Wie wir in dieser Arbeit zeigen, ergibt sich aus der Methode der Attributivdefinition zwangsläufig eine Verbandsstruktur als Abhängigkeitsgeflecht der so erzeugten

Begriffe. Andere Autoren haben dies vermutet oder erwünscht. GOGUEN (1974) z.B. versucht, die Verbandsstruktur von Begriffen mit Hilfe eines etwas obskuren Wahrscheinlichkeitsverfahrens zu beweisen; die Interpretation des maximalen und des minimalen Elements ist aber sehr gekünstelt, die Ableitung zudem nicht zwingend. KINTSCH (1972) weist darauf hin, dass Begriffshierarchien im allgemeinen *keine* Verbände darstellen. Beide Autoren scheitern daran, dass sie von Begriffsdefinitionen ausgehen, die wir in unserer Arbeit als **Nominaldefinitionen** bezeichnen. Aus ihnen können keine zwingenden mathematischen (syntaktischen, formalen) Beziehungen abgeleitet werden. Erst das Konzept der **Attributivdefinition** ermöglicht eine rein mathematische (in diesen Fall: kombinatorische) Generierung von Begriffsverbünden. Die Unterscheidung zwischen diesen beiden Arten von Begriffsdefinitionen und ihre Bedeutung für die Untersuchung mathematischer Strukturen innerhalb von Begriffsfeldern wird in dieser Arbeit zum ersten Mal klar herausgestellt.

Die Beziehung zwischen λ-Kalkül und Verbänden wurde zum ersten Mal von SCOTT in verschiedenen Abhandlungen systematisch untersucht und für die Probleme der Informatik fruchtbar gemacht. SCOTT geht dabei von einem speziellen Modell für den λ-Kalkül aus, in dem die **Approximation von Funktionen** das zentrale Element darstellt, während in unserer Arbeit die **Generierung von Begriffsfeldern** angestrebt und erreicht wird. Dass dennoch auf einer abstrakten Ebene ähnliche Resultate erzielbar sind, zeigt die erstaunliche Flexibilität, Allgemeinheit und zugleich praktische Anwendbarkeit dieser mathematischen Theorie, die in die Informatik mehr Eingang fand als in die Mathematik, aus der sie hervorgegangen ist.

Unsere Arbeit geht weniger systematisch als mehr historisch-didaktisch vor. Manchmal werden verschiedene Zugangsmöglichkeiten zu einem Begriff oder Verfahren aufgezeigt. Mathematische Beweisverfahren wurden im allgemeinen nicht gebracht; vielmehr wurde das Innovatorische des Verfahrens betont und die praktische Anwendung an zahlreichen Beispielen gezeigt. Ausgangspunkt waren dabei die eigenen Untersuchungen über Logogramme und die Anwendung der Automatentheorie auf didaktische Strukturen in RIPOTA 1968 und 1974.

Die **Anwendbarkeit** der in dieser Arbeit entwickelten Konzepte und Verfahren liegt weniger in der konkreten Formulierung von Datenbanksystemen oder der Implementation semantischer Netze, als mehr in deren Vorstufe. Die meisten derartigen Systeme gehen von feststehenden

Begriffssystemen und den zugehörigen Beziehungsgeflechten aus und bieten Mittel und Wege, diese Strukturen durch geeignete mathematische Verfahren (z.B. Prädikatenkalkül) oder Programmtechniken (z.B. verkettete Listen) darzustellen. Über die Begriffe selbst wird aber meist nicht reflektiert. Der darzustellende Ausschnitt der Wirklichkeit wird so übernommen, wie er einem geeignet erscheint. In unserer Arbeit versuchen wir, diese Vorstufe (die wir **semantische Analyse** nennen können) zu systematisieren, die Definition von Begriffen auf eine exakte Grundlage zu stellen und die wechselseitigen Herleitbarkeiten von Begriffen sowie die mathematischen Strukturen der dabei auftretenden Beziehungen mittels geeigneter Formalismen zu erfassen. Dass dabei starke Bezüge zur Linguistik (Kapitel 15) und zur Wissenschaftstheorie (Kapitel 9) auftreten, ist verständlich. (Es ist uns dabei, eher nebenbei, gelungen, die viel umstrittenen Begriffe "Intension" und "Extension" auf sehr einfache Weise zu definieren und zu quantifizieren.) Erst nach Durchlaufen dieser Stufe ist an eine Aufstellung eines semantischen Netzes und dessen Implementierung zu denken. Die programmtechnische Realisierung kann in jeder dafür geeigneten Programmiersprache vorgenommen werden. Obwohl wir für unsere Untersuchungen den λ-Kalkül verwenden, muss die Implementationssprache von diesem mathematischen Verfahren nicht unbedingt Gebrauch machen. Man vergesse nicht, dass die zum λ-Kalkül äquivalente Theorie die **Theorie der Kombinatoren** ist, dass also kombinatorische Verfahren für unsere Arbeit von besonderer Wichtigkeit sind. Und diese sind in praktisch jeder Programmiersprache realisierbar. Insbesondere für den Konzeptgenerator (Kapitel 11) erscheint uns SIMULA als besonders geeignet, da durch das Klassenkonzept jeder Oberbegriff als Klasse, jeder Unterbegriff als Inkarnation dieser Klasse dargestellt werden kann.

Bei der Darstellung von Sachverhalten in semantischen Netzen können wir grundsätzlich zwei Idealtypen darzustellender Systeme unterscheiden. Zur ersten Gruppe gehören die **abstrakten, statischen** Systeme. Hierher zählen wir insbesondere Wissensgebiete, deren Grundbegriffe in systematischer Weise dargestellt werden sollen, beispielsweise die Taxonomie des Pflanzenreichs, die Grundlagen der Atomphysik, die Definitionen soziologischer Begriffe, kurzum: alles, was in der bereits erwähnten "Definitionsebene" repräsentiert werden kann. Für diese Art von Systemen sind unsere Verfahren besonders geeignet, da sie speziell dafür entwickelt wurden. Die Repräsentation eines derartigen Systems kann, nach Analyse durch unsere Verfahren, programmtechnisch oder auch graphisch erfolgen; sie ist vor allen für didaktische Zwecke geeignet, insbesondere zum Aufbau von Lehrprogrammen.

Zur zweiten Gruppe gehören die **konkreten, dynamischen** Systeme. Dazu zählen die meisten, bis jetzt entwickelten Datenbank-, Informations- und Frage-Antwort-Systeme. Sie geben einen beschränkten, konkreten, kontinuierlich veränderlichen Ausschnitt der Wirklichkeit wieder, beispielsweise die Lage geometrischer Objekte auf einer Ebene, die Entwicklung von Aktienkursen oder die Buchungen einer Fluggesellschaft. Derartige Systeme bedürfen auf jeden Fall (wegen der ständigen Veränderung der Datenbasis) einer Rechner-Implementierung mit geeigneten Aufbereitungs- und Folgerungsprogrammen. Auch hier können unsere Verfahren eine Vorklärung der zu verwendenden Grunddaten erbringen (siehe dazu die Kapitel 1 bis 18), während diese zur eigentlichen Datenmanipulation weniger geeignet sind. Es kann z.B. die Anzahl der Grunddaten durch die Schaffung hinreichend abstrakter Oberbegriffe eingeschränkt werden; als Nebeneffekt dieser Einsparung erkennt man auch noch Zusammenhänge, die nicht unmittelbar ein sichtig waren (siehe unsere Beispiele zum Kapitel 18.1 und 18.2). Auch können die Möglichkeiten (das <u>Potential</u>) eines solchen Systems durch unsere Verfahren besser erforscht werden, da z.B. alle möglichen (und sinnvollen) Zustandsveränderungen kombinatorisch erzeugt werden. - Mit diesen Ausführungen sollten nur Anregungen gegeben werden; in den Beispielen wird dies in der Praxis gezeigt.

3. Der typenfreie Kalkül

Bezüglich einer ausführlichen Darstellung des λ-Kalküls und seiner Geschichte sei auf die einschlägigen Lehrbücher verwiesen: CURRY 1958 und 1972, STENLUND 1972. Wir beschränken uns in dieser Darstellung auf das Allernotwendigste, zumal wir von dem formalen Apparat des Kalküls im allgemeinen keinen Gebrauch machen.

Der Lambda-Kalkül ist die Theorie der <u>Funktionen</u>, ihrer Definition und ihrer Anwendung. Sie ist besonders einfach und grundlegend, da es in ihr zunächst nur einen Datentypus gibt - den Typ 'Funktion', und nur zwei Operationen - Abstraktion und Substitution.

Die Abstraktion dient der Funktionsdefinition in der folgenden Art. Was in der gewöhnlichen Schreibweise

$y = f(x)$ heißt, wird im λ-Kalkül als $y := \lambda x.fx$ geschrieben, wodurch die **gebundenen Variablen** (hier: x) von den **freien Variablen** (hier: f) eindeutig unterschieden werden.

Eine weitere Eigentümlichkeit des λ-Kalküls liegt darin, dass alle Formeln linksassoziativ sind, d.h., ein Ausdruck der Form f g x im λ-Kalkül in der Form ((fg)x) gelesen wird.

Mit anderen Worten: Zuerst wird die Funktion f(g) gebildet (sie heiße f'), und dann erst die Funktion f'(x). Eine derartige Interpretation ist in der gewöhnlichen mathematischen Schreibweise nicht möglich, da dort Funktionen nur auf Zahlen angewandt werden können (oder auf geometrische Objekte, wenn es sich um geometrische Transformationen handelt), dass also ein Unterschied zwischen dem Datentypus "Funktion" und dem Datentypus "Argument" gemacht wird. Dieser Unterschied fällt im λ-Kalkül fort - alles ist "Funktion", in unserem Beispiel also auch x. Dadurch wird die semantische Interpretation der λ-Formeln erleichtert; andrerseits führt ein solches Konzept bei einer konsequenten Anwendung auf Bereiche der Logik zu den bekannten Paradoxien vom Lügner, wodurch die Einführung von Typen und Hierarchien erforderlich wird. Doch soll uns dies hier nicht kümmern.

Die Linksassoziativität bedeutet weiterhin, dass jede Funktion jeweils nur ein Argument haben kann. Infolge des ungewohnten Funktionsbildungsmechanismus im λ-Kalkül kann man aber einen Term der Form fxyz als Funktion f von drei Argumenten x, y und z interpretieren, ohne jeweils auf die richtige Klammerung und die Bildung neuer Funktionen zu achten.

Wendet man nun eine vorher definierte Funktion f auf ein Argument a an, so deutet man dies durch (f a) an; man nennt den Vorgang **Applikation** (Anwendung) oder **Reduktion**. Dabei werden die gebundenen Variablen in f der Reihe nach durch die außen angegeben Werte ersetzt. So gibt es im λ-Kalkül einen Term **B**, den Funktions-Kompositor, der folgendermaßen definiert ist: (Operatoren immer in **fett**)

B := λxyz.x(yz)

Wendet man B auf drei Argumente f, g und a an, dann wird der Lambda-Ausdruck reduziert, und zwar in diesem Fall vollreduziert:

(B f g a) = (λ xyz [x(y,z)] f g a) = f(g a)

(in der gewohnten Schreibweise: f(g(a))).

Sind weniger Werte als gebundene Variable angegeben, dann wird der Lambda-Ausdruck nur teilreduziert:

(**B** f g) = ... = λz.f(g z) = λx.f(g x)

(Beim letzten Übergang wurde die gebundene Variable z umbenannt in x. Das ist immer dann möglich, wenn es keine Kollision mit freien Variablen gibt.)

Sind mehr Werte als gebundene Variable angegeben, dann bleiben die nicht verwendeten Werte einfach stehen. Beispiel:

(**B** f g a b) = ... = f(g a)b

d.h., f kann als Funktion von zwei Argumenten interpretiert werden; das erste Argument ist seinerseits eine Funktion, g(a), das zweite die Größe b.

Ein Ausdruck, in dem alle möglichen Reduktionen vollzogen wurden, heißt (oder hat eine) **Normalform**. So ist f(ga) eine Normalform, dagegen (**B** f g a) nicht; λxyz.x(yz) ist eine Normalform (hier gibt es nichts zu reduzieren), dagegen λx[xx]λx[xx] nicht, weil sich bei Reduktion wieder der gleiche Ausdruck ergibt, der unendlich oft reduziert werden kann. (Im letzteren Beispiel wurden die äußeren Klammern weggelassen, eine Vereinbarung, die der Bequemlichkeit halber manchmal eingehalten wird.)

Da wir im Verlauf unserer Arbeit den 'reinen' Kalkül selten verwenden und mehr zur gewohnten Schreibweise aus Mathematik und

Datenverarbeitung zurückkehren, führen wir an dieser Stelle einige **Modifikationen** ein.

Wenn wir früher für eine Reduktion schrieben: (f a b c), so werden wir ab jetzt manchmal der Deutlichkeit halber Funktionen und Argumente gegeneinander absetzen, erstere gegen letztere durch einen Doppelpunkt, letztere untereinander durch Kommatas:

(f: a,b,c)

Bei Teilreduktionen müssten wir die Variablen in Bereich des Lambda-Operators umstellen. Das ist etwas mühsam und erfordert außerdem eine ständige Neudefinition. Wir deuten Teilreduktionen jetzt durch Punkte für die nicht zu reduzierenden Variablen an:

(f: a, . , c) bedeutet eine Reduktion nach der 1. und nach der 3. gebundenen Variablen; die zweite bleibt unreduziert.

Bei der Handhabung des λ-Kalküls haben wir implizit von einem Verfahren Gebrauch gemacht, das als **Globalisierung der Variablen** bezeichnet wird. Dabei werden alle Lambda-Operatoren aus inneren Termen soweit nach außen verlagert, wie dies möglich ist. Voraussetzung dafür ist eine vorherige innere Reduktion, bis eine Normalform (also eine nicht mehr reduzierbare Formel) entsteht. Das abstrakte Globalisierungsschema sieht so aus:

λx[... (λy[...])...] => λxλy [...(...)...]

Eine derartige Globalisierung ist immer dann gefahrlos, wenn die globalisierten Variablen keine Funktionen sind, die selbst keine weiteren Argumente rechts neben sich haben. Die Globalisierung darf aber erst vorgenommen werden, wenn alle inneren Reduktionen ausgeführt sind. (Dass man hier die im λ-Kalkül übliche strikte links-nach-rechts-Reduzierungsfolge verlässt und zu einer innen-nach-außen-Reduktionsweise übergeht, ist dann gerechtfertigt, wenn alle Ausdrücke Normalformen haben, also ein unendlicher Einsetzungsprozess ausgeschlossen ist. In unseren Theorien setzen wir dies deshalb voraus, weil Termen ohne Normalformen kein sinnvoller Begriff entspricht, wir jedoch eine Theorie der Begriffsdefinitionen aufstellen, in der sinnlose Begriffe nach Möglichkeit von vornherein ausgeschlossen sein sollen.)

Da wir im weiteren Verlauf unserer Untersuchungen den typenfreien Kalkül verlassen und zu einem typisierten Kalkül übergehen werden, kommt der Globalisierung besondere Bedeutung zu: Bei rekursiven Einsetzungen