Informatik als Grundbildung

H. Wedekind · E. Ortner · R. Inhetveen

Der Einfluss der Informatik auf unseren Alltag ist enorm. Der Einfluss der Informatik auf die Lehrpläne unserer allgemeinbildenden Schulen ist gering. Man mag das mit der Jugendlichkeit des Faches zu begründen versuchen und einen beliebten Vergleich mit der Mathematik anstreben, die heute ein wesentlicher Teil der Grundbildung ist. Was die Terminologie anbetrifft, so lehnen wir uns eng an das Deutsche PISA–Konsortium an [1].

Grundbildung, um die es uns geht, vermittelt eine Basisqualifikation und ist nicht mit Allgemeinbildung zu verwechseln. Es geht wie in der PISA-Studie um Basiskompetenzen, die Voraussetzung für den Erwerb einer Allgemeinbildung sind. Allgemeinbildung ist durch einen kaum auszumachenden Horizont begrenzt und steht deshalb nicht zur Debatte.

Bildung ist ein Zentralbegriff der Aufklärung. „Ein Mensch gilt als gebildet," heißt es noch bei J.G. Fichte 1808 in seinen berühmten Reden an die deutsche Nation „wenn er ganzheitlich aus sich selbst heraus will, was allgemein und sittlich erfordert ist". Das idealistische Bildungsziel „Sittlichkeit" eines Fichte löst heute nur noch ein Schmunzeln aus. Was von Fichte und der Aufklärung aber geblieben ist, ist die Forderung nach Selbsttätigkeit. „Aus sich selbst heraus" muss etwas geschehen, was heute oftmals vergessen wird. Zum Selbsttätigwerden bedarf es aber einer Orientierung mit dem dazugehörigen Wissen (Mittelstraß). Bloßes Verfügungswissen wird dem zugewiesen, was wir Ausbildung („education") nennen. Wenn wir Terhart in seiner Nach-PISA-Studie folgen ([6], S. 22), ist Sittlichkeit heute „durch individuelle Handlungsfähigkeit zum Lösen von Problemen in der modernen Welt" zu ersetzen. Es geht nach Terhart in der Grundbildung um „funktionale Kompetenzen der Welterschließung und Problembewältigung". Und damit sind wir sehr nahe am anglo-amerikanischen Begriff „literacy", der sich von dem einfachen Lese/Schreibvermögen zu dem entwickelt hat, was im Deutschen seit den Zeiten der Aufklärung „Bildung" heißt. „Basic literacy" (Grundbildung) steht speziellen Bildungsformen wie z.B. der „computer literacy" gegenüber. Das Programm „Informatik in den Schulen" hat sich dieser Form angenommen. Rechneraufbau, Algorithmen, Programmiersprachen und alles rund um den PC stehen im Mittelpunkt eines Unterrichts, für den in der Regel sogar ein eigenes Schulfach vorgesehen ist. Es geht uns nicht um das Problem „computer literacy" oder Informatik in den Schulen. Es geht uns um das Thema „Informatik als Grundbildung". Gäbe es die Informatik als Grundbildung schon, dann würde sich diese Arbeit erübrigen. Die Frage, die von uns zu beantworten ist, lautet: *Was muss in einer Informatik-Grundbildung methodisch vorangestellt werden, damit ein spezielles Bildungsangebot im Sinne einer „computer literacy" überhaupt verstanden wird? Es geht uns um eine logische Propädeutik, wie sie von Kamlah u. Lorenzen* [2] *entwickelt wurde, zugeschnitten*

DOI 10.1007/s00287-004-0383-z

Prof. Dr. H. Wedekind
Universität Erlangen-Nürnberg,
Institut für Informatik,
E-Mail: wedekind@informatik.uni-erlangen.de

E. Ortner · R. Inhetveen
TU Darmstadt,
Fachbereich Recht- und Wirtschaftswissenschaften

auf das Fach Informatik, das viele andere Fächer penetriert hat.

Die Beantwortung dieser Frage ist nicht einfach und bedarf einer längeren Darstellung, wenn wir nicht in einer Allgemeinheit stecken bleiben wollen. Es werden ohne Anspruch auf Vollständigkeit sechs Lehrstücke beschrieben, die alle eine entscheidende Rolle spielen – nicht nur in einer dann folgenden Informatik an Schulen und Hochschulen. Andere Fächer, die konstruktiv, d.h. schrittweise, zirkelfrei und alles explizit machend gelehrt werden, hängen in gleicher Weise von diesen Lehrstücken ab. Die sechs Themen, die im Folgenden nur skizzenhaft dargestellt werden, um dann in weiteren Folgen auf sie näher einzugehen, lauten:

- Schema und Ausprägung,
- Bildung von Elementarsätzen,
- Gleichheit und Abstraktion,
- Objektsprache / Metasprache,
- Namensgebung und Kennzeichnung,
- Logik und Geltungssicherung von Behauptungen.

Es handelt sich um einen sechs-dimensionalen Raum, dessen Hauptkoordinaten abzustecken sind. Dieser Raum hat auch eine zeitliche Tiefe, die in dieser Aufzählung verloren geht. Als Ersatz für diese siebte Dimension mag die sechste gelten, seit der Aufklärung die problematischste in unserem Geschäft.

Wir sind uns bewusst, dass eine Forderung „Informatik als Grundbildung“ für viele eine Ungeheurlichkeit darstellt und deshalb hoffentlich zu einer regen Diskussion Anlass gibt. Im Sinne der PISA-Studie [1], die drei Grundbildungsformen kennt („reading literacy“, „mathematical literacy“ und „natural science literacy“), soll eine vierte Form, eine „informatical literacy“ ins Leben gerufen werden. Dabei sei es völlig unbenommen, in welchem Rahmen der Lehrstoff dargeboten wird. Der Mathematik-Unterricht ist hierfür prima facie genauso geeignet wie der Deutsch-Unterricht. Man kann auch an den Informatik-Unterricht denken, falls dieser sich als geeignet erweisen sollte.

Ein Hauptanliegen dieses Beitrags betrifft auch das Thema „Geschlechterunterschiede und Basiskompetenzen“, ein zentrales Kapitel der PISA-Studie [1]. Es ist auffallend, dass Mädchen in Sachen Sprachkompetenz („reading literacy“) international gegenüber den Jungen eine hohe Dominanz aufweisen. In der Mathematik liegt der Fall umgekehrt, nur nicht ganz so drastisch. Die darzustellende Informatik-Grundbildung ist invariant bezüglich des Antagonismus „Sprache vs. Mathematik“, oder krasser „weiblich vs. männlich“. Fundamental-wissenschaftlich, besser gesagt, philosophisch gesehen, ist diese Unterscheidung auch ohne Sinn. Sie löst sich, dialektisch gesprochen, in der Synthese einer Informatik-Grundbildung auf. Das ist natürlich eine Hypothese und eine Hoffnung, und somit vorerst eine Spekulation. An einem empirischen Nachweis im Sinne der PISA-Forschung ist nicht vorbeizukommen. Aber erst muss etwas vorhanden sein, bevor man es testet. Dabei ist klar, dass eine Aufbereitung zu einem vermittelbaren Lehrstoff (Didaktik) eine entscheidende Rolle spielt.

Hier nun sechs Kurzfassungen.

Schema und Ausprägung

Das Begriffspaar „Schema – Ausprägung“ spielt in der Informatik, aber auch in anderen Wissenschaften eine zentrale Rolle. Das Paar wird aber in der Informatik in aller Strenge thematisiert. *Programme* als Schemata haben *Kontrollfäden* („threads of control“) als Ausprägungen. *Classes* sind Schemata und *Objekte* ihre Ausprägungen. *Relationen* in Datenbanken sind konzeptionelle Schemata und haben *Tupeln* als Ausprägungen, das Klicken auf einem Bildschirm folgt einem Zeigehandlungsschema usw. Je nach Sprachmodell und Autor spricht man allgemeiner auch von „competence“ (Schema) und „performance“ (Ausprägung) (**Chomsky**), von „langue“ (Schema) und „parole“ (Ausprägung) (**de Saussure**) und von „type“ (Schema) und „token“ (Ausprägung) (**Peirce**). Schemata sind universelle Beschreibungen von Gegenständen unsere Welt. Ausprägungen sind singulär. Bildung kann als Schemaerwerb aufgefasst werden, natürlich unter der Maßgabe der Selbsttätigkeit („self-acquisition“).

Bildung von Elementarsätzen

„Fido ist ein Hund“ oder „Heute ist Donnerstag“ sind Elementarsätze, die erst in der Logik zu komplexen Sätzen verknüpft werden. Das Schema eines Elementarsatzes lautet: N ε P mit N als einem Namen, allgemein Nominator genannt, und P als einem Prädikator. Es wird nicht Prädikat gesagt, um einer Verwechselung mit dem Terminus „Prädikat“

einer empirischen Grammatik vorzubeugen. Das Zeichen ‚ε' heißt die Kopula und wird im Deutschen durch das vieldeutige Wort ‚ist' wiedergegeben. Eine Hauptaufgabe ist, diese Vieldeutigkeit herauszuarbeiten. An Elementarsätzen wird die wichtige Unterscheidung von Extension (Umfang) und Intension (Inhalt) eines Prädikators eingeübt.

Gleichheit und Abstraktion

Eine explizite Darstellung des Abstraktionsvorgangs ist ein Kernstück der Informatik-Grundbildung. Wenn in einem Text „Margarine" durch „Butter" ersetzt werden darf, ohne die Geltung des Textes zu verändern, dann liegt eine Abstraktion von Prädikatoren zu einem Begriff vor. Abstraktion baut auf der Spezifikation einer Äquivalenzrelation auf. Im Falle von „Margarine" und „Butter" spricht man bei sprachlicher Gleichbehandlung von Synonymität. Berühmte Abstraktionen wie: Von den Ziffern (Zählzeichen) zur Zahl, von Folgen zu Mengen, von Termen zu Funktionen betreffen zwar die mathematische Grundbildung. Dass jede gute Schnittstelle eine Abstraktion ist im Hinblick auf äquivalent zu betrachtende Implementierungen, das ist eine Einsicht, die früh zu vermitteln ist. Es sollte deutlich gemacht werden, dass die Abstraktion beide Grundbildungen, die mathematische und die informatische miteinander verbindet.

In der Informatik-Grundbildung sollte mit höchster Genauigkeit gearbeitet werden. Ungenauigkeit verursacht Unsicherheit. Deshalb ist in diesem Kapitel der Begriff „Information" präzise als Abstraktion einzuführen.

Objektsprache/Metasprache

Wenn Gegenstände einer Sprache selbst zum Gegenstand des Sprechens werden, ist diese Sprache gesondert als Metasprache auszuzeichnen. Die objektsprachliche Aussage *Peter ist kurz* soll bedeuten, dass ein Individuum mit dem Namen Peter von kleinem Wuchs ist. Auf die Ebene einer Metasprache gehoben, muss geschrieben werden: *„Peter" ist kurz*. Gemeint ist: Das Wort, besser das Wortschema „Peter" ist kurz. Wortschemata, oder gar ganze Satzschemata, sind die Objekte einer Metasprache und müssen als solche durch Anführungszeichen hervorgehoben werden, um Missverständnisse und Paradoxien zu vermeiden. Der Satz *„kurz" ist kurz* macht Sinn und zeigt das wichtige Phänomen einer Selbstbeschreibung (Autologie) auf. Der Satz *kurz ist kurz* macht hingegen keinen Sinn. Wichtig ist herauszustellen, dass Schemata als Namen auftreten. Der zugewiese Prädikator *kurz* ist objektsprachlich ein Objekt-Prädikator, metasprachlich aber ein Meta-Prädikator. Das Reden über Sprache ist typisch für die Informatik, die dann Meta-Informatik genannt werden kann. Dieses Phänomen gibt es drüben bei Sprache und Mathematik als Grundbildung auch, nicht aber in den Naturwissenschaften, die objektsprachlich verhaftet bleiben. Die empirische Grammatik z.B. des Deutschen oder des Lateinischen ist eine Sprachbeschreibung und eine metasprachliche Grundlage, auf die in der informatischen Grundbildung zurückgegriffen werden sollte.

Namensgebung und Kennzeichnung

„Die Hauptstadt Frankreichs" ist eine echte Kennzeichnung, weil ein Objekt, das existiert und eindeutig ist, beschrieben wird und mit dem Eigennamen Paris versehen werden kann. Jede URL ist eine echte Kennzeichnung. Existenz, aber nicht Eindeutigkeit ist häufig im praktischen Leben gegeben. Man spricht dann von potentiellen Kennzeichnungen, wenn Eindeutigkeit nicht garantiert werden kann. Die standesamtlichen Festdaten wie Name, Geburtstag und -ort etc. sind streng genommen nur eine potentielle Kennzeichnung. Das Aktenzeichen, das beim Standesamt vergeben wird, macht den Registriervorgang aber eindeutig. In der Informatik sind echte und potentielle Kennzeichnungen gang und gäbe. Ein Zusammenbauteil wird zum Beispiel durch seine Komponenten echt oder potentiell gekennzeichnet. Wichtig ist, die große Bedeutung der Kennzeichnung in unserem Leben herauszuarbeiten. Leben wir doch in einer nahezu namenlosen Welt. Geographische und astronomische Objekte tragen zuweilen einen Eigennamen. Hinzu kommen noch die Lebewesen Mensch und manchmal auch Tier. Zu allen anderen Objekten muss zwecks Referenzierung eine Kennzeichnung aufgebaut werden, die genauso behandelt werden kann wie Eigennamen.

Logik und Geltungssicherung von Behauptungen

Man sagt, die Logik sei kurz, leicht und alt. Das Prädikat „kurz" mag stimmen, „leicht" ist mehr als ein Irrtum, der dazu geführt hat, Logik aus dem Unterricht seit Wilhelm von Humboldt zu verbannen. Das Prädikat „alt" stimmt. Die Logik ist rund 2500 Jahre alt. Sie wurde aber eingeführt, um den Dialog um zusammengesetzte Elementarsätze, die behauptet werden, zu steuern. Das Wort „Kontrolllogik" ist, so gesehen, ein Pleonasmus, ein „weißer Schimmel". Wer behauptet, so lautet der Grundsatz der Logik mit einer tiefen erzieherischen Wirkung, der muss auch Belege herbeischaffen, um zu zeigen, dass das, was ausgesagt wird, auch stimmt. Es geht um Geltung. Man kann auch das stark belastete Wort „Wahrheit" benutzen. Eine Wahrheit fällt nicht vom Himmel. Sie muss in der Regel schwer erarbeitet werden. Logik in einer informatischen Grundbildung darf nicht verwechselt werden mit einer Logik, die heute die Theorie der Informatik bestimmt. Die Verwechselung von Theorie und Grundlagen ist ein großes Missverständnis des Faches, das beseitigt werden sollte. In dem bekannten Lexikon der Philosophie von Mauthner steht unter dem Stichwort „Logik der Tatsachen" die eindringliche Mahnung: „Nicht nur Hegel, ..., hat sich der Sünde schuldig gemacht, Logik außerhalb der Sprache zu suchen".

Teil I: Schema und Ausprägung

Gliederung:

1 Schema und Ausprägung als Gegenstände einer Rede.
2 Schemaerwerb und Bildung.
3 Schemaerwerb und Gerechtigkeit

1. Schema und Ausprägung als Gegenstände einer Rede

Dem Begriffspaar „ Schema – Ausprägung" ist in der Informatik eine dominante Stellung zugewiesen worden.

Rechner-Programme werden als Schema eines Ablaufs oder Vorgangs aufgefasst. Zur Laufzeit eines Programmschemas werden Ausprägungen („instances") erzeugt. Der Gedanke des zeitlichen Ablaufs einer Ausprägung wird auch durch den Begriff „ Faden" bzw. „Ablauffaden" („thread", „thread of control") wiedergegeben [4]. Der Bewegungsablauf eines Roboterarms ist eine Ausprägung eines Programms, das als Schema vorhanden ist. Für Datenbanksysteme werden konzeptionelle Schemata häufig als relationale Schemata entwickelt. Tupeln oder Zeilen, die zu einer Relation gehören sind ihre Ausprägungen. Eine Lehrer-Relation oder eine Lehrer-Tabelle ist, so gesehen, ein Schema, die einzelnen Zeilen, für jeden Lehrer genau eine, sind die Ausprägungen dieser Relation. Abstrakte Datentypen sind wie alle Typen Schemata. Betrachten wir z.B. den Abstrakten Datentyp „Quadrat", objektorientiert dargestellt durch eine *Class* mit einer Methode zum Zeichnen von Linien. Ein besonderes Quadrat, das z.B. herausgezeichnet wird, ist eine Ausprägung, die zuweilen auch *Objekt* einer *Class* „Quadrat" genannt wird. Alle einfachen Datentypen der Programmiersprachen sind Schemata. Der Typ *Integer* z.B. ist ein Schema für die Ausprägungen 1, 2, 3, ..., n.

Die Beispiele zeigen, dass das Begriffspaar „Schema-Ausprägung" zum Kern der Informatik gehört. Aus diesem Grunde ist auch berechtigt vorgeschlagen worden, das Fach *In.forma.tik* mit seinem Basiswort *Form*, vom Lateinischen *forma*, durch *In.schema.tik* zu ersetzen. Denn das griechische Wort *schema* (σχημα) entspricht dem lateinischen Wort *forma*. Der Begriff „Form" wird in den Wissenschaften sehr vielschichtig benutzt. Man stellt z.B. gerne Form und Inhalt (Materie) gegenüber. Logiker sprechen von Schlüssen kraft der Form, also von Gültigkeiten, die allein schon aus formalen Gründen gesichert sind. Man könnte aber auch von Schlüssen kraft des Schemas und von schematischen Gründen sprechen. Es wäre wünschenswert, wenn der Formbegriff durch den Schemabegriff expliziert würde. Man weiß dann sofort, dass es nicht um Geometrie und den dort dominanten Formbegriff geht. Insofern ist das Wort *Inschematik* ernst, und nicht nur als ein kleiner Scherz am Rande aufzufassen.

Was ist nun in allgemeiner Sprache ein Schema und was seine Ausprägungen? Von K. Lorenz [4] stammt die folgende, einfache Definition: Ein Schema stellt einen universellen (allgemeinen), eine Ausprägung einen singulären (besonderen) Aspekt eines Gegenstands einer Rede dar. Das Wort *Gegenstand* wie auch das aus dem Lateinischen stammende Fremdwort *Objekt* steht für nichts Geheimnisvolles. Gegenstände kommen in einer menschlichen Rede vor. „Nur wo menschliche Rede ist, werden Gegenstände von anderen Gegenständen unterschieden", heißt es in der logischen Propädeutik von Kamlah/Lorenzen [2] ganz ohne ontologisches Pathos. So wie wir sagen „ Dieser Gegenstand ist ein Hammer", also das Wort *Gegenstand* als Verlängerung des Demonstrativpronomens *dies* benutzen, wollen wir sagen „dieser Gegenstand ist ein Schema“, wenn wir z.B. das Schema einer Datenbank meinen, oder wir sagen „dieser Gegenstand ist eine Ausprägung“, wenn ein Tupel oder eine Zeile einer Relation bzw. Tabelle gemeint ist. Statt „dies ist ein P“, wobei P für Hammer, Schema, Ausprägung oder für was immer stehen mag, redet man aus bloß syntaktischen Gründen in der Form (Schema) „dieser Gegenstand ist ein P“, ohne dass der Inhalt verändert wird.

Das Begriffspaar „Schema – Ausprägung“ hat eine lange Tradition in der Geschichte der Sprachlehre, wobei unterschiedliche Wörter benutzt wurden, je nachdem, welcher Rahmen um die vorgetragene Sprachlehre gelegt wurde. Der berühmte amerikanische Linguist N. Chomsky (geb. 1928) nannte das Begriffspaar „competence – performance“. Mit einem Schema ist Kompetenz verbunden, eine Ausprägung zu vollziehen („to perform“). Statt Ausprägung werden in allgemeineren Sprachrahmen auch die Wörter *Aktualisierung* bzw. Instantiierung („occurrence“ und „instantiation“) benutzt. Weiter zurück in der Geschichte ist der Schweizer Linguist F. de Saussure (1857–1913) zu nennen, der das Begriffspaar „langue – parole“ (Sprache-Rede) prägte. Eine ganze Sprache, gemeint ist die Grammatik einer Sprache plus ihrem Wortschatz, wurde von de Saussure als ein universeller Aspekt, als ein Schema gesehen, unter dem tatsächliche Rede auch in beliebigen Wiederholungen ausgeführt werden kann. Noch weiter zurück ist der amerikanische Logiker C.S. Peirce (1839–1914) zu erwähnen, der die Wörter „typen – token“ insbesondere im Hinblick auf Symbole einführte. Den schematische Aspekt eines Symbols nannte er „type“, seine Aktualisierung z.B. in einem hingeschriebenen Zeichen „token“. Seit den Zeiten von Peirce, und nicht erst seit dem Aufkommen moderner Programmiersprachen, wird *integer* ein Typ und die Zeichen 1, 2, 3, ..., n Token oder Ausprägungen genannt.

Es war für die moderne Weiterentwicklung des Begriffspaars von großer Bedeutung, Schema und Ausprägung in die Lehre von der Natur des Menschen, in eine Anthropologie sagt man gelehrt, einzubeziehen [4]. K. Lorenz (geb.1932) ist dies gelungen, indem er von der Elementarsituation eines Dialogs zwischen einem aktiven Sprecher und einem passiven Hörer ausging. Vom Sprecher, also von der Person, die eine Sprechhandlung vollzieht („to perform“), wird gesagt, dass sie eine Ausprägung erzeugt. Vom Hörer, also der Person, für die es die Sprechhandlung passiv gibt, wird gesagt, dass sie ein Schema der Handlung als etwas Allgemeines erkennt, um die Sprechhandlung zu verstehen. Der Sprecher produziert eine Ausprägung nach einem Schema, das der Hörer verfügbar haben muss, um die Ausprägung zu verstehen. Beide, Sprecher und Hörer, haben ein Schema gemeinsam, der eine, um zu erzeugen („to perform“), der andere, um zu erkennen („to recognize“) und zu verstehen („to understand“). Sie treten ein in eine Kommunikation (vom Lateinischen communis: gemeinsam).

Eine Schreiber/Leser-Situation ist ähnlich der Situation zwischen einem Sprecher und einem Hörer. Wir wollen eine Schreiber/Leser-Situation darstellen, um ein Nicht-Verstehen zu exemplifizieren. Ein Schreiber möge die für einen Leser unlesbare Zeichenfolge *Haus* an eine Tafel schreiben. Die Kreidespuren auf der Tafel sind keine Ausprägung des Zeichenschemas *Haus*, weil das Schema für den Leser elementar nicht erkennbar, nicht lesbar ist. Unlesbarkeit ist als Phänomen allseits bekannt. Nehmen wir das Beispiel der chinesischen Schriftzeichen (Piktogramme). Europäer können die Zeichen im allgemeinen nicht lesen, weil ihnen die Schemata nicht verfügbar sind. In einer Gesellschaft, die sich Wissensgesellschaft nennt, müsste man sagen „Die Europäer *wissen* nicht, mit Piktogrammen umzugehen“. Statt k*önnen* ist heute *wissen* gefragt. „Die Grammatik des Wortes *wissen* ist offenbar eng verwandt der Grammatik der Worte *können, imstande sein*. Aber auch eng verwandt der des Wortes *verstehen.* (Eine Technik beherrschen)“ sagte der Philosoph Ludwig Wittgenstein im § 150 seiner philosophischen Untersuchungen. Von dem einflussreichen Wittgenstein wird gleich unten nochmals die Rede sein. Er ermahnt uns aber mit dem Wort *wissen* sorgfältig umzugehen. Man stelle sich vor, das Wort *Könnensgesellschaft* würde demnächst eingeführt, weil *können* ja sehr nah bei *wissen* liegt. Ob die Gesellschaft, in der so geredet wird, dann endlich merkt, wie albern das alles klingt?

Es ist heute üblich, Schemata von drei verschiedenen Standpunkten aus zu betrachten. Man nennt sie pragmatisch, semantisch und syntaktisch. Pragmatisch heißt ein Standpunkt, wenn danach gefragt wird, was man mit einem Schema anfangen kann. Semantisch heißt der Aspekt, wenn nach der Bedeutung gefragt wird, und von Syntax ist die Rede, wenn zur Debatte steht, ob das Schema auch korrekt, d.h. regelgerecht (man sagt auch wohlgeformt) gebildet wurde. Eine ganzheitliche Sicht stellt sich auf alle drei Standpunkte. Ein großes methodische Problem entsteht aber, wenn die Reihenfolge des Betrachtens auszuwählen ist: pragmatisch, semantisch, syntaktisch (Reihenfolge I), oder umgekehrt: syntaktisch, semantisch, pragmatisch (Reihenfolge II).

Die Frage, welche der Reihenfolgen auszuwählen ist, spaltet die Geister, nicht nur in der Informatik. Die Anhänger von Reihenfolge I nennen sich Pragmatiker. „Sprechen ist Handeln" (vom Griechischen πραξιζ, praxis) ist die Hauptthese eines pragmatischen Vorgehens. Wenn wir sprechen, erzeugen wir handelnd Ausprägungen eines Schemas. Aus Handlungen lassen sich Bedeutungen ableiten. „Die Bedeutung eines Wortes ist sein Gebrauch in der Sprache" lautet ein berühmter Satz des für die Informatik so wichtigen Philosophen Wittgenstein (1889–1951). Der Satz ist in seinen philosophischen Untersuchungen unter § 43 zu finden. Wittgenstein ist der Mann, der eine sprachpragmatische Wende eingeleitet hat. Seine Wirkung auf alle Wissenschaften, die sich mit Sprache befassen, war enorm. Wenn die aus einer Sprechhandlung erwachsene Bedeutung klar ist, bedarf das Ergebnis in einem weiteren Schritt einer wohlgeformten Darstellung, einer Syntax. Man darf auch von Orthographie sprechen.

Wenn Pragmatiker in dieser Reihenfolge eine Rangordnung der Probleme erblicken, geraten sie mit anderen Geistern in der Informatik in Streit, die die Reihenfolge II favorisieren. Wir nennen sie Entwickler. Bitte schön, werden die Anhänger der Reihenfolge II sagen, hier ist mein Schema, mein Rechnerprogramm. Es läuft noch nicht, es produziert noch keine Ausprägungen, weil noch einige syntaktische Fehler drin sind. Der Rechner nimmt syntaktische Fehler sehr übel und versagt seine Dienste. Aber das ist für die Herrn Pragmatiker nur drittrangig. Ob die Semantik den Spezifikationen, den festgelegten Bedeutungen entspricht, kann ohne ein laufendes Programm nicht festgestellt werden. Das ist aber ja auch nur zweitrangig für die Herrn Pragmatiker.

Pragmatiker und Entwickler leben in einem ironischen Verhältnis, obwohl beide wissen, dass sie voneinander abhängen. Der Pragmatiker erreicht nichts ohne eine syntaktisch und semantisch saubere Entwicklung. Dem Entwickler droht die Belanglosigkeit, wenn es keinen Gebrauch für seine Schemata gibt. Die bekannte Ironie ist fehl am Platze. Sie entstammt nicht der Reihenfolge, sondern der Gewichtung der einzelnen Schritte. Der Satz „the most important part comes first" ist wie bei jeder Zurücklegung eines Weges als Grundsatz falsch. Hochmütige Pragmatiker und hochmütige Entwickler mit Rangordnungen im Kopf sollten die Geschichte vom Lahmen und Blinden lesen, die alleine nichts erreichen, zusammen aber wenigstens etwas auf ihrem gemeinsamen Wege fertig bringen. Die Lehre über Reihenfolgen wird in den Wissenschaften auch Lehre vom Wege oder Methodologie genannt. Richtige und falsche Methodologien gibt es nicht in Wissenschaften, die sich kritisch nennen und die einem Dogmatismus abgeschworen haben. Es gibt nur begründete und unbegründete Methodologien, und jeder Wissenschaftler sollte wissen, dass Hochmut vor dem Fall kommt, den doch jeder gerne vermeiden möchte.

2. Schemaerwerb und Bildung

Mit den historischen und systematischen Bemerkungen im obigen Abschnitt soll auch auf einen wichtigen Zusammenhang beim Erwerb einer Grundbildung und beim Bildungserwerb überhaupt hingewiesen werden. Es ist zwar so, dass die Informatik das Thema „Schema – Ausprägung" zu ihrem zentralen Anliegen gemacht hat, was von anderen Disziplinen nicht behauptet werden kann. Man könnte mit anderen Worten auch sagen: Die Informatik ist sich des großen Themas bewusst, das anderen aus dem Blickfeld geraten ist und nur unscharf am Rande wahrgenommen wird. Wie kommt das? Die Antwort auf diese Frage ist relativ einfach. Wer etwas auf den Rechner bringen will, was für uns lebensbedeutsam ist, der muss unsere Lebenswelt kritisch, häufig sogar ab ovo, rekonstruieren. Der Rückgriff auf so etwas Grundsätzliches wie die Natur des Menschen, was man Anthropologie nennt, ist dabei unerlässlich. Für die Physik und anderen Naturwissenschaften zum Beispiel ist die Rede von Schema und Ausprägung etwas Nebensächliches. „Das weiß man doch" lautet die bekannte Antwort. Im Sprachunterricht der Schulen geht die Lehre von Schema und Ausprägung zum Leidwesen der Informatik wegen der Fülle des Stoffes in vielen Fällen einfach unter. Zwar lernt der Schüler im Englischunterricht brav das Schema „to suffer from something“ (an etwas leiden) als infinite Verbform. Er lernt auch, dass „I suffered from measles“ (Ich litt an Masern) eine korrekt gebildete Ausprägung ist. Dass aber eine ganze moderne Wissenschaft, die wir *Inschematik* nannten, von dem, was da im Sprachunterricht bloß als Exempel vorgeführt wird, lebt, das erfährt der Schüler nicht. Wer erfährt schon im Deutschunterricht, dass der Satz „Ich gratuliere Dir zum 15. Geburtstag“ nur etwas Singuläres ist gegenüber dem allgemeinen Schema „Jemand gratuliert jemandem zu etwas“. Oder schauen wir auf den Mathematikunterricht. Zur Debatte mag der Satz stehen: Für alle x: Wenn $x>1$, gilt auch $x^2>1$. Im allgemeinen sagt der Lehrer schon, dass man zur Erleichterung auch den großen Buchstaben X einführen darf, um zu schreiben: Wenn $X>1$, gilt auch $X^2>1$, wobei er hinzufügt, dass X als fest, aber beliebig anzusehen ist. Hoffentlich sagt der Lehrer auch, das X ein schematischer Buchstabe ist, der nicht quantifiziert werden darf und keine Variable im üblichen Sinne ist. Hoffentlich verweist der Lehrer auf das Fach *Inschematik* mit dem Hinweis, dass man dort über Schematisierungen eine Menge mehr lernen kann, was für das spätere Leben von sehr, sehr großer Bedeutung ist. Unter glücklichen Umständen sagt der Lehrer sogar, wenn man über Eigenschaften von Schemata redet, wie etwa bei *X>1* oder bei *„to suffer from“ ist zwölf-buchstabig*, dass man dann auf die Ebene einer Metasprache gerät, über die man nicht nur im Fach Inschematik sehr viel lernen kann. Auch im Deutschunterricht sollte eine Regel wie „Alle Subtantiva werden am Anfang groß geschrieben“ (für alle S gilt, S ist großbuchstabig) als Satz einer Metasprache aufgefasst werden. Man nennt ein solches Regelwerk im allgemeinen Grammatik. Den Satz *“schön“ ist ein Adjektiv*, mag ein Grammatiklehrer an die Tafel schreiben. Hoffentlich vergisst er dabei nicht die Anführungszeichen, die aus einem Wort der deutschen Objektsprache („use language“, Gebrauchssprache) den Namen eines Satzes in einer Metasprache („mention language“, Erwähnungssprache) machen. Ohne Anführungszeichen, wie z.B. beim Satz: *Schön ist das Mädchen,* schriebe der Grammatiklehrer etwas höchst Zweifelhaftes an die Tafel, weil dann nicht herausgestellt würde, dass das Wort *„schön“* nur als ein Sprachkonstrukt, nur als ein Schema auch ein Adjektiv ist.

Was ist Bildung? Man kann zu diesem Thema viele geistreiche Bücher finden, die ganze Bibliotheken füllen. Wenn Bildung aber nicht mehr unbedingt etwas Sittliches bewirken soll, wie noch bei J.G. Fichte im Jahre 1808, sondern nach PISA heute „funktionale Kompetenz der Welterschließung und Problembewältigung “ (Terhart) zum Gegenstand hat, und – wie im Deutschen üblich – dann als Anglizismus unter dem Wort *literacy* firmiert (man könnte im Deutschen auch statt *toll collect Maut-* bzw. englisch-näher *Zolleinzug* sagen), ja dann sollte es auch erlaubt sein, auf die anthropologischen Konzepte „ Schema – Ausprägung“ zurückgreifen zu dürfen, wobei in unserer scheinbar anglophilen Welt ohne Bedenken und Einschränkungen auch die Wörter „type – token“ (seit Peirce) oder „competence – performance“ (seit Chomsky) verwendet werden sollten. Gemeint ist immer der universale bzw. der singuläre Aspekt in Bezug auf einen Gegenstand. Ein Hauptanliegen des ersten Teils unserer sechsteiligen Darstellung über „Informatik als Grundbildung“ ist eine Fokussierung auf die Aussage: Bildung, das sind erworbene Schemata, plus die Fähigkeit zum Selbsterwerb (Fichte), plus Gedächtnisleistung. Gebildete Menschen haben durchweg ein gutes Langzeit-Gedächtnis, weshalb das Auswendiglernen von Gedichten eine vorzügliche Übung ist, was die heutigen Pädagogik vergessen zu haben scheint. Ohne ein Selbsttätigwerden reduziert sich Bildung auf Ausbildung („education“). Bildung ist ein Selbstläufer, wenn eine Orientierung gegeben ist. Ausbildung bedarf einer permanenten Auffrischung in Kursen mit Credit points als Belohnung in abfragbaren Einheiten. Man sagt im Deutschen „Ich bilde mich“ (Aktivform) und „Ich werde ausgebildet“ (Passivform). Man sagt nicht „Ich werde gebildet“ (Passivform) und „Ich bilde mich aus“ (Aktivform). Bildung wie Ausbildung haben aber das Verfügen über Schemata und den Schemaerwerb gemein. Um einen höheren Bildungs- bzw. Ausbildungsstand zu erreichen, ist der Besuch von Schulen, Hochschulen und Universitäten unerlässlich. Lehrer werden wissenschaftlich ausgebildet, um Schemata und nicht Ausprägungen zu vermitteln. Schemata befähigen, Ausprägungen zu verstehen, nicht umgekehrt, obwohl in Lehr- und Lernsituationen häufig das Erlernen von Schemata durch ein Vorführen von Ausprägungen eingeübt werden kann. Schreiben und nicht ausgeprägtes, schönes Schreiben (Kalligraphie) wie in der zum großen Teil schemafreien Kunst stehen zur Debatte. Es gibt, frei nach Kant, keine schöne Wissenschaft, es gibt nur schöne Kunst.

Es ist ein großer Fehler zu glauben, dass ein Schemaerwerb nur in einem institutionellen Rahmen von Schulen stattfinden kann. Lehr- und Lernsituationen sind in unserem Leben dominant und mindestens so bedeutsam wie Entscheidungs-, Muße-, Transport- oder Herstellsituationen. Nehmen wir als Beispiel einen Vater, der seinem Sohn das Schwimmen beibringen will [2]. Der Vater demonstriert Schwimmbewegungen als Ausprägungen im Trockenen, indem er Armzüge mit gleichzeitigem Beingrätschen vorführt. Der Sohn imitiert das Schema ebenfalls als Ausprägung. Ob er die Ausprägungen als Schema gelernt hat, versucht er dann mit einiger Überwindung im Nassen. Beispiele für nicht institutionalisierte Lehr- und Lernsituationen aus der Welt der Sprache sind in gebotener Kürze schwieriger vorzutragen. Nehmen wir einen Ortsunkundigen an, der Fußgänger nach einer bestimmten Straße im Ort befragt. Der Fußgänger beschreibt das Schema des Wegs zum gesuchten Ziel, nicht mehr. Vorführen wäre eine besondere Gunst, die der Ortsunkundige nicht erwarten kann. Der Ortsunkundige ist auf sich selbst gestellt. Ob er sein Ziel erreicht, ist fraglich. Deshalb ist der Erfolg eines Schema-Erlernens im Bereich der Sprache immer auch ein Problem der Kontrolle. Viel komplizierter als das Wegesuchen in einer fremden Stadt sind z.B. das Erlernen des Schemas eines Revisionspfades („audit trail“) einer Debitorenbuchhaltung oder das in den Griff bekommen des Schemas zum Justieren einer NC-Maschine.

Auf jeden Falle spielt eine Schemaerwerbs-Fähigkeit („schema acquisition capability“) der Menschen eine zentrale Rolle.

3. Schemaerwerb und Gerechtigkeit

Schemaerwerb im Bildungs- und Ausbildungsprozess erfordert je nach Art und Schwierigkeit unterschiedliche Fähigkeiten und ein vielfältiges Können. Was den Abruf menschlicher Fähigkeiten anbetrifft, gilt seit alters her der alte römische Rechtsgrundsatz: „Ultra posse nemo obligatur“ oder „Über sein Können hinaus ist niemand verpflichtet“. Wenn ein Nicht-Können vorliegt, dann ist auch ein Nicht-Sollen gegeben. Schematisch kann geschrieben werden: $\neg A \rightarrow \neg B$ (1), wobei der Haken ($\neg$) die Verneinung und die schematischen Buchstaben A und B durch Können bzw. Sollen zu aktualisieren sind. Dieses römische Gerechtigkeitsprinzip gilt auch heute nach bald 2000 Jahren noch in der westlichen Welt, obwohl es sozialpolitisch gerne durch ein zumutbares Können oder psychologisch durch ein Können-Mögen verwässert wird. Logik ist viel ernster, als Sozialpolitiker oder Psychologen das vermuten. Die Ernsthaftigkeit des römischen Gerechtigkeitssatzes wird z.B. auch herausgestellt, wenn wir mit dem großen Philosophen I. Kant (1724–1804) den Umkehrschluss (man sagt in der Logik auch „Kontraposition“) auf (1) anwenden. Klassisch logisch gilt:

Aus $\neg A \rightarrow \neg B$ (1) folgt $B \rightarrow A$ (2) . „Wenn Du sollst, dann kannst Du auch“(2) wird aus (1) gefolgert. Kant formulierte „... der Mensch kann etwas, weil ihm bewusst ist, dass er es soll“ (Kritik der Praktischen Vernunft, A 54).

Wir wollen im Folgenden (1) das Sozialprinzip und (2) das Leistungsprinzip beim Schemaerwerb nennen. Es ist klar, dass nach (1) etwa Behinderte, die schon allein nach dem Augenschein nicht bildungsfähig sind, nicht verpflichtet werden können. Das ist aber ein problemloser Fall. Problematisch wird die Situation, wenn keine Evidenz, sondern ein Feststellungsverfahren in Gang gesetzt werden muss, um auf Nicht-Sollen zu erkennen. Wer trägt die Beweislast des Nicht-Könnens? Z.B. der Staat, der das Prinzip (1) vertritt, oder die Person, die in Frage steht. Wir werden im Teil VI unserer Darstellung „Logik als Geltungssicherung von Behauptungen“ erfahren, dass auch die Beweislastfrage logisch einwandfrei geklärt ist. In der dialogischen Logik lernt man, dass der Proponent, in unserem Fall der Staat, zu nichts verpflichtet wird, wenn der Opponent, in unserem Fall die betroffene Person, die Behauptung des Nicht-Könnens nicht nachweist. Einfach nur zu sagen: „Ich kann nicht“ reicht nicht aus. Eine Begründung der Behauptung muss her, damit ein „Nicht-Können“ Geltung erlangt. Man erkennt: Bei der Frage der Beweislast „hört im allgemeinen der Spass auf“ und es wird ernst, logisch ernst.

Das Leistungsprinzip (2), das in gleicher Weise dialogisch logisch behandelt werden kann, ist wegen der positiven (affirmativen) Formulierung für jedermann einfacher zu verstehen. „Wenn etwas verlangt wird, dann muss das auch erreichbar sein“, ansonsten liegt ein unbilliges Verlangen vor. Äußerst problematisch bei (2) aber ist sein Er-schließen aus (1) mit Hilfe des Umkehrschlusses. Kant galt und gilt heute noch als „bewunderter Rigorist". Er glaubte vor mehr als 200 Jahren noch, dass die Logik ein abgeschlossenes Fach und nicht mehr entwicklungsfähig sei. Da hatte sich der große Mann, der körperlich nur 1,59 m groß gewesen sein soll, aber gewaltig geirrt, wie das bei bedeutenden Leuten auch zuweilen vorkommt. Die gewaltige Entwicklung der Logik, die mit G. Frege (1848–1925) 100 Jahre später einsetzte, konnte Kant nicht vorausahnen. Hätte er es, dann wäre diesem kritischen Mann sofort klar geworden, dass sein Umkehrschluss so ohne weiteres nicht gilt. Der Schluss: Aus $\neg A \rightarrow \neg B$ folgt $B \rightarrow A$ gilt nämlich konstruktiv-logisch nicht. Wir werden im Teil VI noch sehen, dass die klassische Logik, die Kant noch benutzte, den berühmten Satz vom ausgeschlossenen Dritten (tertium non datur) als Annahme zugrunde legt. Wird dieser Satz aufgegeben, dann gilt der Umkehrschluss mit dem Schema, das Kant benutzte, nicht. Wenn wir den Satz vom ausgeschlossenen Dritten, also $A \vee \neg A$, erweitern und auch noch den häufig lebensweltlich anzutreffenden Fall des „es ist noch nicht entschieden, ob $A \vee \neg A$ gilt", die Römer sagten dazu *non liquet* (es fließt noch nicht) zulassen, dann entsteht eine viel allgemeinere Logik, die konstruktiv (intuitionistisch sagt man auch) genannt wird. Und dass etwas noch nicht fließt, das kommt häufig vor, insbesondere, wenn man an menschliche Arbeit denkt. Unser Problem ist, dass der Satz „Können $\vee \neg$ Können ein Drittes gibt es nicht" eine grobe und unzulässige Vereinfachung unserer Lebenswelt ist. Häufig, gerade im Ausbildungsbereich, ist ein Können noch im Werden, es ist noch nicht vorhanden, aber es mag eine Hoffnung bestehen, dass es bald einsetzt. Wenn das so ist, gilt der Umkehrschluss nicht, was Kant noch nicht wusste. Hochinteressant ist natürlich die Tatsache dass die Umkehrung, also von (2) nach (1), auch konstruktiv gilt. Aus (2) kann (1) konstruktiv wie klassisch erschlossen werden. Konstruktiv-logisch gesprochen ist (2) stärker als (1). Aus diesem Grunde könnte der Satz „Du kannst etwas, weil Dir bewusst ist, dass Du es sollst" im Emblem einer jeden Schule und Hochschule stehen.

Mit diesem abschließenden Eingehen auf „Logik und Geltungssicherung von Behauptungen" wird zum Abschluss des Teils I, der als Öffnung einer Klammer aufzufassen ist, auf das Schließen dieser Klammer mit Teil VI verwiesen. Zuvor haben wir uns aber im nächsten Beitrag „Bildung von Elementarsätzen" mit dem wichtigen Thema „Was ist eine rationale Grammatik?" auseinander zu setzen. Eine empirische Grammatik wird im Sprachunterricht schon sehr früh eingeführt, z.B. wenn gelernt wird, dass zwischen Haupt- und Beiwörtern zu unterscheiden ist.

Literatur

1. Deutsches PISA-Konsortium (Hrsg.): PISA 2000 – Basiskompetenzen von Schülerinnen und Schülern im internationalen Vergleich. Opladen: Leske + Budrich 2001
2. Kamlah, W., Lorenzen, P.: Logische Propädeutik, Vorschule des vernünftigen Redens. 3. Aufl. Stuttgart: J. B. Metzler 1996
3. Janich, P.: Logisch-pragmatische Propädeutik. Ein Grundkurs im philosophischen Reflektieren. Weilerswist: Velbrück Wissenschaft 2001
4. Lorenz, K.: Einführung in die philosophische Anthropologie. Darmstadt: Wissenschaftliche Buchgesellschaft 1990
5. Rechenberg, P., Pomberger G. (Hrsg.): Informatik Handbuch, 3. Aufl. München: Hanser 2002
6. Terhart, E.: Nach PISA – Bildungsqualität entwickeln. Hamburg: Europäische Verlagsanstalt 2002

Informatik als Grundbildung

Teil II: Bildung von Elementarsätzen

H. Wedekind · E. Ortner · R. Inhetveen

Informatik als Grundbildung wird als eine methodische Vorbereitung der Schüler auf einen Informatikunterricht an allgemeinbildenden Schulen neben eine sprachliche, mathematische und naturwissenschaftliche Grundbildung gestellt [5]. Das Gebiet wird in sechs Teilen vorgetragen. Teil I behandelte das Thema „Schema und Ausprägung". Der vorliegende Teil II ist Elementarsätzen einer Rationalen Grammatik gewidmet. Der als nächstes folgende Teil III „Gleichheit und Abstraktion" stellt das Abstraktionsprinzip in den Mittelpunkt der Betrachtungen.

Rationale Grammatik

Grammatiken

Schon früh wird im schulischen Sprachunterricht der Begriff *Grammatik* eingeführt. Man spricht dann über Sprache, um die eigene Muttersprache oder die jeweiligen Fremdsprachen besser zu verstehen. Man lernt Wortarten wie Haupt-, Bei- und Tuwort und den Aufbau von Sätzen, etwa die Gliederung in Haupt- und Nebensätze. Es wird unter den Sprachen eine große Unterschiedlichkeit festgestellt, weil eben die so genannten *natürlichen Sprachen* verschiedenen Kulturkreisen entstammen. Zu selten wird im Unterricht die Frage nach den Gemeinsamkeiten der Sprachen gestellt. Wenn der Satzaufbau zur Debatte steht, beschränkt man sich auf die Feststellung, dass alle indoeuropäischen Sprachen eine *Subjekt-Prädikat-Objekt-Gliederung* aufweisen. Die große Frage aber, ob es eine Sprache gibt, die über allen empirischen bzw. natürlichen Sprachen steht und somit transkulturell genannt werden kann, bleibt in einem Unterricht ohne Informatik als Grundbildung in der Regel unbeantwortet. Die Antwort auf diese wichtige Frage lautet: Ja, es gibt eine solche Kunstsprache, deren Grammatik *Rationale Grammatik* genannt wird. Der berühmte Wittgenstein, den wir im Teil I zweimal zitierten, sprach noch von einer Logischen Grammatik (Tractatus 3.325). Ein Riesenvorteil einer solchen Grammatik ist, dass sie die Grundlage für alle Sprachen ist, die unsere Rechner *verstehen*. Eine Rationale Grammatik ist ein Universalschema, das wegen seiner Allgemeingültigkeit viel einfacher sein muss als empirische Grammatiken mit ihren Aktiv-Passiv-Formen, komplizierten Flexionen, Indikativ-Konjunktiv-Formen in Gegenwart, Vergangenheit und Zukunft, Unregelmäßigkeiten en masse, äußerst komplexen Satzstrukturen etc. Es ist auch sehr einfach zu erkennen, ob ein Schema zu einem Rationalen Schema gehört oder nicht. Wenn jemand argumentieren kann, dass ein Sprachschema nur in diesen oder jenen empirischen Sprachen vorkommt, wobei z.B. die Sprachen der Hopi-Indianer und das Chinesische selbstverständlich eingeschlossen sind, dann hat das diskutierte Schema seine Kandidatur um Aufnahme in den Bereich einer Rationalen Grammatik verwirkt. Menschen können sich in einer rationalen Sprache nicht unterhalten, dazu ist die Sprache viel zu primitiv. Menschen können aber mit Hilfe einer rationalen Sprache eine methodische Rekonstruktion unserer

DOI 10.1007/s00287-004-0393-x

H. Wedekind · R. Inhetveen
Universität Erlangen-Nürnberg,
Fachbereich Informatik,
Deutschland
E-Mail: wedekind@informatik.uni-erlangen.de

E. Ortner
TU Darmstadt,
Fachgebiet Wirtschaftsinformatik- Entwicklung
von Anwendungssystemen,
Deutschland

gesprochenen Sprache vornehmen. Und dieses Schematisieren geht einem Automatisieren – einem Hauptanliegen der Informatik – voran.

Eine Rationale Grammatik hat, wie jede empirische Grammatik auch, eine Wort- und Satzlehre. Ein *Satz*, der aus *Wörtern* besteht, ist aus dem Aspekt der Hörer und Sprecher (also kommunikativ) ein vollständiger sprachlicher Ausdruck. Ein sprachlicher Ausdruck wiederum ist jedes zum Ausdruckbringen einer sprachlichen Handlung. In Anlehnung an den Teil I „Schema und Ausprägung" können wir auch sagen, ein Ausdruck ist die Aktualisierung, eine Ausprägung eines Schemas. Das Heben der Kopfbedeckung ist ebenfalls ein Ausdruck. Diese Handlung, die z.B. einen Gruß ausdrücken soll, gehört aber nicht zu den sprachlichen Handlungen, um die es uns hier geht. Ein Satz im Sinne einer Rationalen Grammatik heißt elementar, wenn er nicht zusammengesetzt ist. Zusammengesetzt werden elementare Sätze zu einem komplexen Satz mit Hilfe von logischen Partikeln wie UND ($\wedge$), ODER ($\vee$), Wenn... Dann... ($\rightarrow$). Man nennt diese verbindenden Operatoren auch Junktoren („connectives"). Ihre Behandlung steht im Teil VI „Logik und Geltungssicherung von Behauptungen" zur Debatte. Wir konzentrieren uns in diesem Teil nur auf Elementarsätze.

Aufforderungen und Aussagen

Besonders einschränkend empfindet es jeder sprachbegabte Mensch, dass eine Rationale Grammatik nur zwei Ausdrucksschemata zur Verfügung stellt: Aufforderung („request") und Aussage („proposition"). Für eine Aufforderung stellt eine natürliche Sprache eine Fülle von Möglichkeiten bereit, die rational alle auf ein Auffordern verdichtet werden können. Z.B. im Deutschen: verlangen, bitten, befehlen, anweisen etc. und – zur Überraschung – auch fragen. In der Tat entpuppt sich eine Frage als eine Aufforderung. Betrachten wir den englischen Satz: „Passengers are kindly asked to leave the ship", so wird deutlich, dass dies eine besonders höfliche Aufforderung ist. Man könnte auch ruppig sagen: „Passengers, get off the ship". Im Sinne einer logischen Rekonstruktion sind beides Aufforderungen. Höflichkeit und Anstand sind nicht Sache einer logischen Rekonstruktion, obwohl sie lebenswichtig sein mögen. Was die sprachliche Fassung einer Aufforderung betrifft, so haben sich Informatiker als Programmierer an das Wort „Anweisung" gewöhnt. Manchmal sagen sie auch *Instruktion*, um das englische „instruction" nachzuahmen. Auch in dem beliebten Informatiker-Satz „I call a subroutine at that point" ist „to call" als eine Aufforderung zu rekonstruieren. Ein Wörterbuch sagt, „to call" ist neben *rufen* in einer Zweitbedeutung gleichzusetzen mit „to summon" (auffordern).

Auf der Seite der Aussagen ist in natürlichen Sprachen eine ebenso große Vielfalt zu konstatieren: feststellen, definieren, beschreiben, erklären, bestreiten, darlegen etc. Informatiker reden gerne vom Definitionsteil oder auch vom Deklarationsteil ihrer Programme, um eine Unterscheidung zum Anweisungsteil vorzunehmen.

Beide Schemata, für ein Auffordern und für eine Aussage, sollen nun im Folgenden auch konkret hingeschrieben werden. Dabei beschränken wir uns – wie es die Überschrift unseres Beitrags verlangt – auf Elementarsätze. Wir können das natürlich nur tun, indem wir Wörter benutzen. Soll dies reflektiert geschehen, so ist zuvor eine Wortlehre bereitzustellen. In einem zu betrachtenden Elementarsatz unterscheidet man in einer Rationalen Grammatik zwei Teilschemata. Ein Teilschema nennt man das Schema des Namens. Man sagt auch Eigenname („proper name"), wenn in einer Aussage ein Gegenstand benannt wird, gelehrt sagt man statt *benennen* auch *referenzieren* (Bezug nehmen). In „Fido ist ein Hund" ist Fido ein Eigenname. Namen können aber auch als Gattungsnamen („generic name") auftreten, z.B. in „Hunde sind vierbeinig". In beiden Fällen wählen wir den schematischen Buchstaben N und sagen dazu einheitlich *Nominator*, ein Wort der Rationalen Grammatik. Einem benannten Gegenstand wird nun in einer Elementaraussage innerhalb eines Elementarsatzes eine Eigenschaft zugesprochen. Für die zugeordnete Eigenschaft gibt es das Wort *Prädikator*. Beide Termini, *Nominator* wie *Prädikator*, wurden vom bedeutenden Logiker Rudolf Carnap (1891–1970) in die Rationale Grammatik eingeführt. Er sagte auch bewusst *Prädikator*, um eine Unterscheidung zum Prädikat einer empirischen Grammatik aufzuzeigen. Für Prädikatoren nehmen wir den schematischen Buchstaben P. In dem elementaren Aussagesatz „Fido ist ein Hund" stellt das deutsche Wörtchen *ist* bei der logischen Rekonstruktion ein zunächst unerwartetes, großes Problem dar, weil es – wie noch zu zeigen ist – fast beliebig mehrdeutig verwendet wird. Mehrdeutigkeiten sind aus ersichtlichen Gründen für logische Rekonstruktionen ein Feind, der bekämpft werden muss. Wenn man aber sorgfältig vorgeht, wie wir das tun wollen, kann nichts passieren. Mit Hilfe des Wörtchens *ist* wird – so sagt man – einem benannten Gegenstand eine Eigenschaft zugesprochen. Für dieses Zusprechen, als eine Handlung begriffen, sagt man auch *Prädikation* und wählt den Buchstaben ε aus, um sich von dem deutschen Wörtchen *ist* zu lösen. Man schreibt nun für „Fido ist ein Hund" als einen elementaren Aussagesatz auch „Fido ε Hund" oder das Schema $N \varepsilon P$. Das Zeichen ‚ε' heißt Kopula, weil es den nominativen mit dem prädikativen Teil verbindet. Der Logiker Gottlob Frege (1848–1925) hat sich um die Präzisierung der Prädikationshandlung große Verdienste erworben. Wenn ein Satz $N \varepsilon P$ nicht nur ausgesagt, sondern auch behauptet wird, schreibt man $\vdash N \varepsilon P$ und nennt ihm zu Ehren das Zeichen $\vdash$ das Fregesche Behauptungszeichen. Behaupten („to assert") ist schwieriger als aussagen, mehr nicht. Wer behauptet, der verkündet seinen Mitmenschen, dass er Belege, Gründe oder Beweise hat – wie immer man eine Geltungssicherung auch nennen mag – um seine Aussage bestätigt zu bekommen. Das kann ein schweres Geschäft sein, weshalb man mit dem Behaupten vorsichtig sein sollte. Im Teil VI „Logik als Geltungssicherung von Behauptungen" wird dieses Geschäft behandelt. Bevor wir aber so weit sind, sollte noch die logisch wichtige Negation als Handlung vorgestellt werden. Der Satz „Fido ist kein Mensch" wird logisch rekonstruiert zu $N \varepsilon'$, P, mit ε' als Zeichen für ein Absprechen eines Prädikators. Äquivalent dazu ist eine Formulierung mit dem Zusprechen eines Negativ-Prädikators zu sehen. Beispiel: „Fido ist ein Nicht-Mensch". Beide Sätze „Fido ε' Mensch" und „Fido ε Nicht-Mensch" sind syntaktisch verschieden, semantisch aber gleichbedeutend. Man sagt auch gelehrt, beide Sätze sind synonym vom Inhalt oder von der Intension her. Synonymität (Gleichbedeutendheit) ist eine wichtige Gleichheitsbeziehung, man sagt auch Äquivalenzrelation, die im Teil III „Gleichheit und Abstraktion" eine wesentliche Rolle spielt.

Obwohl in der Informatik Aufforderungen als Anweisungen eine ebenso große Rolle spielen wie Aussagen als Behauptungen, wollen wir uns bei der logischen Rekonstruktion von elementaren Aufforderungen kurz fassen. Selbstverständlich sind „Hol Wasser im Eimer", wenn ein Haus brennt und gelöscht werden soll, und „i=i+1" in einem Programm, wenn ein Index um eins zu erhöhen ist, um eine Sortierfolge herzustellen, Aufforderungen. In den Wissenschaften wird eine Aufforderung, in der ein Zweck in der Aufforderung gleich mitgeteilt wird, eine finale Aufforderung genannt (z.B.: „Lösche das brennende Haus". Mit welchen Mitteln das gemacht wird, ist eine andere Angelegenheit). Bleibt der Zweck außen vor wie in den obigen Beispielen, spricht man von afinalen oder schlichten Aufforderungen [3]. Wir vergegenwärtigen uns, dass in unserem Bereich durch eine Aufforderung eine Handlung veranlasst werden soll, die ein Mittel ist, um einen Sachverhalt, einen Zweck zu erzielen. Eine (konkrete) Aussage wiederum stellt einen (abstrakten) Sachverhalt („fact") dar, was im Teil III noch zu behandeln ist. Wenn das so ist, dann kann man auch eine Aussage benutzen, um eine Aufforderung schematisch zu behandeln. Mit anderen Worten: Wir formulieren alle Aufforderungen final. Das syntaktische Zeichen für eine Aufforderung ist das Ausrufungszeichen (!), das wie das Behauptungszeichen ($\vdash$) anzusehen ist und vorangestellt wird. Die Rekonstruktionen

> "! E ε herbeigeholtes_Wasser_im _Eimer" und
> "! I ε um_eins_erhöhter_Index"

weisen ein gemeinsames Aufforderungsschema auf, nämlich: ! N ε P.

In Worten: Bitte mach (!), dass der Sachverhalt N ε P gilt. Das Machen muss in der Informatik programmiert werden und „fällt nur vom Himmel", wenn ein anderer das Machen schon für uns programmiert hat. Der Adressat der Aufforderung ist kein Mensch, sondern ein Rechner. Das ist übrigens der Witz der ganzen Informatik, mehr nicht. Dieses Mehr kann aber sehr viel sein und sollte von Außenstehenden niemals unterschätzt werden, was der Fall ist, wenn man die reden hört, die da sagen, „das muss ja nur noch programmiert werden". Die beste Empfehlung, als Aufforderung verstanden, für solche Redner ist: !Bitte vormachen.

Apprädikatoren

Im Gegensatz zu empirischen Grammatiken, die kompliziert und geschlossen sind, weil eine Vielzahl von Menschen sich ihrer Schemata unbekümmert bedient, ist eine rationale Grammatik primitiv, aber sie ist offen. Ohne Schwierigkeiten ist es möglich, eine Rationale Grammatik um das Operieren mit Gedanken, man nennt das Logik, aber auch um das Operieren mit Zahlen, das ist die Arithmetik, zu erweitern. Für den Schulunterricht ist die Offenheit bzw. Erweiterbarkeit ein wichtiges Phänomen, weil man mit dem Aufbau jederzeit aufhören kann, und das auch tut, wenn man z.B. die Logik bedauerlicherweise außen vor lässt. Bei empirischen Grammatiken geht das nicht, es sei denn man argumentiert für Bescheidene und sagt z.B., dass der Konjunktiv nicht mehr gelehrt werde, weil er im Fernsehen ungebräuchlich sei.

Apprädikatoren können zwanglos eingeführt werden, weil sie den Adjektiven (Beiwörtern) einer empirischen Grammatik entsprechen. In dem erweiterten Satz „Fido ist ein brauner Hund" ist *braun* ein Apprädikator. Apprädikatoren sollten nicht in gewöhnliche Prädikatoren umgewandelt werden, wie das die Logik des logischen Atomismus tut, wenn geschrieben wird: „Fido ist braun UND ($\wedge$) Fido ist ein Hund". Aus diesem Satz könnte man auch den Satz „Fido ist ein hündisches Braun" herauslesen. Es ist aber nicht beabsichtigt, *Hund* mit dem Adjektiv *hündisch* apprädikativ zu stellen.

Eine beliebige Zahl von Apprädikatoren können einem Eigen- oder Gattungsprädikator hinzugefügt werden. Die Sätze „Fido ist ein brauner, teurer, weiblicher Hund" bzw. „Alex ist ein schwarzer, billiger, männlicher Hund" sind rational-grammatisch korrekt gebildet. Ihr allgemeines Schema lautet: $N\ \varepsilon\ (A_1, A_2, A_3)\ P$. Das besondere Schema im Kontext von Hunden kann wie folgt beschrieben werden: Name ε (Farbe, Wert, Geschlecht) Hund. Einem Informatiker wird an dieser Stelle sofort klar, dass hier das berühmte Relationen-Modell der Datenbanksysteme als Schema hingeschrieben wird. Es ist nur eine Umstellung der schematischen Buchstaben erforderlich. Ein relationales Schema notiert man wie folgt: $P\ (N, A_1, A_2, A_3)$, oder in unserem speziellen Fall: Hund (Name, Farbe, Wert, Geschlecht). P nennt man relational den Relationen-Namen. N heißt Primärschlüssel. Die Apprädikatoren A_i werden Sekundärschlüssel genannt. Man sagt *Schlüssel*, weil an einen Zugriff auf Daten in einem technischen Sinne gedacht ist. In einer Informatik als Grundbildung stehen Fragen des Zugriffs nicht zur Debatte. Viel wichtiger ist, aus diesem Aspekt zu erkennen, dass der Entwerfer von Datenbanksystemen in Elementarsätzen einer *Rationalen Grammatik* denkt.

Wichtig im Unterricht ist auch, dass Schemata dieser Art zwanglos und möglichst einvernehmlich eingeführt werden. Es ist in einer Informatik als Grundbildung völlig inadäquat, eine Relation – wie üblich – als Untermenge eines Kreuzproduktes zu definieren, um zu suggerieren, die Ausprägungen oder Tupeln dieser Relation könnten in wunderbarerweise durch eine „Kreuzproduktmaschine" erzeugt werden. Nicht extensionales, mengenorientiertes Denken, das in der Informatik überreichlich vorhanden ist, sondern Denken in Sprechhandlungen und deren Schemata in einer intensionalen, inhaltlichen Deutung stehen methodisch am Anfang, was im zweiten Abschnitt noch genauer auszuführen ist. Die Tupeln oder Ausprägungen <<u>Fido</u>, braun, teuer, weiblich> und <<u>Alex</u>, schwarz, billig, männlich> sind Elementarsätze, in denen syntaktisch nur die Kopula fehlt. Dass Fido und Alex Nominatoren sind, wird aber in der Notation der relationalen Datenbanken häufig durch ein Unterstreichen kenntlich gemacht, auch schon im Schema, z.B. durch Hund (<u>Name</u>, ...). Es ist wichtig zu vermerken, dass in relationalen Systemen komplexe, logisch beliebig verknüpfte Sätze erst in Fragen an eine Datenbank auftreten. Man nennt solche Aufforderungen auch im Deutschen „Query". „! Gib mir bitte die Namen der Hunde, die braun ODER ($\vee$) schwarz sind" könnte eine Frage an unsere Mini-Datenbank sein. Als Antwort würden uns Fido, Alex und andere mehr, falls vorhanden, präsentiert. Es heißt im Informatik-Jargon, in Datenbanken sei Information gespeichert. Wenn das so ist, welche Einheit kann man anbieten, wie das bei „anständigen" Grundbegriffen der Fall ist? Stoffe haben eine Masseeinheit, Kilogramm genannt, und Energie misst man in der Einheit Joule. Klar ist, dass Information sprachlich verfasst ist. Als Einheit bietet sich mit Blick auf relationale Datenbanksysteme ein irgendwie zu normierender Elementarsatz an. Auf jeden Fall darf man im Unterricht über Informatik als Grundbildung den Informationsbegriff nicht defätistisch ausschließen, wie das heute üblich ist.

Es sei nur am Rande vermerkt: Auch die hochmoderne Sprache XML kennt wie das Relationen-Modell zwei Arten von Prädikatoren. Attribute nennt man hier die Apprädikatoren. Elemente heißen im XML-Jargon die Eigen- bzw. Gattungsprädikatoren. Man kann in XML also auch zwischen einem *braunen Hund* und einem *hündischen Braun* unterscheiden. Mit der Darstellung einer Rationalen Grammatik gewinnt man somit auch mühelos Zugang zu modernen Austauschformaten wie XML, auch schon in der Grundbildung, was von einer Grundbildung von Format auch verlangt werden kann.

Wir haben bisher nur Ding-Prädikatoren behandelt. Rational-grammatisch stehen die Ding-Prädikatoren den Geschehnis-Prädikatoren gegenüber, die sich auch im Unterricht ganz zwanglos einführen lassen, da ihnen empirisch-grammatisch die Verben (Tuwörter) entsprechen. Geschehnisse („occurrences“) können gegliedert werden in Vorgänge („processes“), Handlungen („actions“) und Widerfahrnisse („to happen“, eine Verbal-Substantivierung *happening* ist aus ersichtlichen Gründen unbrauchbar). Vorgänge oder Prozesse (der Informatikbegriff *Prozedur* hat die gleiche Wurzel) sind neutral als Abläufe zu verstehen. Z.B.: „Ein Stein rollt vom Berg“ ist eine Beschreibung eines Vorgangs. Hingegen sagt man, dass Handlungen von Menschen absichtsvoll ausgeführt werden, was deutungsbedürftig ist. Ein Mann geht zu Boden, weil er sich schützen will, ist etwas ganz anderes, als wenn ein Mann bei Glatteis zu Boden geht. Letzteres widerfährt ihm („it happens to him“). Man spricht auch vom Verhalten („behaviour“) und sagt, der Mann verhält sich fallend (Verhalten), er tut nicht fallen (Handlung). In der Informatik steht durchweg die neutrale Version als Prozess oder Prozedur zur Debatte. Deutungen sind im Allgemeinen nicht die Sache der Informatiker, weil man dann sehr schnell in hermeneutische (*auslegerische*) Gefilde der Geisteswissenschaften gelangen würde. Dass man objektorientiert statt *Prozedur* auch *Methode* sagt, wird wegen der tiefen Bedeutung dieses Wortes als sehr unglücklich empfunden.

Geschehnis-Prädikatoren mit ihren Apprädikatoren werden schematisch genauso wie Ding-Prädikatoren behandelt. Wenn wir den folgenden Elementarsatz „Fido läuft lange und schnell“ schematisiert wiedergeben wollen, schreiben wir analog „Fido ε (lang, schnell) laufen“ oder allgemein „N ε (B_1, B_2) G“. B_1 und B_2 heißen Geschehnis-Apprädikatoren. Sie entsprechen den Adverbien (Umstandswörtern) einer empirischen Grammatik. Wenn unser Alex, den wir oben schon dinglich charakterisiert hatten, nun kurz und langsam läuft, schreiben wir: „Alex ε (kurz, langsam) laufen“. Jeder Informatiker erkennt *laufen* sofort neutral als Prozedurnamen und die Apprädikatoren als Parameter, die die Umstände einer Prozedur genauer schildern. Bei der schematischen Darstellung der Prozedur *laufen* ergibt sich nur ein Problem. *Lang* und *schnell* wie auch *kurz* und *langsam* sind keine Beschreibungs-Prädikatoren, die einfach wie etwa Bleistift und Kugelschreiber durch Beispiel und Gegenbeispiel eingeführt werden können. Es handelt sich um Beurteilungs-Prädikatoren, die einem Schüler sehr wohl durch die leidigen Zensuren in der Schule geläufig sind. Er weiß, dass ein schlechter Schüler (*schlecht* ist hier ein Ding-Apprädikator) von einem guten nur unterschieden werden kann, wenn ein Regelwerk, ein Maßstab vorliegt, der bei Beurteilungs-Prädikatoren immer erforderlich ist. Wir wollen im Fall der laufenden Hunde annehmen, dass *lang* eine Dauer über zwei Stunden ist: Dauer *>2 Stunden ist gleichbedeutend mit lang*. Alles, was darunter liegt, heißt kurz. Für die Geschwindigkeiten schnell und langsam soll gelten: Geschwindigkeit *>10 km/h bedeutet schnell*, alles andere ist langsam. Um Fido und Alex laufen lassen zu können, schreibt man schematisch:

laufen (Name: char, Dauer : integer, Geschw: integer)

und nennt Dauer und Geschw. Formalparameter vom Typ oder Schema *ganze Zahl* („integer“). Name ist vom Typ oder Schema *Buchstaben* („character“). Man sollte besser Schema-Parameter sagen, weil das Wort *Formalparameter* entbehrlich ist. Fido und Alex laufen aber nur lang oder kurz bzw. schnell oder langsam auf Aufforderung hin. Die Aufforderung wird durch ein „call“ wiedergegeben. Der Aufruf: „*call laufen (Fido, 3, 11)*“ verlangt von Fido einiges. Wenn er trainiert ist, schafft er das aber spielend. Mit dem lahmen, untrainierten Alex ist die Sache nicht so. Deshalb fordern wir ihn nur mit einem „*call laufen (Alex, 1, 5)*“ zu einem kurzen und langsamen Lauf auf. Die in Klammern angegebenen Parameter heißen Aktual-Parameter. Gegen die Bezeichnung ist nichts einzuwenden, weil man ja ein Schema aktualisiert. Ausprägungs-Parameter als Terminus ginge aber auch.

Zum Abschluss dieses Abschnitts nur noch zwei Bemerkungen zu den Wörtern *Fido* und *transkulturell*. Der Hundename *Fido* wurde unseres Wissens von R. Carnap als Beispielname in die Literatur eingeführt (in: Meaning and necessity. A study in semantics and modal logic) und erfreut sich mittlerweile auch in der Informatik-Literatur zunehmender Beliebtheit. Was das Wort *transkultu-*

rell in Hinblick auf eine durch Logik erweiterte Rationale Grammatik anbetrifft, so soll es tatsächlich heute noch Menschen geben, die behaupten, die Logik sähe ganz anders aus, wenn Aristoteles (384–322 v. Chr.), ihr Schöpfer, Chinese gewesen wäre. Für diese Menschen, wie aber für uns alle, gilt der Satz des Aristoteles am Beginn seiner zweiten Analytik (Lehre vom Beweis):

„Alles vernünftige Lehren und Lernen geschieht aus einer vorangehenden Erkenntnis".

Extension und Intension

Jeder Schüler wird das Verstehen des syntaktischen Aufbaus eines Elementarsatzes als einfach empfinden. Unmittelbar einleuchtend für ihn ist es auch, ein schriftliches oder mündliches Ausdrücken eines Satzes als Handlung zu begreifen. Aber die Semantik eines Satzes zu erfassen, ist häufig schwierig, nicht nur für Schüler. Auch er sieht und hört, wenn Erwachsene in Politik und Wirtschaft, in Kirche und Kunst, in Sport und Schule, eigentlich fast überall durcheinander reden. Wittgenstein hilft sicher weiter, wenn er in seinen philosophischen Untersuchungen (§ 43) herausstellt: „Die Bedeutung eines Wortes ist sein Gebrauch in der Sprache", und uns damit deutlich sagt, was pragmatisch heißt, dass somit Semantik (Bedeutung) aus der Pragmatik (Handlungsorientierung) folgt. Philosophen, nicht nur Wittgenstein, haben aber die Neigung, ihre gehorsamen Schüler, zu denen wir ja alle gehören, im Regen stehen zu lassen.

So stellte auch der berühmte Kant einmal fest: „Hundert wirkliche Taler enthalten nicht das mindeste mehr, als hundert mögliche" (Kritik der reinen Vernunft, B 627). Er wollte damit sagen, dass Sein keine Eigenschaft der Dinge ist. Als Menschen, darunter auch der in Teil I zitierte J.G. Fichte, in finanzieller Not hundert wirkliche Taler von ihm erbaten, bekamen sie das Geld nicht. Sie blieben im Regen stehen. Wittgenstein, der uns auf die Sprünge geholfen hat, sagt uns auch nicht, dass die Bedeutung eines Wortes nur durch ein oft mühsames Studium seines Gebrauchs („use") verstanden werden kann. Und bei diesem Studium spielen die Begriffe *Extension* (Umfang) und *Intension* (Inhalt) eines Prädikators eine zentrale Rolle.

Beginnen wir mit dem noch relativ einfachen Begriff *Extension* eines Prädikators. Um uns unsere Aufgabe zu erleichtern, erweitern wir zuvor unsere Darstellung eines Elementarsatzes um den Begriff der Mehrstelligkeit. Der Satz „Fido ist die Mutter von Alex" ist ein mehrstelliger Elementarsatz, weil zwei Nominatoren, nämlich Fido und Alex, zusammen auftreten. Schematisiert schreiben wir: *Fido, Alex ε Mutter_von*. Das Wort *Mutter_von* wird zweistelliger Prädikator genannt. Man kann sich auch n-stellige Prädikatoren vorstellen. Dann schreibt man allgemein: $N_1, N_2, \ldots, N_n \; \varepsilon \; P$. Nehmen wir nun an, unser Fido sei mit dem Hund Karo ein zweites Mal Mutter geworden. Dann darf auch der Satz: *Fido, Karo ε Mutter_von* behauptet werden. Man sagt nun gelehrt: Der Prädikator *Mutter_von* hat die Extension oder hat den Umfang <Fido, Alex> und <Fido, Karo>. Ganz gelehrte Leute sprechen auch von einer Klasse (von Tupeln), der der Prädikator *Mutter_von* zukommt. Drüben in der Mathematik wird statt Klasse Menge gesagt. Bei Mehrstelligkeit spricht man statt von Menge auch von *Relation*. Es wird dort auch der Begriff Mächtigkeit (Kardinalität) einer Menge oder einer Relation eingeführt. Diese Terminologie wird vom extensionalen Teil der Rationalen Grammatik übernommen. Der Satz: „Die *Relation Mutter_von* hat die Mächtigkeit zwei" ist sinnvoll.

Wesentlich schwieriger ist es nun, die Intension (Inhalt) des zweistelligen Prädikators *Mutter_von* zu klären. Der berühmte Frege sprach noch von Sinn, statt von Intension. Das Wortpaar *Extension-Intension* wurde erst von seinem Schüler R. Carnap in die Rationale Grammatik eingeführt. Man spricht auch heute von einer intensionalen Semantik, wenn man ein Gebrauchsstudium eines Prädikators betreibt. Wer Untersuchungen eines Umfangs durchführt, von dem sagt man, dass er eine extensionale Semantik eines Prädikators aufbaut. Zur mühsamen Festlegung der Intension des Prädikators *Mutter_von* gehört sicherlich der zweistellige Geschehnisprädikator *gebären*. Wenn Fido den Alex gebärt, dann ist Fido die Mutter von Alex. Zur intensionalen Semantik von „Mutter_von" gehört aber auch die empirische Wahrheit; wenn Fido die Mutter von Alex ist, dann ist Fido weiblich. Diesen Satz wollen wir auch schematisch hinschreiben: *Fido, Alex ε Mutter_von* $\rightarrow$ *Fido ε weiblich*. Es liegt hier schon ein komplexer Satz vor, weil zwei Elementarsätze durch ein Wenn ... Dann ($\rightarrow$) verknüpft werden. Ein verknüpfender Wenn ... Dann-Pfeil ($\rightarrow$) ist aber zu unterscheiden von dem Doppelpfeil einer Regel ($\Rightarrow$). Die Regel $A \Rightarrow B$, schematisch hingeschrieben, bedeutet, dass es erlaubt ist, vom Satz A zum Satz B überzugehen. Es liegt eine Übergangserlaubnis und keine Verknüpfung vor. Wie beide, Verknüpfung und Regel, rational grammatisch zusammenhängen, wird im Teil VI „Logik und die Geltungssicherung von Behauptungen" zu klären sein. Wir wollen es an dieser Stelle mit einer methodischen Bemerkung belassen: Eine *Regel* geht im Aufbau einer *Wenn ... Dann-Verknüpfung* methodisch voran. Erst muss die Regel feststehen, dann kann eine komplexe Verknüpfung zu einem Satz mit *Wenn ... Dann* erfolgen. Eine *Regel* selbst, das sind noch zwei Sätze, nämlich ein Satz vor und ein Satz nach dem Doppelpfeil. Ein *Wenn ... Dann-Satz*, das ist ein komplexer, ein zusammengesetzter Satz.

Allgemein können wir sagen, dass die Intension eines Prädikators durch eine Fülle von *Regeln*, und dann bei ihrer logischen Untersuchung, durch eine Fülle von *Wenn ... Dann-Verknüpfungen* festgelegt wird. Wenn das nicht geschieht wie im täglichen Leben, ja dann kann es zu dem großen, beklagenswerten Durcheinander in dieser Welt kommen. Es ist leicht, an dieser Stelle den Begriff *Wissenschaft* einzuführen, den jeder Schüler in diesem Kontext sofort intuitiv versteht: Wissenschaft ist der beharrliche Versuch, dem Durcheinader in dieser Welt – die Griechen sprachen auch von Chaos – den Garaus zu machen. Seit den Zeiten eines Wilhelm von Humboldt (1767–1835) ist und bleibt *Bildung durch Wissenschaft* halt ein hehres Ziel.

Die Mehrdeutigkeit des Wörtchens *ist*

In der offiziellen Grammatik der deutschen Sprache heißt es: *ist* ist die dritte Person, Singular, Präsens des Hilfszeitwortes *sein*. Bis der Logiker G. Frege Ende des 19. Jahrhunderts auftrat und eine Rationale Grammatik konzipierte, lebte dieses Wörtchen wie auch seine Geschwister in anderen natürlichen Sprachen, vorneweg auch das englische „is", *unschuldig* im Dornröschenschlaf vor sich hin. Im Wörterbuch von Mittelstraß [2] steht in Band 2 unter dem Stichwort *ist*: „Es handelt sich um ein Musterbeispiel dafür, dass die Grammatik der Alltagssprache den Anforderungen der Logik nicht genügt, weil es wichtige logische Unterschiede verdeckt". In der Tat: *ist* ist hochgradig mehrdeutig und deshalb auch schon in einer Informatik als Grundbildung mit einem Warnschild zu versehen. Wir wollen uns hier in aller Kürze zunächst auf fünf Bedeutungen beschränken.

Die Prädikation ‚ε'

In unserem Beipielsatz „Fido ist ein Hund", rational-grammatisch wiedergegeben mit der Kopula, „Fido ε Hund" (N ε P), liegt eine Deutung als Sprachhandlung zugrunde. Alle anderen Bedeutungen können mit dieser Ursprungsbedeutung rekonstruiert werden. Eine Prädikation als Sprachhandlung ist unhintergehbar, d.h. davor gibt es keine Aussage und damit auch keine Information. ‚ε' ist kein Prädikator und kein Nominator, sondern ein Zeichen eigener Art (lat. sui generis).

Die Elementbeziehung von Mengen ‚$\in$'

Man sagt jetzt für „Fido ist ein Hund" auch: Fido ist ein Element der Menge *Hund* und schreibt „Fido ∈ Hund" (N ∈ P). Rekonstruiert wird diese mengentheoretische Lesart durch: *Fido, Hund* $\varepsilon \in$ oder allgemein *N, P* $\varepsilon \in$. Hierin wird ‚∈' als zweistelliger Prädikator gedeutet, der auch zuweilen Relator genannt wird.

Gegenüber der Mathematik ist dies eine ungewöhnliche Schreibweise. Aber es handelt sich um eine Rekonstruktion und nicht um eine mathematische Notation, gegen die niemand etwas einzuwenden hat.

Identität ‚$=$'

Klassisch wird in diesem Zusammenhang der Fregesche Satz genannt: „Der Morgenstern *ist* der Abendstern". Das ist aber extensional gemeint. In beiden Fällen, in seiner „Abendfunktion" wie auch in seiner „Morgenfunktion", handelt es sich um unseren Planeten Venus als einzige Extension, die die Identitätsbedingung erfüllt. „*Morgenstern, Abendstern* $\varepsilon =$" oder „*N, P* $\varepsilon =$". Auch unser altbekanntes mathematisches Gleichheitszeichen entpuppt sich als ein zweistelliger Prädikator.

Inklusion ‚$\subseteq$'

Beispiel: Hund ist enthalten in Tier. Gemeint ist, dass die Menge der Hunde eine Untermenge der Tiere ist. Das Inklusionzeichen ‚⊆' kann auf das Element_von-Zeichen ‚∈' zurückgeführt werden, und das wiederum wird, wie oben gezeigt, mit Hilfe der Kopula rekonstruiert. Allgemein wird $N \subseteq P$ definiert durch: für alle x (wir schreiben später dafür abkürzend „$\forall_x$ bzw. $\bigwedge_x$) $x \in N \rightarrow x \in P$.
Bemerkung: Eine Inklusion wird durch einen komplexen Satz rekonstruiert.

Subordination ‚is_a'

‚is_a' spielt in der Datenmodellierung der Informatik eine zentrale Rolle und gehört alleine deshalb schon zum Pensum einer Informatik als Grundbildung.

Beispiel: „Hund ist ein Tier" oder „*Hund is_a Tier*". Es ist ein Ausschnitt aus einer Begriffshierarchie gemeint, die dann mit „Tier ist ein Lebewesen" nach oben hin weitergeführt werden kann. ‚is_a' ist der Inklusion ‚$\subseteq$' sehr ähnlich, nur mit dem feinen Unterschied, dass eine Inklusion mengentheoretisch oder extensional gedeutet wird, während ‚is_a' im Sinne einer Begriffshierarchie intensional zu sehen ist. Man sagt, dass eine Hierarchie von Begriffsinhalten – und nicht wie bei der Inklusion eine Hierarchie von Begriffsumfängen – zur Debatte steht. Was Hunde und Tiere intensional sind, ist in letzter Instanz die Angelegenheit der Zoologie. Extensionen von Hunden festzustellen, ist z.B. die Angelegenheit eines städtischen Hundesteueramts, das Hundemarken mit Hunde-Nummern als Primärschlüssel vergibt. Zwischen dem Fach Zoologie und Hundesteuerämtern besteht ein gewaltiger Unterschied, so wie zwischen Intension und Extension, was jeder Schüler sofort einsieht.

N is_a P wird, in der oben eingeführten Schreibweise für den so genannten Allquantor $\wedge$, definiert durch die Subordination: $\wedge_X\ x\ \varepsilon\ N \rightarrow x\ \varepsilon\ P$, und ist ebenfalls rational-grammatisch gesehen ein komplexer Satz. Mit schematischen Buchstaben, die im Teil I eingeführt wurden, darf man aber auch vereinfacht schreiben: $X\ \varepsilon\ N \rightarrow X\ \varepsilon\ P$. X ist ein schematischer Buchstabe. Er gilt als fest, aber beliebig. Bemerkung: Eine Rekonstruktion „*N, P ε is_a*" ist unpräzise, weil die hierarchische Subordination N unter P verloren geht.

Alle fünf Mehrdeutigkeiten sind in der Fachliteratur wohl bekannt. Dass aber die handlungsbasierte Prädikation eine Deutungsbasis ist, bleibt nicht selten unberücksichtigt. Es kann noch eine sechste Mehrdeutigkeit vorgetragen werden, die im Zusammenhang mit Teil I „Schema und Ausprägung" verständlich ist und im Fach Informatik, das wir Inschematik nannten, eine große Rolle spielt. Eine Rationale Grammatik ist offen. Deshalb kann z.B. auch der Satz : „Drei *ist* ein Integer" rekonstruiert werden zu „*3, Integer ε instance_of*" . Man könnte auch in einem anderen Beispiel statt: „4711 ist eine Schüler-Nummer" schematisch schreiben „*4711, Schüler-Nummer ε instance_of*". Das hat bisher in der Grundlagenliteratur noch niemand so gemacht. Wie beim *Element_von*-Prädikator ($\in$) – dort wird eine Menge vorausgesetzt – ist beim *instance_of* -Prädikator ein Schema vonnöten. Die Prädikation (ε) ist hingegen voraussetzungslos. Sie ist unhintergehbar (Mittelstraß) und kein Prädikator. Wer handelt, der operiert. Die Prädikation ist – so gesehen – ein herausgestellter Basis-Operator. Es ist nun kein Geheimnis: Schemata werden sehr häufig intensional gedeutet und nicht ausschließlich extensional wie der *Element_von*-Prädikator ($\in$). Aber wie gesagt, eine Rationale Grammatik ist offen und die Informatik bedarf offensichtlich einer Rekonstruktion von *instance_of*. Systematisch sieht der sechste Fall wie folgt aus:

Subsumtion ‚instance_of'

In „*A instance_of B*" ist *A* eine Ausprägung, eine Aktualisierung des Schemas (Typ) *B*. Man sagt auch in einer anderen Sprechweise: *A* wird unter *B* subsumiert oder *A* wird unter das Schema *B* gebracht. Rekonstruiert schreibt man: *A, B ε instance_of*. Es handelt sich um eine elementare Subsumtion, weil nur ein Elementarsatz zur Rekonstruktion benötigt wird. Wenn beispielsweise im Bereich der Gesetze ein Sachverhalt, z.B. der Strafsachverhalt *Diebstahl*, unter einen gesetzlichen Tatbestand (hier §242 Strafgesetzbuch [Diebstahl]) gebracht werden muss, spricht man von einer juristischen Subsumtion, die nur mit komplexen Sätzen rekonstruiert werden kann, wie in [4] dargestellt wird. Im Teil VI „Logik und die Geltungssicherung von Behauptungen" wird auf komplexe Subsumtionen eingegangen werden.

Literatur

1. Gerhardus, D.; Kledzik, S.M.; Reitzig, G.H.: Schlüssiges Argumentieren. Göttingen, VR Verlag 1975
2. Mittelstraß, J. (Hrsg.): Enzyklopädie, Philosophie und Wissenschaftstheorie, Bd 1–4. Stuttgart: Metzler 1980–1996
3. Lorenzen, P.: Lehrbuch der konstruktiven Wissenschaftstheorie. Stuttgart, Metzler 2000
4. Inhetveen, R.: Logik – Eine dialog-orientierte Einführung. Leipzig: Edition am Gutenbergplatz 2003
5. Wedekind H., Ortner E.: Toward Universal Literacy – From Computer Science upward, in: Commun ACM, Vol. 47, No. 6 (June 2004), p. 101–104

Informatik als Grundbildung

Teil III: Gleichheit und Abstraktion

H. Wedekind · E. Ortner · R. Inhetveen

Informatik als Grundbildung wird als eine methodische Vorbereitung der Schüler auf einen Informatikunterricht an allgemeinbildenden Schulen neben eine sprachliche, mathematische und naturwissenschaftliche Grundbildung gestellt [1]. Das Gebiet wird in sechs Teilen vorgetragen. Teil I behandelte das Thema Schema und Ausprägung, Teil II war den Elementarsätzen einer rationalen Grammatik gewidmet. Der vorliegende Teil III stellt das Abstraktionsprinzip und seine Anwendungen vor. Der folgende Teil IV wird das Thema Objekt- und Metasprache behandeln.

Identität und Gleichheit

Unsere natürliche Sprache macht zwischen Identität und Gleichheit einen Unterschied: Dass zwei verschiedene Damen bei einem festlichen Ball das gleiche Kleid tragen, ist zwar peinlich, aber möglich, dass sie dasselbe Kleid tragen ist unmöglich. *Dasselbe* gibt es nur einmal. Sprachlich ist eine Identität also so etwas wie eine *Gleichheit in jeder Hinsicht*, Gleichheit wäre dann eine *Identität in einer bestimmten Hinsicht*. Diesen (sozusagen hermeneutischen) Befund gilt es zuerst einmal zu präzisieren.

Identität

Was man unter der Idee einer Gleichheit in jeder Hinsicht verstehen sollte, hat wohl als erster Leibniz (1646–1716) präzise vorgeschlagen: Zwei *A* und *B* genannte Gegenstände werden identisch genannt, wenn es keine Möglichkeit gibt, sie sprachlich durch einen Satz zu unterscheiden, der auf einen der beiden zutrifft und auf den anderen nicht. Für jeden (denkbaren, sinnvollen) Satz P(*A*) gilt also P(*A*) genau dann, wenn P(*B*) gilt. (Die für eine auch formal präzise Fassung erforderlichen Mittel werden in Teil VI bereitgestellt.) Man nennt diese Leibnizsche Einsicht heute gerne das *Prinzip der Identität des Ununterscheidbaren*. Dass hier von einem Prinzip die Rede ist – und nicht schlicht von einem Satz – liegt daran, dass darin auch auf Sätze Bezug genommen wird, die erst in der Zukunft einzuführende Prädikatoren benutzen: Auch für sie soll die Ununterscheidbarkeit gelten. Diese Eigenschaft haben normale Sätze nicht.

Betrachten wir zur Illustration das Fregesche Beispiel unseres Gebrauchs von *Morgenstern*, *Abendstern* und *Venus*: Diese Wörter sind natürlich in jeweils unterschiedlicher Absicht eingeführt worden. Dafür sagt man auch: Die Eigennamen sind intensional verschieden. Fragt man freilich einen Astronomen, so bekommt man die Auskunft, es handle sich in allen 3 Fällen um *dasselbe* Objekt. Jeder Satz, der für einen der 3 Himmelskörper zutrifft, gilt auch für die beiden anderen. Scheinbare Ausnahmen fallen einem leicht ein, etwa der Satz „Der Abendstern heißt so, weil er im Frühsommer als erster (weil hellster) Stern am Abendhimmel erscheint." Ersetzt man hierin *Abendstern* durch *Morgenstern*, so wird der vorher wahre Satz natürlich falsch. Aber hier wurde, wie Wittgenstein sagen würde, unser Verstand durch die Mittel unserer Sprache verhext. In genauer Formulierung hätten wir sagen müssen: „Das ‚Abendstern' genannte Himmelsobjekt

DOI 10.1007/s00287-004-0411-z

H. Wedekind · R. Inhetveen
Universität Erlangen-Nürnberg,
Fachbereich Informatik,
Deutschland
E-Mail: wedekind@informatik.uni-erlangen.de
E. Ortner
TU Darmstadt,
Fachgebiet Wirtschaftsinformatik –
Entwicklung von Anwendungssystemen,
Deutschland

heißt so, weil ...“ oder auch „Der Abendstern heißt ‚Abendstern‘, weil er im Frühsommer ...“. Im letzten Satz zeigt sich deutlich, dass darin zwar die Ersetzung durch *Morgenstern* vorgenommen werden darf, aber nur an der Stelle, wo es sich um das Objekt handelt, und nicht dort, wo sein Name (in einfacher Anführung) steht. Doch dazu wird sich Näheres im folgenden Teil IV dieser Serie finden.

Resümieren wir unsere Erkenntnisse kurz:

- Identität wird durch das Leibnizsche Prinzip definiert.
- Identität ist Einmaligkeit. Je zwei oder mehr identische Objekte sind immer nur ein Individuum.
- Intensional verschiedene Objekte können identisch sein.

Zu dieser kurzen Liste wird gleich noch ein Punkt hinzuzunehmen sein:

- Identität ist eine Äquivalenzrelation.

Gleichheit

Dass Gleichheit zwischen verschiedenen Dingen immer nur in gewisser Hinsicht besteht, lernen wir im Allgemeinen stillschweigend. Wir lernen etwa, dass alle 10-EUR-Noten gleich viel *wert* seien, unabhängig vom Herstellerland, vom Besitzer oder vom Grad ihrer Verschmutzung. Und lange davor lernen wir, dass runde, luftgefüllte Gummispielsachen *Ball* genannt werden, ganz unabhängig davon, wie sie bemalt sind oder wer sie geschenkt hat. In diesem letzten Beispiel handelt es sich um eine Schemabildung, genauer: um eine *spontane* Schemabildung (vgl. Teil I). Wenn wir uns mit Gleichheiten beschäftigen, tun wir gut daran, die Hinsicht, in der Gleichheit besteht, ausdrücklich mit anzugeben. Wir sprechen also von Wertgleichheit im Fall von Banknoten. Wovon sollen wir bei Bällen sprechen? Kein Mensch sagt das, aber wir können hier z.B. von Spielgleichheit sprechen, wenn sich mit verschiedenen Bällen gleich gut spielen lässt. Diese Spielgleichheit ist ein Beispiel für eine Funktionsäquivalenz, wie sie uns in technischen Zusammenhängen, aber auch in der Informatik, ständig begegnet. Damit ist schon angedeutet, dass eine (zweistellige) Relation nur dann eine Gleichheit genannt wird, wenn sie eine Äquivalenzrelation ist.

Äquivalenzrelationen liegen vor, wenn die folgenden 3 Eigenschaften erfüllt sind (das Relationszeichen sei ~):

- Reflexivität: $x \sim x$
- Symmetrie: wenn $x \sim y$, dann $y \sim x$
- Transitivität: wenn $x \sim y$ und $y \sim z$, dann $x \sim z$

Ein besonders prominentes Beispiel einer Gleichheit bildet der Fall gleicher Resultate beim Abzählen von Dingen. Diese *Gleichzähligkeit*, die nachgewiesen wird über die Existenz (und das heißt: Konstruktion!) einer ein-eindeutigen Abbildung, bildet bei Hume (1711–1776) die Grundlage für eine Definition dessen, was heute auf andere Weise als die gleiche *Mächtigkeit* von endlichen Mengen definiert wird [2]. (Hume spricht für eine betrachteten Begriff **F** von der „Anzahl der **F**s“). Bei Frege wird die Gleichzähligkeit dann zur Basis einer Definition des Zahlbegriffs. Die dabei verwendete Methode lässt sich allgemein beschreiben und deshalb auch allgemein nutzen.

Das Abstraktionsprinzip

Das sog. Abstraktionsprinzip tritt meist in Form von konkreten Fällen von Abstraktion auf. Bei Hume war das die Gleichheit der Anzahlen, bei Frege sind es später die Gleichheit der Richtung (von Geraden in der euklidischen Geometrie) sowie in seinen *Grundgesetzen* die Gleichheit von sog. Wertverläufen. Immer wird dabei vorausgesetzt, dass wir zwischen betrachteten Gegenständen (Begriffe, Geraden, Funktionen) eine Äquivalenzrelation (Gleichzähligkeit, Parallelität, extensionale Gleichheit) schon kennen.

Formulierung

Diese Voraussetzung wird nun in einem allgemeinen Abstraktionsprinzip folgendermaßen genutzt: Wir setzen voraus, dass wir

- von wohlbekannten Dingen $a, b, c, \ldots$ sprechen und
- dass zwischen ihnen eine Äquivalenzrelation ~ definiert ist.

Dann schreiben wir

$$\tilde{a} = \tilde{b} \leftrightarrow a \sim b$$

und lesen das als: Der abstrakte Gegenstand $\tilde{a}$ ist gleich dem abstrakten Gegenstand $\tilde{b}$ genau dann, wenn $a \sim b$ gilt.

Wenn man es hierbei belässt, ist wenig gewonnen: Es scheint, als hätten wir nur eine Gleichheit (~) durch eine andere (=) ersetzt und dabei offen gelassen, was denn die geheimnisvollen *abstrakten*

Gegenstände $\tilde{a}$ und $\tilde{b}$ seien. Zugleich handelt man sich eine Menge Probleme ein, etwa die Frage, was das denn für eine Gleichheit zwischen abstrakten Gegenständen sei. Hier hat nun Lorenzen (1915–1994) für eine weitere Klärung gesorgt (vgl. [4]). Er zeigte, dass die fragliche Gleichheit sogar eine Identität im Sinn des Leibnizschen Prinzips ist, vorausgesetzt nur,

- dass es sich bei $\sim$ auch um eine Äquivalenzrelation handelt (diese Bedingung ist nicht nur hinreichend, sondern auch notwendig) und
- dass wir uns bei den im Leibnizschen Prinzip auftretenden Prädikatoren P auf solche beschränken, die hinsichtlich $\sim$ im folgenden Sinn *invariant* sind:

$$a \sim b \leftrightarrow (\mathrm{P}(a) \leftrightarrow \mathrm{P}(b)).$$

Terminologisch sagt man auch: der (indefinite) Allquantor (zweiter Stufe) über die Sätze P im Leibnizschen Prinzip ist einzuschränken auf invariante P. Auf Deutsch heißt das: über die $\tilde{a}$ usw. werden nur invariante Aussagen gemacht. Auf alle anderen wird verzichtet. *Deshalb* haben Zahlen keine Farben, Richtungen keinen Geruch und Wertverläufe keinen Besitzer.

Konsequenzen

Aber wir müssen noch ein paar mehr Anmerkungen machen. Die erste betrifft die Frage nach der *Existenzweise* der Abstrakta: Wo sind die eigentlich? In unseren Köpfen, in einem Computer oder gar im Platonschen Ideenhimmel? In welchem Sinn existieren sie? Die Antwort lautet: wir brauchen über ihre Existenz nichts zu behaupten, es gibt sie nur im Sinn einer sehr praktischen und nützlichen „façon de parler“. Nur wenn wir die selbst auferlegte Beschränkung verletzen, uns auf invariante Prädikatoren zu beschränken, treten Probleme auf.

Mathematiker nennen alle Abstrakta *Äquivalenzklassen* und vertrauen hinsichtlich ihrer Existenz in aller Regel auf Platon. Angelelli [3] nannte dieses Verfahren der Existenzsicherung die Looking-around-Methode. Im menschlichen Alltag wird looking-around gerne als Zeichen von Verwirrung gedeutet. Die alternative Vorgehensweise, also die Rede von Abstrakta als „façon de parler“ zu betrachten, nennt man gelegentlich auch (zusammen mit dem Abstraktionsprinzip und der Invarianzbedingung) *moderne Abstraktionstheorie*. Obwohl im Allgemeinen die Rede von *modern* nichts (mehr) mit der von *gut* zu tun hat: Hier schon.

Ein zweiter Punkt betrifft den Zusammenhang mit der Rede von Schema und Ausprägung. Schon der Verweis auf die Invarianzbedingung verdeutlicht, dass die Rede von abstrakten Gegenständen methodisch *nicht* dasselbe ist, wie z.B. die Rede von einer Maus. Jedes Abstraktum $\tilde{a}$, das in einer Behauptung $\mathrm{P}(\tilde{a})$ auftritt, muss, wenn diese Behauptung geprüft werden soll, durch einen *Repräsentanten* (z.B. a) ersetzt werden. Abstrakta sind deshalb auch Schemata, ihre Ausprägungen heißen traditionell Repräsentanten. *Aber*: Abstrakta müssen wir nicht spontan bilden, sondern wir können sie (bei Bedarf!) selbst explizit konstruieren. Das ist der Punkt, an dem die Abstraktion für die Informatik geradezu zur *Tugendlehre* wird. Die Mathematiker führen beispielhaft vor, wie ihre ganze Wissenschaft durch abwechselnde Konstruktionsschritte (für die benötigten Gegenstände) und Abstraktionsschritte aufgebaut werden kann. Die Informatik ist nur in einer leicht verschiedenen Lage: Sie bezieht die Hardware aus der Industrie, die Software aber konstruiert sie selbst. Abstraktionsschritte sind für beide Arten von Grundgegenständen sinnvoll und möglich.

Als drittes muss auf eine Tatsache hingewiesen werden, die mit dem Leibnizschen Prinzip zusammenhängt: Abstrakta sind Individuen. Es gibt eine bestimmte Melodie immer nur einmal, egal wie oft und auf welche Weise sie repräsentiert wird. Es gibt den Burrows-Wheeler-Kompressionsalgorithmus nur einmal, egal wie oft und von wem er wie elegant oder unelegant programmiert wird. Das schließt nicht aus, dass es von einer bestimmten Sorte von Abstrakta mehrere bis beliebig viele geben kann: Zahlen, Melodien, Algorithmen, Konstruktionspläne usw. gibt es jeweils viele. Das ermöglicht Unterschiede (gerade – ungerade; populär – unbeliebt; sortieren – vergleichen; sparsam – verschwenderisch), die genau dann interessant sind, wenn die fragliche Eigenschaft wieder invariant (zwischen Repräsentanten) ist. Die ganze Zahlentheorie lebt von dieser Bemerkung.

Man kann den letzten Punkt auch anders herum wenden: nicht immer, wenn man eine Äquivalenzrelation kennt, macht es auch schon Sinn, darauf eine Abstraktion zu stützen. So lange nämlich nicht, wie man keinen halbwegs interessanten Vorrat an invarianten Prädikatoren hat. Als Beispiel sei hier an die Rede von Energie erinnert. Energie tritt bekanntlich in verschiedenen Formen auf: als kinetische, potenzielle, elektrische, chemische oder als Wärmeenergie. All diese Formen lassen sich jeweils mittels eines *Äquivalents* ineinander umrechnen. Aber weil es nur eine Handvoll Repräsentanten von Energie gibt, verzichtet man in der Physik auf einen expliziten Abstraktionsprozess und beschränkt sich darauf, die genannten Apprädikatoren zu verwenden.

Schließlich ist noch eine Bemerkung zur Rede von *Abstraktionsstufen* zu machen. Das bekannteste Beispiel einer solchen Stufung bildet unser Zahlsystem. Es beginnt mit konkreten Zählergebnissen, die z.B. Strichlisten, Häufchen kleiner Muscheln, Kerben in einem Holz oder Knoten in einer Schnur sein können. Dann kommt die erwähnte Zählgleichheit ins Spiel: Wenn zwei Zählergebnisse zählgleich sind (das stellt man operativ fest), dann repräsentieren sie dieselbe Zahl. Hat man die Grundzahlen, so bildet man Paare (a, b) und definiert zwischen ihnen wieder eine Äquivalenzrelation durch

$$(a,b) \sim (c,d) \leftrightarrow (a+d) = (b+c).$$

Invariante Rede führt zu den ganzen Zahlen einschließlich der Null. Danach bildet man erneut Paare (q, r) und zwischen ihnen eine Äquivalenzrelation

$$(q,r) \overset{*}{\sim} (s,t) \leftrightarrow q \cdot t = r \cdot s \quad mit \quad r \neq 0 \wedge t \neq 0.$$

Das Ergebnis sind die rationalen Zahlen. Wieder kommt ein Konstruktionsschritt: wir bilden Folgen $f_* = f_1,\ f_2,\ \ldots$, zeichnen darunter durch das bekannte Konvergenzkriterium Cauchyfolgen aus und definieren als Äquivalenzrelation zwischen Cauchyfolgen die Relation „die Differenzenfolge ist eine Nullfolge“. So gelangen wir zu den reellen Zahlen. Von dort gelangt man zu den komplexen Zahlen wieder durch Paarbildung und eine naheliegende Äquivalenzrelation.

Eine *Hierarchie* oder Stufung ist damit in folgendem Sinn entstanden: komplexe Zahlen werden repräsentiert durch Paare von reellen Zahlen; diese werden repräsentiert durch Folgen rationaler Zahlen; die rationalen Zahlen werden repräsentiert durch Paare ganzer Zahlen und diese schließlich durch Paare von Grundzahlen, deren Repräsentanten dann endlich *handfest* sind. Sind wir damit auf einen soliden Boden zurückgekehrt? Das ist eine Frage, die pragmatisch zu beantworten ist. Für die Arithmetik mag die Antwort *ja* lauten. Für einen leidenschaftlichen Ontologen könnte freilich das letzte Wort noch nicht gesprochen sein. Aber Ontologen sind in der Regel keine guten Pragmatiker, weil sie selten aus dem Elfenbeinturm herauskommen. Ein mit Grundbildung befasster Lehrer muss sich der Entscheidung vermutlich nicht stellen: Für ihn sind Strichlisten *kindgerecht*.

Bei der Vermittlung von Grundbildung, so ist hier der Rat, geht es um lehr- und lernbare Einzelschritte und *nicht* um ein ontologisches *Fundament*. Die Basis für die jeweils notwendigen Abstraktionsschritte wird pragmatisch bestimmt aufgrund der genannten Kriterien. Ontologie soll eine philosophische Spezialdisziplin bleiben und weder in die Grundbildung noch in die Informatik hineinwirken. Dass freilich in letzter Zeit die Rede von *Ontologien* auch in der Informatik Verbreitung findet, wäre einen eigenen Beitrag wert.

Anwendungen

Doch nun sollen noch einige Beispiele für Abstraktionsprozesse folgen, die zum Teil in der Literatur bekanntes wiederholen, zum Teil vielleicht neue Anregungen geben. Wegen ihrer Bedeutung für die Grundbildung beginnen wir dabei mit einigen auf die (Normal)sprache bezogenen Beispielen und schließen dann einige der Informatik näherstehende an. Insgesamt ist die Auswahl als exemplarisch zu verstehen und verbunden mit der Aufforderung, sie durch eigene Beispiele zu ergänzen. Weitere Hinweise und Beispiele zur Abstraktion finden sich in [5].

Gelungene Anwendungen

Beispiel 1. *Begriffe* lassen sich als Abstrakta rekonstruieren, wenn wir Wörter, genauer: Prädikatoren, als Gegenstandsbereich hernehmen und die zwischen ihnen manchmal bestehende Relation der *Synonymität* herstellen. (Beides kann man problematisieren: Wörter, Prädikatoren sind selbst schon Schemata – wir bewegen uns also in der Hierarchie schon in einem höheren Stockwerk; und dass es echte Synonyme gibt, wird von manchen Sprachwissenschaftlern bezweifelt: hier sind wir großzügiger und lassen etwa ein Wörterbuch deutsch-englisch als ein Verzeichnis von synonymen Wortpaaren zu.) So ergibt sich das Abstraktionsprinzip

Begriff(A) = Begriff(B) $\leftrightarrow$ synonym (A,B).

Zu zeigen ist, (a) dass *synonym* eine Äquivalenzrelation ist (das ist einfach) und (b) dass es invariante Prädikatoren gibt. In unserem Fall sind etwa *anwendbar auf* oder *dient der Unterscheidung von* invariant. Wir können also sagen: „Der Begriff *rot* dient zur Unterscheidung von Farben“ und „Der Begriff *Revolution* ist anwendbar auf politische Ereignisse“. Das sind zwei korrekt gebildete Sätze (und wahrscheinlich stimmen sie auch).

Beispiel 2. *Sachverhalte* lassen sich als Abstrakta rekonstruieren, wenn wir Aussagesätze S_1, S_2... als Gegenstände hernehmen und verwenden, dass die Relation „S_i und S_j sind zugleich wahr oder zugleich falsch“ eine Äquivalenzrelation ist (intensionale Gleichheit des Sinnes der Sätze). Das Abstraktionsprinzip lautet

$$\text{Sachverhalt}(S_1) = \text{Sachverhalt}(S_1) \leftrightarrow (S_1 \leftrightarrow S_2).$$

Beispiele für Satzpaare, die in der genannten Relation stehen, erhält man z.B. durch Aktiv- und Passivkonstruktion, durch syntaktische Umstellungen oder ähnliches. Invariante Prädikatoren bilden etwa die Ausdrücke „ist strafrechtlich relevant“ oder „setzt voraus, dass ...“. Wir können also sagen „Der Sachverhalt ‚Herr X wurde bestohlen‘ ist strafrechtlich relevant.“ oder „Der Sachverhalt ‚Herr Y ist schuldig‘ setzt voraus, dass Y schuldfähig ist“. Natürlich kann auch ein Sachverhalt einen zweiten Sachverhalt (kausal) voraussetzen.

Es ist eine gute Tradition, solche Sachverhalte, die von *wahren* Sätzen repräsentiert werden, als *Tatsachen* zu bezeichnen. So wird dieses Wort auch im Alltag gewöhnlich verwendet.

Beispiel 3. *Mengen* lassen sich als Abstrakta rekonstruieren, wenn wir Aussageformen (Prädikatoren) $P_1(x)$, $P_2(x)$... als Gegenstände hernehmen und verwenden, dass die Relation „$P_i(x)$ und $P_j(x)$ sind für jedes zulässige x zugleich wahr oder zugleich falsch“ eine Äquivalenzrelation ist (extensionale Gleichheit der Prädikatoren). Das Abstraktionsprinzip lautet

$$\text{Menge}(\mathrm{P}_1(x)) = \text{Menge}(\mathrm{P}_2(x)) \leftrightarrow \wedge_x(\mathrm{P}_1(x)) \leftrightarrow (\mathrm{P}_2(x)).$$

Mathematiker schreiben dies gewöhnlich in der Form „$\{x|P_1(x)\}$“. Das sieht einfacher aus, aber wenn man die in der Mathematik allgegenwärtige Aussage „$a \in \{x|P_1(x)\}$“ korrekt als „$P_1(a)$“ wiedergibt, sieht „$P_1(a)$“ wiederum einfacher aus.

Beispiel 4. Auch für *Relationen* lassen sich Äquivalenzrelationen einführen. Das ist ein Weg zu dem, was in der Mathematik „cartesisches (Mengen)produkt“ genannt wird. Für die Informatik interessanter sind Fälle wie die der Relation „n ist ein Name von x“, kurz: $n\mathrm{N}x$. In Anwendung des Leibnizschen Prinzips können wir hier eine *Benennungsinvarianz* definieren

$$n\mathrm{N}x \sim m\mathrm{N}y \leftrightarrow x \equiv y$$

Das ist – wegen der rechts stehenden Identität – selbstverständlich eine Äquivalenzrelation. Die dazugehörigen Abstrakta haben keinen eigenen Namen bekommen. Aber genau sie sind es, die in der Beschreibung von Datenbankrelationen R(N,P) oft als Primärschlüssel (N) auftauchen: Primärschlüssel sind benennungsinvariant und eindeutig.

Zusatz: Den Zusammenhang der Abstraktion mit dem Informationsbegriff behandelt [6] (in diesem Heft).

Weniger gelungene Anwendungen

Beispiel 5. Gelegentlich trifft man auf den Satz „Die Wertzuweisung ist eine Abstraktion, weil man dem Wert nicht mehr ansieht, welchem Ausdruck er zugewiesen wurde“. Damit wird natürlich auf eine Invarianz angespielt. Aber dieser Satz ist mit Vorsicht zu genießen. Was dahinter steckt ist die Tatsache (vgl. Beispiel 2), dass sich für Ausdrücke, denen Werte zugewiesen werden sollen, eine einfache Äquivalenz dadurch definieren lässt, dass bei jeder (an allen möglichen Stellen gleichen) Variablenersetzung identische Werte entstehen. Im einfachsten Fall von Termen der Mathematik erhält man so die Termäquivalenz

$$\mathrm{T}(x) \sim \mathrm{T}^*(x) \leftrightarrow \wedge_x \mathrm{T}(x) = \mathrm{T}^*(x)$$

Die Termabstrakta werden dann *Funktionen* genannt. Der zitierte Satz könnte also reformuliert werden als „Wertzuweisung dient als Test auf Termäquivalenz“. Die Neuformulierung klingt zwar ein wenig anders, macht aber – im Gegensatz zum Ausgangssatz – einen guten Sinn. Für den Fall, dass eine Wertzuweisung zur Laufzeit eines Programms stattfindet, gelten analoge Überlegungen.

Beispiel 6. Dieses soll das letzte dieser Liste sein. Es benutzt ein Zitat aus [7]:

Was ist ein Cluster? Abstrahiert man etwas von der jeweiligen Hardware- und Software-Implementierung, so besteht ein Computer-Cluster (Anhäufung) aus mehreren so genannten Rechen-Knoten (Nodes), die über ein oder mehrere Netzwerke (Interconnect) miteinander verbunden sind.

Ersichtlich ist die Rede von *Abstraktion* hier nur *schmückend*, zur Sache trägt sie gar nichts bei. Man kann den Satz verlustfrei verkürzen zu: „Ein Cluster besteht aus mehreren etc.". Aber viel schlimmer ist die Formulierung „abstrahiert man *etwas*". Das hervorgehobene *etwas* ist hier ja deutlich ein Apprädikator zu „abstrahieren" im Sinn von: to abstract slightly. Wie das gehen soll, sagt uns der Autor nicht. Wir können nur vermuten, dass er meint, es komme auf all die vielen weiteren Einzelheiten (den Hersteller der Hardware, den Preis, den Standort, den Sysadmin, den Internetprovider und und und) auch nicht an; aber worauf es ankommt, das sagt er uns nicht. Hier haben wir einen Vertreter der klassischen Methode „Abstraktion ist Absehen von" vor uns. Diese Methode ist z.B. an Gymnasien sehr beliebt, aber sie ist leider ein völlig untauglicher Versuch, Abstrakta einzuführen. Nicht nur ist die Liste dessen, wovon man *absehen* muss, unendlich lang; auch die Frage, was danach eigentlich noch übrig bleibt, wird nicht beantwortet und kann auch nicht beantwortet werden. Und was das Wichtigste hierbei ist: *Absehen von* ist aus den genannten Gründen weder lehrbar noch lernbar und *kann* deshalb nicht zu einer vernünftigen Grundbildung gehören.

Literatur

1. Wedekind H., Ortner E.: Toward Universal Literacy – From Computer Science Upward. Commun ACM 47(6), 101–104 (June 2004)
2. Hume, D.: A Treatise of Human Nature, chap. III.
3. Thiel, C.: Zu Begriff und Geschichte der Abstraktion. In: Prätor, K. (Red.) Aspekte der Abstraktionstheorie. (S. 36–48) Aachen 1988
4. Lorenzen, P.: Gleichheit und Abstraktion. Ratio 4, 77–81 (1962)
5. Inhetveen R.: Logik. Eine dialog-orientierte Einführung. Leipzig 2003
6. Wedekind H.: Gibt es eine passende Antwort auf die Frage „Was ist Information?"? Informatik-Spektrum 4 Forum, Seite 372–377.
7. Cornelius H.: Cluster. Clustering im Enterprise Computing. Linux – Enterprise 7/8, 24 (2004)

Informatik als Grundbildung

Teil V: Namengebung und Kennzeichnung

H. Wedekind · R. Inhetveen · E. Ortner

Im Teil II „Bildung von Elementarsätzen“ wird der Teil „Namengebung und Kennzeichnung“ vorbereitet. Seine Überschrift könnte auch lauten: Was sind Nominatoren und wie kommen sie zustande? Teil V ist wie seine Vorgänger Teil III „Gleichheit und Abstraktion“ und Teil IV „Objektsprache/Metasprache“ als eine Erweiterung einer rationalen Grammatik aufzufassen. Der noch folgende Teil VI „Logik und Geltungsicherung von Behauptungen“ schließt die Serie „Informatik als Grundbildung“ ab.

Benennungen

Wir werden namenlos in eine fast namenlose Welt hineingeboren. Nur über Vater oder Mutter sind wir eindeutig bestimmt, was dann auch auf dem Standesamt institutionell festgestellt und registriert wird. Man kann auch sagen: Wir sind durch Prädikatoren, die uns Vater und Mutter vererben, bei der Geburt eindeutig gekennzeichnet und bekommen auf dem Standesamt und/oder in der Kirche (bei der Taufe) institutionell einen Namen, der uns eigen ist, also einen Eigennamen (engl. „proper name“). Jedes Gemeinschaftswesen legt schon alleine aus Gründen der Verwaltung größten Wert auf eine eindeutige Registrierung seiner lebenden Bürger. Bemerkenswert an dem Verfahren ist, dass eine Kennzeichnung, also eine Charakterisierung eines Gegenstands durch seine Eigenschaften, einer Namengebung in der Regel vorangeht. So erfahren wir von der Namenkunde, dass der höchste Berg Deutschlands, der heute den Eigennamen „Zugspitze“ trägt, früher durch „die Spitze, wo es die Lawinen herunterzieht“ gekennzeichnet wurde. Erst später haben die umliegenden Bewohner und Kartographen aus Gründen der Effizienz den Berg kurz Zugspitze getauft. „Zugspitze“ ist ein Eigenname, „der höchste Berg Deutschlands“ und „die Spitze, wo es die Lawinen herunterzieht“ sind zwei Kennzeichnungen. Die Onomastik, so nennt man Namenkunde oder Namenforschung auch, ist eine spannende und kulturhistorisch bedeutsame Wissenschaft, die u.a. Umwandlungen von Kennzeichnungen in Eigennamen unserer natürlichen Sprache begründet. Wie sonst könnte es verlässlich möglich sein, die Tätigkeit eines *Gänsehirten* als Kennzeichnung für Personen zu erklären, für die dann später der Eigenname *Genscher* eingeführt wurde.

„Die Existenz unbenannter Gegenstände ist der Normalfall, für unbelebte Dinge haben wir fast nie Eigennamen (außer in der Geo- und Kosmographie), für belebte Dinge fast nur bei Menschen und Haustieren“, bemerkt Lorenzen [2]. Das Schwein im Stall trägt klassisch den Eigennamen „Jolante“. Das Schwein in der freien Wildbahn hat keinen Namen. Kühe tragen heute aus Gründen der Kontrolle ihren Eigennamen als Nummer im Ohr. Berühmte Bäume heißen z.B. „Kaisereiche“, weil ein Kaiser sie vermutlich gesetzt hat. Andere, gewöhnliche Bäume im Wald gehen leer aus.

Unsere moderne, informationstechnisch getragene Welt kennzeichnet und „benamst“ unaufhörlich. Postleitzahlen, Kfz-Nummern, Europäische Artikel-Nummern (EAN), die dann zum leicht gerätelesbaren Strich-Code auf Produkten zu sehen sind,

DOI 10.1007/s00287-004-0436-3

H. Wedekind · R. Inhetveen
Fachbereich Informatik,
Universität Erlangen-Nürnberg,
91058 Erlangen
E-Mail: wedekind@informatik.uni-erlangen.de

E. Ortner
Fachgebiet Wirtschaftsinformatik-
Entwicklung von Anwendungssystemen,
TU Darmstadt,
64289 Darmstadt

Internationale Standard-Buchnummern (ISBN), die Unique Resource Identifiers (URI) als Benennung für Internet-Dateien, Personen-, Sach- und Vorgangsnummern sind Beispiele für Bemühungen zur exakten symbolischen Erschließung unserer Welt, indem Zeichen als Vertretung eines Gegenstands oder einer Menge von Gegenständen eingeführt werden. Dass der Bürger sich in seinem Unverständnis oftmals gegen einen vermeintlichen Nummernwahn zur Wehr setzt, mag auch daran liegen, dass er in der Schule nur unzureichend auf die Probleme der Benennung (engl. „naming", „reference", „denotation") in einer Welt, die auf Präzision wert legen muss, vorbereitet wird.

Ein Eigenname ist ein benennender sprachlicher Ausdruck im Gegensatz zu einem unterscheidenden sprachlichen Ausdruck, einem Prädikator. Eigennamen können auch (und dann mehrere) in mehrstelligen Prädikationen auftreten, etwa in dem Satz *Hans liebt Maria* bzw. *Hans, Maria ε lieben*. Von jedem beliebigen Gegenstand, dem ein Prädikator zugesprochen werden kann, wird gesagt, dass er auch durch einen Eigennamen benannt werden kann oder benannt werde könnte [3]. Wenn wir noch keinen Namen verfügbar haben, behelfen wir uns auch ersatzweise mit Zeigehandlungen, etwa: „Dieser Beamer dort in der Ecke", und benutzen dazu ein Demonstrativ-Pronomen oder den bestimmten Artikel. Auch in dem Satz *Heute ist Donnerstag* ist eine Benennung enthalten. Man nennt Orts- und Zeitadverbien wie *heute* und *morgen*, *hier* und *jetzt*, aber auch Personalpronomina wie z.B. *ich, du, er* ... Indikationen, die einen Gegenstand benennen. Zeigehandlungen und Indikationen haben den Nachteil, situationsabhängig zu sein, um damit nur in der jeweiligen Redesituation verstanden zu werden. Auch das Anklicken einer Ikone oder einer Menuposition auf dem Bildschirm ist eine Zeigehandlung zwecks Benennung, die nach einem Schema abläuft und manchmal auch situationsabhängig ist. Man spricht in diesem Zusammenhang allgemeiner von Kontextabhängigkeit. Damit ist die Umgebungsabhängigkeit einer sprachlichen Äußerung gemeint. Wissenschaften versuchen, Kontextabhängigkeiten zu vermeiden oder gar nicht erst aufkommen zu lassen. In der rationalen Grammatik werden Benennungen auf solche Eigennamen beschränkt, die nach Vereinbarung genau einen Gegenstand sprachlich vertreten. Abgesetzt von den Eigennamen stehen die Gattungsnamen, die ganze Gegenstandsbereiche benennen. Wenn wir von Eigennamen im Sinne einer rationalen Grammatik sprechen, dann gehören die Zahlwörter dazu. *Zwei ist eine Primzahl* ist ein gültiger Satz, der verdeutlicht, dass auch Abstrakta und nicht nur Konkreta durch einen Eigennamen vertreten werden. Für Kennzeichnungen gilt das selbstverständlich auch. *Die kleinste Primzahl* ist eine solche Kennzeichnung, die ein abstraktes Objekt sprachlich vertritt.

Im Teil II wurden Elementarsätze einer rationalen Grammatik durch N ε P schematisch dargestellt, wobei N ein Nominator, also ein Eigenname oder eine Kennzeichnung, und P ein Prädikator ist. Was passiert, wenn wir den Nominator einfach weglassen und für ihn nur zur Erinnerung eine Leerstelle (.) einführen, also: *. ε P* schreiben? Man kann an der Leerstelle (.) auch die Variable x als Platzhalter einsetzen, die nichts benennt, und schreiben: x ε P. Frege nannte in seinem berühmten Beitrag über „Funktion und Begriff" (1891) einen solchen Ausdruck ungesättigt. Aus „Fido ist ein Hund" wird so *x ε Hund* und aus *Paris ist die französische Hauptstadt* wird *x ε französische Hauptstadt*. Über ungesättigte Ausdrücke kann nur ausgesagt werden, dass sie unvollständig und ergänzungsbedürftig sind. Sie machen noch nicht einmal einen Sinn, sind also überhaupt nicht zu beurteilen. Und ein Wahrheitswert kann ihnen auch nicht zugesprochen werden. Wie eine mathematische Funktion y = f(x), die einen Wert y nur dann liefert, wenn der unabhängigen Variablen x ein Wert zugewiesen wird, so bedarf der ungesättigte Elementarsatz x ε P eines Eigennamens oder allgemein eines Nominators, um Sinn zu machen und einen Wahrheitswert zu liefern. Eine Leerstelle (.) oder eine Variable x alleine sind objekt-sprachlich gesehen ein Nichts. Belegungen von x durch *Fido* bzw. *Paris* lassen nur die Sätze auf der Metastufe „*Fido*" ε Eigenname bzw. „*Paris*" ε Eigenname zu. Erst eine Sättigung z.B. durch *Fido ε Hund* bzw. *Paris ε französische Hauptstadt* vermittelt einen Sinn und einen Wahrheitswert. Die Sättigung eines ungesättigten Ausdrucks, um die es uns ja in einer methodischen Vorbereitung in diesem Teil V geht, kann wie das Auflösen einer Dissonanz in der Musik aufgefasst werden. Was soll man aber mit einer dissonanten, schlampigen Rede in der Informatik machen, wie z.B. „Wenn das Prädikat P wahr ist, dann ..."? P ist doch so gesprochen ungesättigt und kann gar nicht beurteilt werden. Es müsste heißen „Wenn das Prädikat P gesättigt bzw. ergänzt wird und wahr ist, dann ...". Klar ist, dass ein großzügiges Darüberhinwegsehen fatale Folgen haben kann, wenn es um Unterricht geht. Jugendliche, die nur Äpfel im Korb sehen, glauben schließlich, die Äpfel wüchsen auch im Korb.

Kennzeichnungen

Es war *Bertrand Russell* (1872–1970), der Mann mit dem trutzigen Gesicht, der auch politisch sehr aktiv war, dem wir den größten Beitrag zur Theorie der Kennzeichnungen zu verdanken haben [1]. Natürlich, wie könnte es auch anders sein, geschah dies auf den Grundlagen, die *Gottlob Frege* (1848–1925) gelegt hatte. Man spricht deshalb auch gerechterweise von der Frege-Russellschen Kennzeichnungstheorie. Die Engländer haben für das sehr präzise deutsche Wort *Kennzeichnung* nur den Ausdruck *description* parat. Wenn sie darauf hinweisen wollen, dass ein Gegenstand durch eine eindeutige Darstellung benannt wird, sagen sie *singular* oder *definite description.* Wenn im Deutschen gesondert darauf hingewiesen wird, dass Eindeutigkeit und Existenz vorliegen müssen, dann spricht man von einer echten Kennzeichnung. *Die französische Hauptstadt* und *die kleinste Primzahl* sind Beispiele für echte Kennzeichnungen. *Der gegenwärtige König von Frankreich* und *die größte Primzahl* sind Pseudo-Kennzeichnungen, weil es die beschriebenen Objekte gar nicht gibt. Bei Pseudo-Kennzeichnungen spricht man in der deutschen wie auch in der englischen Umgangssprache lautmalerisch von Bla-Bla.

Ganz anders liegt der Sachverhalt bei *potenziellen Kennzeichnungen*, die dann vorliegen, wenn Eindeutigkeit und Existenz noch nicht nachgewiesen sind. *Der Autor von „Romeo und Julia"* gilt als potenzielle Kennzeichnung, solange nicht gesichert ist, dass Shakespeare seine Werke, insbesondere natürlich Romeo und Julia, alleine verfasst hat.

Potenziellen Kennzeichnungen, die einer historischen Forschung bedürfen, um evtl. in eine echte Kennzeichnung überführt zu werden, sind den potenziellen Kennzeichnungen gegenüberzustellen, die auf die Zukunft gerichtet sind. Informatiker und Ingenieure sprechen dann in ihren Bereichen von *Spezifikation*, ein zentraler Begriff der *Allgemeinen Ingenieurwissenschaft*, den es im Abschnitt *Anwendungen* näher zu erläutern gilt. Wichtig ist es zu erkennen, dass Kennzeichnungen nicht nur zur Benennung, sondern auch zur Darstellung von Objekten in einer Entwurfsphase herangezogen werden können. Für eine Informatik als Grundbildung ist dieser Tatbestand von erheblicher Bedeutung. *Spezifizierende Kennzeichnungen* sollten im Unterricht durch ein umfangreiches Beispielmaterial unterfüttert werden. Von Implementierungen kann Abstand genommen werden.

Was man häufig in Anwendungen einen Klassifikationscode oder eine Klassifikationsverschlüsselung nennt, entspricht einer potenziellen Kennzeichnung. Der Gegenstandsbereich (engl. „domain of discourse") wird in disjunkte Klassen (Mengen) eingeteilt (engl. „to partition"). Als Blattknoten verbleiben im Klassifikationsbaum Klassen und keine Individuen. Machen Individuen die Blattknoten aus, so liegt eine echte Kennzeichnung oder ein Identifikationscode bzw. eine -verschlüsselung vor.

Beim üblichen Verfahren zur Festlegung des Begriffs *Kennzeichnung* benötigt man aus logischer Sicht eine kennzeichnende, einstellige Aussageform A(x) und einer internationalen Konvention gemäß einen Kennzeichnungsoperator, der Jota-Operator (ι) genannt wird und die Eigenschaft hat, die freie Variable x zu binden. Man schreibt: ι_x A(x) als Abkürzung für: Dasjenige x mit der Eigenschaft A(x). Hat man einen Eigennamen u verfügbar, so darf definierend geschrieben werden: $u =_{def} \iota_x A(x)$.

ι_x (x ε französische Hauptstadt) ist ein Beispiel für eine schematisch hingeschriebene Kennzeichnung, wobei die spezielle Kopula-Schreibweise anstelle einer allgemeinen Aussageform A(x) gewählt wurde. In der Kopula-Schreibweise wird eine Kennzeichnung auch wie folgt notiert:

$u =_{def} \iota_x$ (x ε P) als Abkürzung für: Dasjenige x, dem der Prädikator P zukommt.

Eine Kennzeichnung ist ein Nominator und kann somit auch wie ein Eigenname behandelt werden. Für *Paris ε groß* darf auch geschrieben werden:

ι_x (x ε französische Hauptstadt) ε groß. Beide Aussagen sind synonym.

Anwendungen

Aus einer unübersehbaren Fülle von praktischen Anwendungen einer spezifizierenden Kennzeichnung wählen wir nur eine aus: Das Problem der Belegung von Unterrichtsräumen einer Schule steht zur Debatte. Sollte es einmal in einer Schule Kollisionen geben, so wird nicht nur den Schülern vorgeführt, dass sich hinter dem Problem *Raumbelegung* keine Trivialität verbirgt.

Tabelle 1

Schema	Ausprägung	Bemerkung
Raum-Nr. (R-Nr.)	p_1	was
Kalender-Woche (KW)	p_2	wann
Wochentag (WT)	p_3	wann
Anfang-Zeit (AZ)	p_4	wann
End-Zeit (EZ)	p_5	wann
Klasse/Kurs (K/K)	p_6	wofür
Belegungs-Nr. (Bel-Nr.)	u	Eigenname

Was ist eine Belegung eines Unterrichtsraums? Mit Fragen dieser Art beginnen in der Regel *Spezifikationsprobleme*. Eine kennzeichnende Antwort sieht wie folgt aus (Tabelle 1): Bereitstellen eines Unterrichtsraums *(was)*, zu einer bestimmten Zeit *(wann)*, für eine Klasse/Kurs *(wofür)*.

Die p_i (i =1, 2, ..., 6) sind schematische Buchstaben, die mitteilen, dass auf einer Objektebene Ausprägungen zur Verfügung stehen, fest und beliebig in Schranken, die gesondert angegeben werden müssen. So sind Räume in ihrer Kapazität beschränkt, Zeitangaben müssen sich an die vorgeschriebenen Unterrichtszeiten halten und einige Klassen/Kurse können nur in hierfür besonders vorgesehenen Räumen abgehalten werden.

In der nun folgenden Kennzeichnung $u =_{def} \iota_x A(x)$ werden die Ausprägungen $p_1$bis p_6 als Prädikatoren eingesetzt. Eine einzelne Belegung u ist eine Ausprägung einer Belegungs-Nr.

$$u =_{def} \iota_x(x \, \varepsilon \, p_1 \wedge x \, \varepsilon \, p_2 \wedge \ldots\ldots \wedge x \, \varepsilon \, p_6)$$

So wie in der Arithmetik das Summenzeichen (Σ) benutzt wird, dienen in der Logik Quantoren als Abkürzung. Wir benutzen den Allquantor oder Großkonjunktor ($\wedge_i$) und schreiben:

$$u =_{def} \iota_x \wedge_i (x \, \varepsilon \, p_i)$$

Die Indexvariable i wird durch den Allquantor und die Nominatorvariable x durch den Jota-Operator gebunden.

In Datenbankschemata nennt man die Zusammenfassung der sechs Teilschemata von Raum-Nr. bis Klasse/Kurs einen *Schlüsselkandidaten*. Für die Belegungs-Nr. als Primärschlüssel wird sehr zutreffend der Begriff *Surrogat (Ersatz)* eingeführt. Er entspricht in der Definitionslehre dem Definiendum; das ist das, was definiert wird und in einer Definition in der Regel links steht. Der Schlüsselkandidat ist dann das Definiens und wird in der gewohnten Schreibweise rechts aufgeführt. Es dürfte nicht verwunderlich sein, dass Rechnungs-Nummern, Auftrags-Nummern, Lieferschein- Nummern etc. in der Regel alle Vorgangsnummern, mit denen wir täglich zu tun haben, als Surrogate durch echte Kennzeichnungen definiert werden. Unser Datum TT/MM/JJJJ gehört auch dazu. Manche Firmen vergeben einen Eigennamen und sprechen dann vom Firmen- oder Fabrikkalender. Mit 445 als Tage-Nr. weiß z.B. jeder in einer Firma XYZ, welcher Tag in welchem Monat und Jahr gemeint ist.

Wenn wir uns gar vornehmen wollen, ein ganzes Belegungssystem für Unterrichtsräume mit diversen Operatoren zu spezifizieren, geraten wir in den Bereich der potenziellen Kennzeichnungen. Denn ein ganzes System „auf den Punkt genau" als echte Kennzeichnung zu spezifizieren, das gelingt in der Regel allein schon deshalb nicht, weil die Anforderungen an das System nicht vollständig und genau vorgetragen werden. Ein System ist offen und wächst mit der Zeit, was durch aufeinander folgende Freigabeschritte (engl. „releases") nach außen auch sichtbar wird.

Man beginnt den Aufbau komplexer Systeme mit einer Liste elementarer Operationen wie: Eintragen einer Belegung, Verschieben einer Belegung, Zusammenfassen von zwei und mehreren Belegungen zu einer, Zeigen der aktuellen Belegung zu einem Zeitpunkt in Vergangenheit, Gegenwart und Zukunft etc. Komplexe Operationen sind dann das Einbringen einer ganzen Stundentafel oder das Erarbeiten von Vorschlägen für Änderungsketten, die den ganzen Wirkzusammenhang einer Belegungsänderung deutlich machen.

Man braucht heute viele komplexe Anwendungssysteme nicht mehr selbst zu entwickeln (engl. „to make"), man kann sie kaufen (engl. „to buy"); aber spezifizierend kennzeichnen muss man sie immer noch. Beim Kauf wird verlangt, dass man sagt, was man haben will. Und ein Installieren erfordert ein spezifizierendes Kennzeichnen allemal. SAP und

Oracle sind unter den Verkäufern zwei bekannte Firmen, die gerne verkaufen. Für die Installation gibt es Spezialfirmen, die das auch gerne tun. Aber ohne eine abgestimmte Spezifikation unserer Wünsche sind diese Firmen hilflos. Man kauft nur einen allgemeinen Rahmen und muss diesen dann „zuschneidern". Der Ausdruck „customizing" wurde dafür eingeführt. Der Vorteil gekaufter Anwendungssysteme ist, dass große Teile der Operatoren nicht mehr programmiert werden müssen. Jedoch ein spezifizierendes Kennzeichnen, insbesondere bei den Daten, bleibt einem nicht erspart. Es findet dann eine besondere Form des spezifizierenden Kennzeichnens, nämlich ein zuschneiderndes Kennzeichnen („customizing definite description") statt.

Um ein Problembewusstsein zu erzeugen, sind Spezifikationsaufgaben für potenzielle Kennzeichnungen schon in der Grundbildung von Bedeutung. Implementieren von einfachen Systemen ist etwas für spätere Ausbildungsgänge. Und komplexe Systeme? Ja, das ist die Angelegenheit von Profis, was man aber auch schon früh verstehen lernen kann. Ein wenig Hochachtung ist nicht schädlich.

Definitionen

Wenn man seine Mitmenschen in Verlegenheit bringen will, braucht man nur mit der Aufforderung an sie heranzutreten, einen *Begriff* zu definieren. So zu reden, ist gängige Bildungssprache. Was ist ein Messer? Was ist eine Schule? Was ist eine Eisenbahn? Das „arme Opfer", dem solche Fragen gestellt werden, sucht einem Ansinnen dieser Art im Allgemeinen zu entfliehen, es sei denn die Frage kommt aus kindlichem Mund oder wird „dienstamtlich" gestellt. Der Geplagte beginnt in seiner Not, Prädikatoren mehr oder weniger willkürlich aufzuzählen, die ihm zur Frage einfallen. Eine „literarische Besonderheit" zur Definition der Eisenbahn ist uns aus den Veröffentlichungen (1879) des alten Deutschen Reichsgerichts in Leipzig überliefert worden (s. [4]):

> „Eine Eisenbahn ist ein Unternehmen, gerichtet auf wiederholte Fortbewegung von Personen und Sachen über nicht ganz unbedeutende Raumstrecken auf metallener Grundlage, welche durch ihre Konsistenz, Konstruktion und Glätte den Transport großer Gewichtsmassen beziehungsweise die Er-zielung einer verhältnismäßig bedeutenden Schnelligkeit der Transportbewegung zu ermöglichen bestimmt ist, und durch diese Eigenart in Verbindung mit den außerdem zur Erzeugung der Transportbewegung benutzten Naturkräften – Dampf, Elektrizität, tierischer oder menschlicher Muskel-tätigkeit, bei geneigter Ebene der Bahn auch schon durch eigene Schwere der Transportgefäße und deren Ladung usf. – bei dem Betrieb des Unternehmens auf derselben eine verhältnismäßig gewaltige, je nach den Umständen nur bezweckterweise nützliche oder auch Menschenleben vernichtende und menschliche Gesundheit vernichtende Wirkung zu erzeugen fähig ist".

Man nennt eine solche Definition eine Real- oder Sachdefinition und grenzt sie von den Nominal- oder Wortdefinitionen ab. Eine Real- oder Sachdefinition ist das, was man im täglichen Leben unter einer Definition versteht. Nominal- oder Wortdefinitionen sind in den Wissenschaften gang und gäbe. Real- und Sachdefinitionen ziehen zu Recht das Gespött und den Hohn der Menschen auf sich. Ein findiger Mitbürger hat denn auch sogleich eine kabarettreife Realdefinition dessen geliefert, was man unter einem Reichsgericht zu verstehen hat [4]:

> „Was ist ein Reichsgericht? Ein Reichsgericht ist eine Einrichtung, welche dem allgemeinen Verständnis entgegenkommen sollende, aber bisweilen durch sich nicht ganz vermeiden haben lassenden, nicht ganz unbedeutende beziehungsweise verhältnismäßig gewaltige Fehler im Satzbau auf der schiefen Ebene des durch verschnörkelte und ineinander geschachtelte Perioden ungenießbargemachten Kanzleistils herabgerollte Definition, welche das menschliche Sprachgefühl verletzende Wirkung zu erzeugen fähig sind, liefert".

An Fastnacht könnte man Schülern die Scherzaufgabe stellen, zu definieren, was Schule ist, um dann die lustigste Antwort von einer Jury prämieren zu lassen.

Wenn dann in der nächsten Stunde nach Fastnacht die wissenschaftliche Nominaldefinition behandelt wird, ist das Kontrastprogramm perfekt und der Spaß leider vorbei.

Eine Nominaldefinition kommt dadurch zustande, dass einer Gruppe bereits eingeführter Wörter (auf der rechten Seite einer Definition) ein bislang unbekanntes Wort gleichgesetzt wird. $x\ \varepsilon\ Schimmel =_{def} x\ \varepsilon\ weißes\ Pferd$ ist eine bekannte Nominaldefinition des Prädikators *Schimmel*, die den Zweck hat, ein einfacheres Wort bereitzustellen. Nominaldefinitionen haben im Kern einen sprachökonomischen Sinn. Das wird sehr deutlich, wenn eine Definition von Prädikatoren allgemein eingeführt wird: $x\ \varepsilon\ P =_{def} A(x)$, wobei A(x) beliebig komplex aufgebaut sein kann. Ein einfaches Beispiel: $x\ \varepsilon$ Onkel$=_{def} \vee_y$ (y ist Geschwisterkind von $x \wedge x\ \varepsilon$ männlich), mit $\vee_y$ als dem großen Adjunktor oder Existenzquantor, der die Variable y bindet, so dass nur x frei vorkommt.

Kennzeichnungsdefinitionen vom Schema $u =_{def} \iota_x A(x)$ sind ebenfalls Nominaldefinitionen. Wie unterscheiden die sich nun von den vorgetragenen Realdefinitionen? Eine Nominaldefinition muss geordnet erfolgen, alle Prädikatoren, die auf der rechten Seite als Definiens erscheinen, müssen bereits vorher eingeführt sein. Es darf nicht zirkelhaft definiert werden, d.h. ein zu definierendes Wort darf selbst nicht zur Definition herangezogen werden. Und als drittes Merkmal: Die Aussageform A(x) muss explizit (ausdrücklich) ein Objekt x definieren. Implizite Definitionen, in denen behauptet wird, dass A(x) auch ein Objekt y definiert, werden ausgeschossen. Das ist in der Informatik nicht durchweg der Fall. Bei der algebraischen Spezifikation von abstrakten Datentypen wird axiomatisch, d.h. implizit (einschließend) definiert. Aber dieses Lehrstück steht weit außerhalb einer Informatik als Grundbildung, die im Konstruktiven verhaftet bleibt.

Literatur

1. Inhetveen, R.: Logik. Eine dialog-orientierte Einführung. Leipzig 2003
2. Lorenzen, P.: Lehrbuch der konstruktiven Wissenschaftstheorie. Stuttgart 2000
3. Kamlah, W., Lorenzen, P.: Logische Propädeutik, Vorschule des vernünftigen Redens. 3. Aufl. Stuttgart 1996
4. Seiffert, H.: Einführung in die Wissenschaftstheorie 1. München 1996

Informatik als Grundbildung

Teil IV: Objektsprache/Metasprache

H. Wedekind · E. Ortner · R. Inhetveen

Das Thema „Objektsprache/Metasprache" hat mit dem im Teil III behandelten Thema „Gleichheit und Abstraktion" von der Sache her nichts zu tun. Nur bei der bildlichen Veranschaulichung wird in beiden Teilgebieten eine Stufen- bzw. Ebenen-Metaphorik gewählt, die nicht nur bei Schülern zu der falschen Vorstellung führen kann, dass beide Gebiete zumindest ähnlich sind. Das ist nicht der Fall. Teil IV ist als Kontrastprogramm zum Teil III aufzufassen. Metaphorik – so wichtig sie im Schulunterricht ist – kann eine Täuschung herbeiführen, der man auch unterliegt, wenn man z. B. Eiweiß nur im Weißen eines Eis sucht.

Objektstufe/ Metastufe

In einer Objektsprache sprechen wir über Gegenstände unserer Welt. Zum Beispiel: *Konstanz liegt am Lago Maggiore*. *Konstanz* und *Lago Maggiore* sind Eigennamen und benennen Gegenstände unserer Welt. *Liegen_am* ist ein zweistelliger Prädikator, der die benannten Gegenstände in Beziehung setzt. Durch Sprache wird unsere Welt erschlossen, die wir durch Ausprägungen von Eigennamen und Prädikatoren gliedern. Sprache und Welt bilden eine anthropologische Grundlage. Das Erstaunliche und Außergewöhnliche an unserem Sprechen ist, dass wir nicht nur über die Welt sprechen, sondern auch über das Sprechen sprechen können. Wir sprechen zwanglos *über* (gr. meta) den vorangestellten Satz „Konstanz liegt am Lago Maggiore" und behaupten von dieser Wortfolge, sie sei falsch, in welcher schriftlichen Ausprägung sie auch immer hingeschrieben sein mag. Wir können in gleicher Weise über einzelne Wörter reden. Wenn wir sagen *Peter ist kurz*, dann meinen wir eine Person mit Namen *Peter*, die von kleinem Wuchs ist. Sagen wir aber: *„Peter" ist kurz*, dann wollen wir sagen, dass das Wort *„Peter"*, genauer sein Wortschema *kurz* ist.

Durch Einführung der Anführungszeichen wird seit Frege deutlich gemacht, dass über ein Schema, ein Sprachkonstrukt, geredet werden soll. Der Schritt, den wir durch ein Sprechen über sprachliche Objekte vollziehen, ist beachtlich, weil wir Sprache nicht mehr bloß gebrauchen (engl. „to use"), sondern Sprache durch Markieren in Anführungszeichen auch erwähnen (engl. „to mention").

In *Peter ist kurz* wird *Peter* gebraucht, in *„Peter" ist kurz* wird eine Anführung, nämlich *„Peter"*, gebraucht, aber ein männlicher Vorname erwähnt.

Durch die wichtigen Anführungszeichen wird ein Name, der auch im Falle „Konstanz liegt am Lago Maggiore" als ein ganzer Satz auftreten kann, von einer Objektsprache auf einer Objektstufe zu einem Namen einer Metasprache auf einer Metastufe. Statt Objektsprache sagen die Engländer auch *use language* (Gebrauchssprache) und statt Metasprache fällt dann im Englischen sehr zutreffend der Ausdruck *mention language* (Erwähnungssprache). Wir schreiben:

Objektstufe

- Konstanz liegt am Lago Maggiore bzw. Konstanz, Lago Maggiore ε liegen_am.
- Peter ε kurz.

DOI 10.1007/s00287-004-0432-7

Prof. Dr. H. Wedekind · R. Inhetveen
Institut für Informatik,
Universität Erlangen-Nürnberg,
Erlangen

E. Ortner
Fachbereich Rechts- und Wirtschaftwissenschaften,
TU Darmstadt,
Darmstadt

Prof. Dr. H. Wedekind
Institut für Informatik,
Universität Erlangen-Nürnberg,
Martensstraße 3, 91058 Erlangen
E-Mail: wedekind@informatik.uni-erlangen.de

Metastufe

- „Konstanz liegt am Lago Maggiore" ε falsch.
- „Peter" ε kurz.

„Falsch" und „kurz" werden Prädikatoren auf der Metastufe genannt. Man erkennt, dass jetzt auf einer Meta-Metastufe oder der 3. Sprachstufe die folgenden Sätze hingeschrieben werden dürfen:

Meta-Metastufe

„falsch" ε Prädikator und „kurz" ε Prädikator.

Auf der 4. Sprachstufe, der Meta-Meta-Metastufe, ist dann das Ende der Fahnenstange erreicht.

Meta-Meta-Metastufe

„Prädikator" ε Prädikator.

Wenn ein Prädikator auf sich selbst zutrifft, nennt man dieses Phänomen eine Selbstbeschreibung oder Autologie. Eine Nicht-Selbstbeschreibung heißt dann Heterologie. Wir können das Phänomen der Selbstbeschreibung auch schon auf der 2. Sprachstufe, der Metastufe hinschreiben. Es gilt: *„kurz" ε kurz.* Gleiches trifft im Deutschen für das Wort „dreisilbig" zu, d. h.: *„drei.sil.big" ε dreisilbig.* Wir können auch schreiben: *„drei.sil.big" ε autologisch.* Hingegen gilt im Deutschen: *„vier.sil.big" ε heterologisch.* Im Englischen heißt es aber: *„four.syl.la.bic" ε autologic.* Man erkennt an diesen Beispielen, dass Autologie bei Wörtern der normalen Sprache ein rein empirisches Phänomen ist. Das ist nicht so bei Wörtern einer Grammatik, in der Autologie absichtlich herbeigeführt werden kann. Zum Beispiel: Das Schema eines Elementarsatzes: *Nominator ε Prädikator* lässt sofort auf einer höheren Sprachstufe die Autologien zu: *„Nominator" ε Nominator* und *„Prädikator" ε Prädikator*. Auch: *„Deutscher Ausdruck" ε Deutscher Ausdruck* ist eine Autologie, die eine empirische Grammatik erlaubt.

Man kann den Übergang von einer Objekt- zu einer Metastufe auch operativ erklären und so den Gegensatz zur Abstraktion deutlich herausstellen. Dazu wird der Begriff „Funktor" herangezogen, der von dem schon mehrfach zitierten Rudolf Carnap (1891–1970) in Logik und Mathematik eingeführt wurde. Man stellt einen Funktor im Allgemeinen schematisch durch das Funktionssymbol *f* dar. In einem speziellen Gebrauch transformiert ein Funktor ein oder mehrer Wörter in einen Namen und wird dann ein namenbildender Funktor (nbf) genannt (s. [1], Stichwort „use and mention", Bd. 4). Das Wort *Mond* wird z. B. durch einen nbf zu einem Namen : „Mond"=nbf (Mond). Wenn ein Name auf einer Metastufe verfügbar ist, dann kann auch ein Satz auf dieser Stufe gebildet werden, falls ein geeigneter Meta-Prädikator gefunden wird, z. B.: „Mond" ε vierbuchstabig. Auch gilt: „Der Mond ist aufgegangen" = *f* (Der Mond ist aufgegangen), um den Meta-Satz zu prägen: „Der Mond ist aufgegangen" *is an initial verse of a famous German poem by Matthias Claudius.* Im Deutschunterricht für Ausländer kann so z. B. Englisch als Metasprache eingesetzt werden, während Deutsch als Objektsprache fungiert.

Die Wirkungsweise eines Funktors kann auch leicht am Beispiel *f* (2, 3) gezeigt werden. Man ersetzt z. B. *f* durch das Additionszeichen ‚+' und schreibt + (2, 3) oder in Infix-Notation (2+3). Noch ein weiteres Beispiel: Ein sehr einprägsamer Funktor wird von Carnap selber angegeben. Er nennt den Funktor x-te, definiert auf einer Folge. Man schreibt z. B.: Mittwoch = 3-te (Wochentag).

Was Meta-Prädikatoren anbetrifft, so können wir bei Wörtern einer Grammatik sicher entscheiden, ob ein Meta-Prädikator vorliegt. „Adjektiv" ist z. B. ein Wort einer empirischen Grammatik, und sofort kann der Beispielsatz: *„schön" ist ein Adjektiv* gebildet werden. Auch werden in der Logik die Wahrheitswerte ‚wahr' und ‚falsch' als Meta-Prädikatoren eingeführt, da mit ihnen ein Aussagesatz beurteilt wird. Bei anderen Prädikatoren ist die Zuordnung nicht so klar. Es hängt von der Verwendung ab. Manchmal wird ein Prädikator einer Objektstufe gleich auf einer Metastufe als Meta-Prädikator wieder verwendet, im Beispiel der Prädikator „kurz". Es ist sogar für Wissenschaftler erstaunlich zu erfahren, dass einige ihrer Prädikatoren, man nennt sie auch wegen ihrer strengen Einführung „Termini" (daher das Wort „Terminologie"), in anderen Disziplinen als Meta-Prädikatoren Eingang finden. So ist z. B. der Terminus „erblich" oder „vererbbar" (engl. „inheritable") in Jurisprudenz und Biologie objektsprachlicher Natur. Seit der berühmten „Begriffsschrift" von Frege (1879) wird „erblich" in Logik und Mathematik metasprachlich verwendet (s. auch [1], Stichwort „erblich", Bd. 1). Wie könnte es anders sein: Die Informatik folgt der Logik und Mathematik.

Metasprache im Einsatz

Im Zeitalter der Kommunikation in Rechnernetzen haben Sprachen zur Beschreibung und zum Austausch von Datenschemata eine zentrale Bedeutung erlangt. Kommunikationspartner bedürfen eines gemeinsamen Schemas. Das Beschreiben eines Schemas ist ein Reden über ein Schema, das man seinem Kommunikationspartner zur Verfügung stellt. Das Übermitteln des Schemas geht dem Verschicken von Ausprägungen voraus. Aus dem Aspekt der Netze ist die Frage nach den Programmiersprachen relativ uninteressant. Im Vordergrund stehen Schemabeschreibungssprachen als Metasprachen, in denen die Grammatik der Kommunikation festgelegt wird. Unter diesen Sprachen hat XML (eXtensible Markup Language) eine herausragende Stellung eingenommen. Wir wollen uns auf diese Sprache beschränken.

Bevor wir die Metasprache von XML im Einsatz zeigen, ist es aus methodischen Gründen erforderlich, zunächst auf einer Objektstufe ein Beispiel in XML anzugeben, dessen Struktur dann auf einer Metastufe beschrieben wird. Wir wählen dazu unsere Artikelserie im Informatik Spektrum aus. Das ist nahe liegend.

Ein Dokument auf einer Objektstufe

```
<Artikelserie>
  <Titel>Informatik als Grundbildung</Titel>
  <Autor>
    <Name>Wedekind</Name>
    <Vorname>Hartmut</Vorname>
    <Mail>wedekind@informatik.uni-erlangen.de</Mail>
  </Autor>
  <Autor>
    <Name>Ortner</Name>
    <Vorname>Erich</Vorname>
    <Mail>ortner@winf.tu-darmstadt.de</Mail>
  </Autor>
  <Autor>
    <Name>Inhetveen</Name>
    <Vorname>Rüdiger</Vorname>
    <Mail>inhetveen@informatik.uni-erlangen.de</Mail>
  </Autor>
  <Artikel status="veröffentlicht">
    <Untertitel>Schema und Ausprägung</Untertitel>
  </Artikel>

  <Artikel status="veröffentlicht">
    <Untertitel>Bildung von Elementarsätzen</Untertitel>
  </Artikel>
  <Artikel status="unveröffentlicht">
    <Untertitel>Namengebung und Kennzeichnung</Untertitel>
  </Artikel>
  ...
</Artikelserie>
```

Die Artikelserie besteht nach dieser Darstellung aus einem *Titel*, mehreren *Autoren* und mehreren *Artikeln*. Die Autoren werden durch Name, Vorname und Mail beschrieben. Einen Artikel kennzeichnet ein Untertitel und ein Status: veröffentlicht oder unveröffentlicht.

Ein Dokument auf einer Metastufe

Tabelle 1 zeigt, wie ein solches Dokument aussieht.

Die Metasprache von XML, sie wird DTD (Document Type Definition) genannt, ist ersichtlich rekonstruktionsbedürftig. Der nichts sagende Terminus *Element* ist zunächst einmal rational-grammatisch durch den Terminus *Eigenprädikator* zu ersetzen. Das Wort ATTLIST zeigt eine Liste von Apprädikatoren an. ELEMENT bzw. Eigenprädikator und ATTLIST bzw. Apprädikator sind aus der Sicht des Rechners Graphemsequenzen (Schriftzeichenfolgen). Für den Rechner könnte man aber auch Tralala und Trololo schreiben. Nur Unterschiedlichkeit wird verlangt. Rational rekonstruiert sieht die erste Zeile aber wie folgt aus: *„Artikelserie“ ε Eigenprädikator* mit den Teilschemata: *„Titel“* $\leq$ *„Artikelserie“*, *„Autor$^+$“* $\leq$ *„Artikelserie“*, *„Artikel$^+$“* $\leq$ *„Artikelserie“*. Das Zeichen „ $\leq$ “ stellt die Relation „Teil–Ganzes“

Tabelle 1

```
<!ELEMENT    Artikelserie    (Titel, Autor+, Artikel+)
<!ELEMENT    Titel           #PCDATA>
<!ELEMENT    Autor           (Name, Vorname, Mail)>
<!ELEMENT    Name            #PCDATA>
<!ELEMENT    Vorname         #PCDATA>
<!ELEMENT    Mail            #PCDATA>
<!ELEMENT    Artikel         (Untertitel)>
<!ATTLIST    Artikel
             status          (veröffentlicht|unveröffentlicht)    "veröffentlicht">
<!ELEMENT    Untertitel      #PCDATA>
```

Dokument auf einer Metastufe

dar. Nicht weiter strukturierte Eigenprädikatoren wie im Fall *Titel* in der zweiten Zeile werden durch das Zusprechen eines Datentyps gekennzeichnet, hier PCDATA (Parsed Character Data). Rekonstruiert: „*Titel*“ *ε PCDATA*. Die vorletzte Zeile rekonstruiert sich wie folgt: „status“ ε Apprädikator, mit „*veröffentlicht*“ ***instance_of*** „*status*“ oder „*unveröffentlicht*“ ***instance_of*** „*status*“. Das in der vorletzten Zeile angeführte „veröffentlicht“ bedeutet, dass *veröffentlicht* ein *default value* (Standardvoreinstellung) ist und somit ersatzweise zum Zuge kommt, wenn bzgl. Artikelstatus nichts ausgesagt wird.

Mit der vorgestellten DTD wird eine Objektsprache zur Beschreibung von Artikelserien vom Typ der laufenden Serie festgelegt. Auf einer Metastufe wird ein Regelwerk, eine Syntax, also eine Grammatik vorgegeben. Widerspricht ein XML-Dokument keiner der in der Metasprache spezifizierten Regeln, so wird es als gültig bezeichnet.

Aus unserer kurzen Darstellung dürfte deutlich hervorgegangen sein, dass Sprachen wie XML nur erlernbar sind, wenn die Lehre von der Objekt- und Metasprache vorangestellt wird.

Modelle

Wir können in dieser kurzen Abhandlung den Modellbegriff nicht in seiner vielfältigen Verwendung darstellen. Wir verweisen auf [3] und die dort angegebene Literatur. Für unsere Zwecke reicht es, Modelle als Beschreibung (description) aufzufassen. Das klingt zwar sehr allgemein, aber immerhin sind Präskriptionen (prescriptions, Vorschriften, Verordnungen etc.) ausgeschlossen. Ethik und Moral, Normen, Sitte und Anstand bleiben bei diesem Modellbegriff außen vor, falls sie nicht in Beschreibungen umgesetzt werden können. Auch Erklärungen, die als Subsumtion eines Einzelfalls unter ein Gesetz aufzufassen sind, stehen nicht zur Debatte.

Modelle mit einem Anspruch auf eine hohe Beschreibungsgenauigkeit spielen bei den „Hauptkunden“ der Informatik, nämlich Naturwissenschaft und Technik, Wirtschaftswissenschaft und Medizin, eine herausragende Rolle. Je nach Forschungsprogramm einer Disziplin bemüht man sich um eine umfassende Beschreibung eines Sachverhalts. An dieser Beschreibung, die sich z. B. in Protokollen von Experimenten oder Feldbeobachtungen niederschlägt, werden bestimmte Veränderungen, sog. „Idealisierungen“ vorgenommen: Werteverläufe werden geglättet, unterschiedliche Ausprägungen eines Merkmals durch den Mittelwert ersetzt, bestimmte Einflüsse als „vernachlässigbar“ unterdrückt, um so eine übersichtliche Darstellung eines Sachverhaltes zu gewinnen. Das Ergebnis einer solchen Idealisierung ist ein Idealmodell. $\ddot{x} = g$, mit g = 9,81 [m/s^2], die Beschreibung des freien Falls, ist ein solches Idealmodell. Technische Idealmodelle beschreiben einen technischen Aufbau oder Ablauf unter der Vorgabe, dass alle Bestandteile ihren technischen Funktionsbestimmungen in idealer Weise gehorchen. Konstruktionspläne oder Ablaufplanungen sind typische Beispiele für Idealmodelle von Gebäuden, Geräten und Abläufen [3].

Es gibt eine hübsche Geschichte zur Frage, was denn nun die typischen Modelle der Informatik sein könnten. Als Kurzfassung ist die Geschichte wie folgt erzählt: Amerikanische Quellen verkünden, dass das Kind „Informatik“ 1947 als Computer Science mit der Erfindung des Transistors und mit der Gründung der *Association for Computing Machinery (ACM)* gezeugt wurde. Das Lebenslicht erblickte das Geschöpf nach diesen Quellen als akademisches Fach erst 1968, als in den *Communications of the ACM* ein akademisches Studium des neuen Fachs beschrieben wurde. Damals waren schon alle denkbaren Modellarten erfunden. Als das arme Neugeborene an die Türen der klassischen Fachdisziplinen klopfte, um zu erfahren, welche Modellart noch übrig sei, sagten diese samt und sonders: „Die Objektwelt auf der Objektstufe ist unsere Welt. Geh dahin, wo der Pfeffer wächst“, und knallten ihre Türen zu. Nur eine Wissenschaft war da, die sagte: „Komm doch zu mir, bei mir bist du gut aufgehoben. Ich sitze zwar nicht bei den Königen der Objektwelt auf der Objektstufe. Weil ich Form und Schema um ihrer selbst willen schon liebe, sitze ich auf der Metastufe als Diener der hohen Herrn da unten. Wenn die mich nicht beschäftigen, arbeite ich an mir selbst. Verzeihung, ich habe mich noch gar nicht vorgestellt. Ich habe einen ganz modernen Doppelnamen, bin aber uralt. Ich heiße Mathematik-Logik. Mein zweiter Name zeigt an, dass ich bei den Sprachwissenschaften eine Menge Verwandte habe. Die alle sind – wie du auch – Symbolverarbeiter.“ Ende der Geschichte – mit der klassischen Bemerkung: Und wenn sie nicht gestorben sind, dann leben sie noch heute.

Die Unterstützung der Objektstufe als „Dienerin“ ist zweifelsohne auch eine wichtige Aufgabe

Tabelle 2

Objektstufe:		
Schema:	*Instantiierung:*	
Hund (<u>Name</u>, Farbe)	Fido	**instance_of** Name,
	braun	**instance_of** Farbe,
	<Fido, braun>	**instance_of** Hund.
Metastufe:		
Schema:	*Instantiierung:*	
R (N, A)	"Name"	**instance_of** N
	"Farbe"	**instance_of** A
	"Hund"	**instance_of** R

Relationen-Modell

der Informatik. Die Modelle dieser Stufe müssen so umgeschrieben werden, dass sie mit Hilfe eines Computers dargestellt und bearbeitet werden können. Genauer heißt das: Es müssen effektive Prozeduren angegeben werden. Wenn die Modelle sich jedoch nicht unmittelbar in die Sprache der Informatik umsetzen lassen, muss das Ausgangsmodell zunächst logisch rekonstruiert werden, um den Schema-Bedingungen der Informatik zu genügen. Neben dieser dienenden Funktion hat die Informatik wie Mathematik und Logik die Aufgabe, an sich selbst zu arbeiten, d. h., unabhängig von den vielfältigen Erscheinungsformen auf der Objektstufe eigene Beschreibungsmöglichkeiten oder Modelle auf der Metastufe zu entwickeln. In dem Bilde von *use* und *mention* ist die Informatik nicht nur eine Gebrauchs-, sondern auch eine Erwähnungswissenschaft. Informatik-Modelle sind Meta-Schemata, „zwangsweise", da der Informatik der originäre Zugang zur Objektebene verwehrt wurde, wie uns die Geschichte nach ihrer Geburt lehrt. Berühmtheiten auf der Modellpalette sind: das Relationen-Modell, das Objekt-Modell der objektorientierten Programmierung, das Modell der Begrenzungsflächen-Darstellung (boundary representation), das Modell „Constructive Solid Geometry"(CSG) usw., nicht zu vergessen die vielen formalen Grammatiken (Metasprachen) als Sprachbeschreibungen (Sprachmodelle) des Fachs.

Wir wenden uns 2 Modellen auf der Metastufe zu. Es handelt sich um das Relationen-Modell der Datenbanksysteme und um das Objekt-Modell der objektorientierten Programmierung. Zu beachten ist, dass einer Beschreibung auf einer Metastufe immer eine Beschreibung auf der Objektstufe methodisch voranzugehen hat. Metastufen bleiben ohne Objektstufen unverständlich, ein Rat, der von jeder grammatischen Darstellung einer empirischen Sprache befolgt wird.

Relationen-Modell

Das Relationen-Modell ist in Tabelle 2 dargestellt.

R, N, und A sind schematische Buchstaben. Sie stehen für den Relationen-Namen, auch Relator oder Eigenprädikator genannt (R), für den Primärschlüssel, auch Nominator genannt (N), für sonstige Attribute, auch Apprädikatoren genannt (A). Name, Farbe, und Hund werden auf der Metastufe erwähnt und nicht gebraucht und sind deshalb mit Anführungszeichen versehen.

Das Modell der objektorientierten Programmierung auf einer Metastufe darzustellen ist wesentlich komplizierter. Es ist deshalb eine offene Frage, ob dieser Gegenstand in den Katalog von Gegenständen einer Grundbildung aufgenommen werden sollte.

Objekt-Modell der objektorientierten Programmierung

Objektstufe

Als Programm-Schema wählen wir das Schwingen eines Pendels (engl. pendulum) in einer bekannten Syntax:

Schema

```
Class Pendulum implements Oscillation, Properties {
  float mass;
  float length;
  int cycles;
  float position (float time) {...
  }

  Pendulum (float, float, int) {...
  }


  }
interface Oscillation {
  float position (float time) ;
```

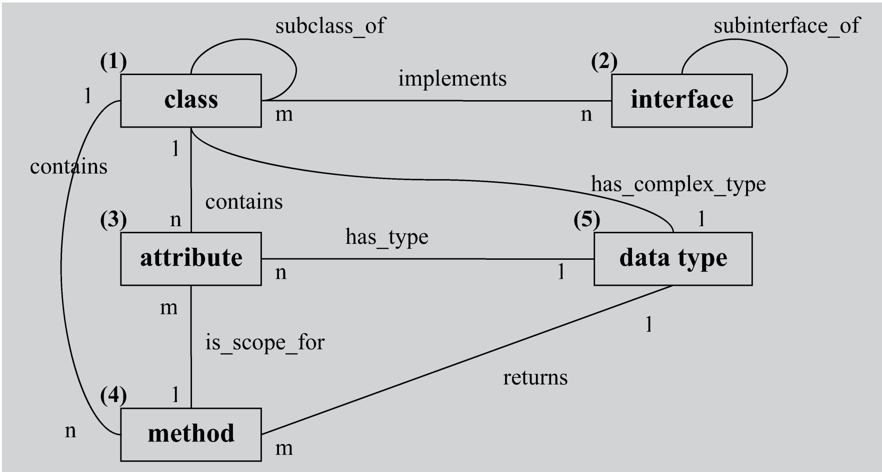

Abb. 1 Objektorientiertes Modell als Entity-Relationship-Diagramm

```
  float get_cycles ( ) ;
  }
interface Properties {
  float get_length () ;
  float get_mass () ;
}
```

Instantiierung

Eine Methode wie in unserem Fall *Pendulum ( )*, die denselben Namen wie *Class* hat, wird Konstruktor genannt. Aufgabe eines Konstruktors ist es, nach einer Belegung der schematischen Parameter Ausprägungen zu erzeugen. Dass diese Ausprägungen in der Sprache der Informatik auch „Objekte" genannt werden, ist befremdlich und zeugt wiederum von einer terminologischen Hilflosigkeit des Fachs.

Wir instantiieren *Class* und die Methode *position ( )*: **Pendulum (10.0, 5.0, 1)**; **position (1.0)**, um zu erfahren, welche Position ein Pendel als Idealmodell mit der Masse 10.0 und der Länge 5.0 im ersten Zyklus nach 1.0 Zeiteinheit einnimmt.

Metastufe

Schema

In der Didaktik aber auch in den Wissenschaften werden Veranschaulichungsmodelle herangezogen, um einen Mangel an Anschaulichkeit zu heilen [3]. Wir zeigen das OO-Modell auf der Metastufe zunächst mit Hilfe einer Veranschaulichung durch ein Entity-Relationship-Diagramm (ER-Diagramm, Abb. 1).

Man erkennt, dass das OO-Modell wegen seiner Komplexität den Bereich „Informatik als Grundbildung" deutlich übersteigt. Gleiches gilt natürlich auch für die Erweiterung in der *Unified Modeling Language (UML)*. Ein späterer Ausbildungsgang „Informatik für Fortgeschrittene" möge sich dieser Lehrgegenstände annehmen. Eine jede Grundbildung hat Grenzen, die da liegen, wo ihre Verantwortlichkeit für andere Fächer verschwindet. Für interessierte zukünftige Naturwissenschaftler, Mediziner, Ingenieure und Wirtschaftler mag Objektorientierung als Thema attraktiv sein. Allen anderen, die nicht zur Informatik hin tendieren, genügen einfache Kenntnisse über Objekt- und Metasprache.

Wir verkürzen deshalb diesen Abschnitt und beschließen ihn mit dem Hinweis, dass ein bloß veranschaulichendes ER-Diagramm auch ohne Schwierigkeiten mit Hilfe des Relationen-Modells dargestellt werden kann. Wir zeigen nur die ersten 3 Relationen.

```
class        (ClassName,....,)
interface    (InterfaceName,....,)
implements   (ClassName, InterfaceName,....,)
```

Instantiierung

„Pendulum" **instance_of** ClassName,
„Oscillation" **instance_of** InterfaceName usw.
Pendulum und Oscillation werden nur erwähnt und nicht gebraucht.

Paradoxien

Kuriositäten üben nicht nur auf Jugendliche eine ungeheure Anziehungskraft aus. Auf Jahrmärkten finden Veranstaltungen einen großen Zulauf, die Wundersames, Rätselhaftes oder viel Unerwartetes zu bieten haben. Das antike Schrifttum ist voll von der Rede über Paradoxien. Sophistische Großrhetoren standen zur Zeit des Aristoteles auf der Agora, dem Markt der Athener, und demonstrierten den diskutierfreudigen Hörern anhand von Paradoxien, wie leicht sich in den Worten Wittgensteins der Verstand mit den Mitteln der Sprache verhexen lässt (Philosophische Untersuchungen, § 109). In der Schule wird auch heute manchmal die Paradoxie „Achilles und die Schildkröte" behandelt, die uns Aristoteles überliefert hat. Der schnellfüßige Achilles gibt beim Wettlauf einer Schildkröte großzügig einen Vorsprung. Bei gleichem Start muss Achilles erst den Vorsprung aufholen. In dieser Zeit ist die Schildkröte aber schon weiter. Das wiederholt sich. Der Vorsprung schmilzt zwar dahin, aber Achilles kann die Schildkröte nie erreichen, geschweige denn überholen. Absurdes wird auf einmal scheinbar wahr.

Eine Verhexung des Verstandes als Strafe dafür, weil Objekt- und Metasprache nicht sorgfältig auseinander gehalten werden, begegnet uns bei der berühmten Lügner-Paradoxie (engl. „liar paradox") der Antiken. Der Kreter Epimenides (Ende des 7. Jhd. v. Chr.) soll die folgende Aussage gemacht haben:

(1) Alle Kreter lügen immer.

Wenn (1) wahr ist, lügen alle Kreter immer, also auch der Kreter Epimenides mit seiner Aussage (1), also ist (1) falsch. Wenn (1) falsch ist, lügen nicht alle Kreter immer, also sagen einige Kreter manchmal die Wahrheit, also auch Epimenides mit seiner Aussage (1).

Der Argumentationsablauf wiederholt sich nun : Wenn (1) wahr ist, lügen alle Kreter immer, also auch Epimenides mit seiner Aussage (1), also ist (1) falsch. Wenn (1) falsch ist......

Das geht nun unaufhaltsam so weiter. Man spricht von oszillierenden Wahrheitswerten, weil (1) einmal wahr ist, dann wieder falsch, dann wieder wahr, usw. Man glaubt, in einem modernen Parlament zu sitzen. Wir fühlen uns von einem griechischen Sophisten und Großrhetor an der Nase herumgeführt, und die Zuhörer auf der Tribüne haben an der Verhexung unseres Verstandes ihr Vergnügen. Der Großrhetor mit seinen demagogischen Fähigkeiten ist der Star; ihm gegenüber scheint alles zu verblassen.

Die mittelalterliche Logik bezeichnete das Lügner-Paradox als „Insolubilia", als unlösbares Problem. Die Lösung des Problems wurde erst im 20. Jahrhundert möglich und ist mit den Namen Alfred Tarski (1901–1983) und Rudolf Carnap verbunden, die als Erste die strikte Unterscheidung von Objekt- und Metasprache in die logische Analyse nach Vorarbeiten von Frege, Wittgenstein und Russel einführten. Seiffert [2] nennt zu Recht die Unterscheidung eine wissenschaftliche Großtat ersten Ranges. Die Lösung der „Insolubilia" sieht wie folgt aus (wir folgen dabei Seiffert wegen seiner prägnanten Darstellung):

In dem Satz: „Der Kreter Epimenides sagt, alle Kreter lügen immer", werden Objekt- und Metasprache in unzulässigerweise miteinander vermengt. Der Satz sagt etwas über sich selbst aus. Das geht nicht. Ein Satz kann immer nur entweder über einen nichtsprachlichen Gegenstand (objektsprachlich) oder über einen anderen Satz (metasprachlich) etwas aussagen. Wenn daher ein Satz über einen Satz etwas aussagen soll, dann kann man das nicht sozusagen „in einem Arbeitsgang" erledigen, sondern dann muss man die Gesamtaussage in 2 Sätze zerlegen: in denjenigen, der etwas ausgesagt (alle Kreter lügen immer), und in denjenigen, der über (gr. meta) den anderen etwas aussagt (Ein Kreter sagt).

Wittgenstein hat das knapper formuliert (Tractatus 3.332): „Kein Satz kann etwas über sich selbst aussagen, ..."

Wenn wir also schreiben: Ein Kreter sagt „alle Kreter lügen immer", dann gilt der Objektsatz in Anführungszeichen, oder er gilt nicht, was ein Problem der Geltungssicherung ist (Teil VI). Der Sachverhalt „alle Kreter lügen immer" ist wirklich, dann gilt das so, oder er ist fiktiv, dann gilt das eben nicht so. Ob ein Kreter, ein Römer, ein Richter, ein moderner Historiker oder wer auch immer die Sachverhaltsprüfung vornimmt, ist unerheblich. Die Prüfung der geschichtlichen Quellen möge dazu führen, dass gesagt werden kann: Die Aussage „Epimenides sagte einmal ‚alle Kreter sind Lügner'" ist wahr. Die Geltungssicherung der Metaaussage, die in einer umrahmenden Meta-Metaaussage festgestellt wird, steht auf einem ganz anderen Blatt und ist die Angelegenheit einer historischen Forschung.

Paradoxien wie die vom Lügner, etwa auch „Dieser Satz ist falsch" oder „Ich lüge jetzt", lassen sich auf das folgende Schema bringen (s. [1], Stichwort „Lügner-Paradoxie", Bd. 2):

(1) (1) ist falsch.

Man spricht von Selbstbezug, da der innere Sachverhalt (1) sich auf den äußeren (1) bezieht und umgekehrt, und von Verneinung.

Dass Selbstbezug und Verneinung und ihre Auflösung im Falle des Epimenides schon in der

Grundbildung demonstriert werden, ist anzustreben. Wegen der Kuriosität ist ein großes Interesse der Jugendlichen zu erwarten. Darüber hinaus gelingt so ein beeindruckender Einstieg in das, was man logische Analyse der Sprache nennt.

Der „große Bruder", das Halteproblem der Informatik, sollte in einer Grundbildung außen vor bleiben. Das Halteproblem lautet in Kurzform: Gibt es ein Programm, das von jedem Programm feststellt, ob es anhält oder nicht? (Es gibt kein solches Programm!) Die Lügner-Paradoxien sind aber eine geeignete Propädeutik für die mit dem Halteproblem aufkommenden Fragen nach Berechenbarkeit und Entscheidbarkeit in späteren Ausbildungsgängen.

Literatur

1. Mittelstraß, J. (Hrsg.): Enzyklopädie, Philosophie und Wissenschaftstheorie, Bd. 1–4. Stuttgart: Metzler 1980–1996
2. Seiffert, H.: Einführung in die Wissenschaftstheorie 1. München: C.H. Beck 1996
3. Wedekind, H., Görz, G., Kötter, R., Inhetveen, R.: Modellierung, Simulation, Visualisierung: Zu den aktuellen Aufgaben der Informatik. Informatik Spektrum 21, 265–272 (1998)

Informatik als Grundbildung

Teil VI: Logik und Geltungssicherung

H. Wedekind · R. Inhetveen · E. Ortner

Mit diesem abschließenden Teil VI über Logik und Geltungssicherung wird zwar eine kleine Aufsatzreihe beendet, nicht aber – so ist zu hoffen – die Diskussion über Informatik als Grundbildung. ...

Logik und Sprache

Mit unserer natürlichen Sprache teilt die Logik das Schicksal, gewissermaßen wildwüchsig erworben zu werden. Dass dieser Erwerb der korrigierenden und fördernden Ergänzung durch die Schule bedarf, ist im Fall der Sprache eine Selbstverständlichkeit, im Fall der Logik jedoch nicht auf der Agenda. Orthographie ist Thema, Orthonoetik nicht. Die von Bacon (1561–1626; „Die Menschen glauben, ihr Verstand gebiete den Worten; es kommt aber auch vor, dass die Worte ihre Kraft gegen den Verstand umkehren". [Credunt enim homines rationem suam verbis imperare; sed fit etiam ut verba vim suam super intellectum retorqueant et reflectant]) über Leibniz (1646–1716; „Die Alltagssprachen, obgleich sie meist für das schlussfolgernde Denken von Nutzen sind, sind doch unzähligen Zweideutigkeiten unterworfen") bis Wittgenstein (1898–1951; „Die Philosophie ist ein Kampf gegen die Verhexung unseres Verstandes durch die Mittel unserer Sprache") immer wiederholten Warnungen blieben in der Bildungsdiskussion ohne Wirkung. Das macht es schwer, die Logik als Teil einer Grundbildung zu etablieren.

Dazu kommt ein zweiter Grund: Die Logik lädt in ihrer üblichen Darstellung nicht gerade zu einer näheren Beschäftigung mit ihr ein. Führende Logiker sprachen und sprechen zwar vom „natürlichen Schließen", ein Blick in ein Logiklehrbuch zeigt aber in der Regel Zeichen und Formeln, die alles andere als „natürlich" aussehen. Dazu kommt eine Reihe von „Prinzipien", die man als plausibel, selbstverständlich, fraglos gültig o. ä. zur Kenntnis zu nehmen hat. Ein solches Prinzip ist das so genannte *Zweiwertigkeitsprinzip*. Man hat jahrtausendelang (nämlich seit der *Stoa*) geglaubt, es etwa folgendermaßen formulieren zu müssen: Eine Aussage hat einen von zwei Wahrheitswerten. Dieses Prinzip wurde dann präzisiert durch die beiden Sätze: Eine Aussage ist nicht zugleich wahr und falsch (Prinzip vom ausgeschlossenen Widerspruch); und: eine Aussage ist entweder wahr oder sie ist falsch. Ein Drittes gibt es nicht. Der letzte Satz ist als „tertium non datur" ins Gedächtnis der Logiker eingegangen. Ob Sätze dieser Art wahre Erfahrungssätze, Vorschriften für die Rede von „wahr" und „falsch" oder (implizite?) Definitionen von „Aussage" sind, wird in aller Regel nicht einmal gefragt. Auch dass die Rede von einem „Prinzip" zumindest in den Formalwissenschaften Mathemtik und Logik meist nur bedeutet, dass da ein metametasprachlicher Satz, also ein Satz der Prädikatenlogik zweiter Stufe, vorliegt, wird (warum wohl?) im Unterricht an Schule und Hochschule nur selten gesagt. Daher fällt es dem Anfänger zwar nicht allzu schwer, das Sprachspiel „Logik" zu lernen, wohl aber, es zu verstehen.

DOI 10.1007/s00287-004-0452-3

H. Wedekind
Universität Erlangen-Nürnberg,
Institut für Informatik,
E-Mail: wedekind@informatik.uni-erlangen.de

E. Ortner · R. Inhetveen
TU-Darmstadt,
Fachbereich Recht- und Wirtschaftswissenschaften

Logik und Informatik

Für die Informatik schien das alles am Anfang nicht so wichtig: Um seinen (Schalt-)*Plankalkül* für die Schaltung von Gattern zu entwickeln, genügte für Zuse (1910–1995) die klassische Aussagenlogik mit den als „Schalter geschlossen“ bzw. „Schalter offen“ interpretierten Wahrheitswerten. Auf der Seite der Hardware ist das heute noch so: die klassische Aussagenlogik genügt. Auf der Seite der Software sieht die Sache mittlerweile freilich ganz anders aus. Der längst vollzogene Übergang zur (klassischen) Prädikatenlogik (erster Stufe) liefert heute nicht mehr all die Mittel, derer die Informatik bedarf. Nicht zuletzt das Auftauchen immer neuer Nicht-Standard-Logiken (nichtmonoton, parakonsistent, fuzzy, mehrsortig, temporal, substruktural …) ist dafür ein deutliches Indiz. Wie geht man damit um? Gar nicht? Oder ist das nur Sache von Spezialisten und somit kein Thema einer Grundbildung?

Ein Thema für letztere ist sicherlich, wie man durch das Einbringen eines *Ordnungsgedankens* in ein „Durcheinander“ zu einer Struktur kommt. Das ist ein *allgemeines Verfahren*, das in konkreter Form an den unterschiedlichsten Stellen auftritt. Es ausdrücklich als solches zu benennen, bringt ein reflexives Moment in die Grundbildung, bei dem sich die Frage, wofür man das brauchen könne, von selbst erledigt. In der Informatik begegnet man diesem Thema etwa bei der Variantenbildung. Doch auch in der (Schul)mathematik taucht das Thema auf, z. B. bei der Rede von Kegelschnitten als Sammelbezeichnung für auf den ersten Blick doch recht unterschiedliche ebene Kurven. Im Fall der Logik(en) ist die Idee der Strukturierung durch Ordnungsgesichtspunkte nur in Ansätzen wenigstens ein Forschungsthema.

Eine Alternative

Nun hat seit Beginn der 60er Jahre des vorigen Jahrhunderts ein neuer Zugang zur Logik von sich reden gemacht: einer, für den das Argumentieren um die Geltung einer Behauptung in Form eines Dialogs den Ausgangspunkt bildet. Dieser „approach“ hält am Zweiwertigkeitsprinzip fest, aber in einer abgeschwächten Form: Es gibt von den (beiden) Dialogpartnern am Ende immer genau einen *Gewinner* und einen *Verlierer*. Es ist ganz wichtig sich klar zu machen, dass damit keinerlei Entscheidung für eine bestimmte Logik, etwa die klassische oder die konstruktive (früher „intuitionistisch“ genannte) Logik, verbunden ist. Mehr noch: Beim Aufbau der Logik kommt man zunächst ganz ohne Wahrheitswerte aus.

Sehen wir uns also etwas näher an, wie ein dialogorienterter Zugang zur Logik aussieht. Er beginnt mit der Vorstellung einer *gemeinsamen Geltungskontrolle* von *Behauptungen*; dabei sind verteilte Rollen einzunehmen: wer eine Behauptung vorträgt, der so genannte *Proponent*, $\mathcal{P}$, ist nicht in jedem Fall verpflichtet, eine (zusammengesetze) Behauptung in allen ihren Teilen gegen einen Zweifler, den sogenannten *Oppponenten*, $\mathcal{O}$, zu verteidigen. $\mathcal{P}$ kann die Begründung eigener Teilbehauptungen davon abhängig machen, dass $\mathcal{O}$ zuvor andere Teilbehauptungen nachweist. Wie die einzelnen Begründungspflichten aussehen, wird durch die (von $\mathcal{P}$ getroffene) Wahl der logischen Junktoren und Quantoren geregelt. Im einzelnen gelten die Angriffs- und Verteidigungsregeln in Tabelle 1.

Wie diese Tabelle zeigt, sind im Falle einer Subjunktion und einer Negation vom $\mathcal{O}$ eigene Behauptungen vorzutragen, um einen wirksamen Angriff zu formulieren. Im Fall der Subjunktion darf der $\mathcal{P}$ die erforderliche Verteidigung solange *zurückstellen*, bis ein von ihm selbst vorgetragener Gegenangriff abgewehrt ist. Im Fall der Negation hat der $\mathcal{P}$ selbst gar keine Verteidigung: ihm bleibt nur, die Behauptung des $\mathcal{O}$ zu widerlegen. Außer diesen Regeln für die logischen Operatoren werden noch einige naheliegende Dinge vereinbart: zum Beispiel, dass $\mathcal{P}$ und $\mathcal{O}$ abwechselnd ihre Argumente vortragen und dass, wer eine fällige Verteidigung nicht länger aufschieben kann, sie dann nachholen muss. Zu Einzelheiten vgl. [2].

Behauptung	Angriff $\mathcal{O}$	Verteidigung $\mathcal{P}$
$\mathbf{A} \wedge \mathbf{B}$	L ?	$\mathbf{A}$
$\mathbf{A} \wedge \mathbf{B}$	R ?	$\mathbf{B}$
$\mathbf{A} \vee \mathbf{B}$	?	$\mathbf{A}$ | $\mathbf{B}$
$\mathbf{A} \rightarrow \mathbf{B}$	$\mathbf{A}$?	$\mathbf{B}$
$\neg\, \mathbf{A}$	$\mathbf{A}$?	–
$\bigwedge_x \mathbf{A}(x)$	x_0?	$\mathbf{A}\ (x_0)$
$\bigvee_x \mathbf{A}(x)$	?	$\mathbf{A}\ (x_0)$

Tabelle 1

	$\mathcal{O}$	$\mathcal{P}$
0		$\neg(\mathbf{A} \wedge \neg\mathbf{A})$
1	$\mathbf{A} \wedge \neg\mathbf{A}$?	–
2	$\mathbf{A}$	L ?
3	$\neg\mathbf{A}$	R ?
4	–	$\mathbf{A}$?
5	†	✓

Tabelle 2

Definitheit und Wahrheit

Ein Dialog endet, sobald einer der beiden Partner kein Argument mehr vorbringen kann: er hat dann *verloren*, der Partner hat *gewonnen*. Da man aus verschiedenen Gründen verlieren kann (durch Aufgeben, Ungeschicklichkeit, weil man sich auf schlechte *Geltungsgründe* beruft oder weil sich „einfach nicht gewinnen lässt"), ist mit einem einzelnen Dialog über die „Wahrheit" einer Behauptung natürlich nicht entschieden. Eines aber steht von vornherein fest: Wenn die Ausgangsbehauptung syntaktisch korrekt gebildet war, endet jeder Dialog nach endlich vielen Schritten, und zwar mit einer Entscheidung darüber, wer gewonnen und wer verloren hat. Behauptungen mit dieser Eigenschaft – ein Dialog ist nach einer endlichen Schrittzahl entschieden – nennt man *dialogdefinit*.

Will man über diese „schwache" Variante des Bivalenzprinzips hinaus, so braucht man natürlich – schon um die „starke" Variante überhaupt formulieren zu können – einen Wahrheitsbegriff; man erhält ihn durch eine Definition, in der verlangt wird, eine Behauptung *gegen jeden denkbaren* Opponenten gewinnen zu können, damit sie „wahr" genannt werden darf. Genauer spricht man an dieser Stelle von *materialer Wahrheit* und sagt, ihr Vorliegen sei an die Existenz einer (materialen) Gewinnstrategie gebunden, die der Proponent nur zu befolgen braucht, um gegen jeden Opponenten zu gewinnen.

Genau diese materiale Wahrheit ist es, um die es im Alltag ständig geht – und vor Gericht – und bei der Repräsentation von Wissen in der Künstlichen Intelligenz. Leider werden daraus nicht immer die angemessenen Konsequenzen gezogen. Das berühmte Beispiel, das beim Übergang zu nichtmonotonen Logiken immer herangezogen wird, mag das illustrieren: Da stehen in einer Wissens-Datenbank die Sätze „Vögel können fliegen" und „Tweety ist ein Vogel", woraus gefolgert wird, dass Tweety fliegen kann. Jener Tweety, der hier für die Namenswahl vielleicht Pate stand (aus einer auch auf Deutsch ausgestrahlten Serie der Warner Bros.), war zwar ein Kanarienvogel, aber nun wird in die Datenbank noch der Satz aufgenommen: „Tweety ist ein Pinguin". Da man weiß, dass Pinguine nicht fliegen können, kann nun also auch Tweety nicht fliegen, und man landet bei einem Widerspruch. Der Grund ist klar: der Satz „(Alle) Vögel können fliegen" ist nicht material wahr. So einen Satz sollte man nicht in eine Wissensdatenbank aufnehmen (weil man zum logischen Schließen wahre Prämissen braucht). Um die nichtmonotonen Logiken zu ermöglichen, bleibt er aber drin, und man behilft sich mit der Einführung von vagen Wörtern („normalerweise").

Nach der Definition der materialen Wahrheit sieht man sehr schnell ein, dass für empirische Allsätze keine Gewinnstrategien existieren. Man kann also nicht *beweisen*, dass sie material wahr sind. Das ist das ganze Geheimnis des einst berühmten Falsifikationismus von Popper (1902–1094). Materiale Wahrheiten werden gebraucht, um für eine zentrale Aufgabe der Logik, das logische Schließen, wahre Prämissen identifizieren zu können. So gesehen ist die materiale Wahrheit – und damit der damit befasste Teil der Logik, die *materiale Logik* – nur ein Zwischenschritt auf dem Weg zur Kunst des Schließens, der *formalen Logik*. Um zu ihr zu gelangen, ist eine weitere Verschärfung der Gewinnidee erforderlich: nicht nur eine Gewinnstrategie für einen Satz soll existieren, sie soll so beschaffen sein, dass der $\mathcal{P}$ nur solche Elementarsätze behauptet, die der $\mathcal{O}$ vorher seinerseits schon behauptet hat. Solche Dialoge kommen ohne Prüfung der genannten Elementarsätze auf ihre Wahrheit aus, und in diesem Sinn ist die Gewinnstrategie von ihnen (inhaltlich) unabhängig. Daher nennt man sie „formal". Glücklich ist diese Wortwahl nicht („allgemeingültig" ist besser), aber sie hat sich nun einmal eingebürgert.

Allgemeingültige Aussagen liest man daher auch als gültige Aussageschemata. Das rechtfertigt die Verwendung *irgendwelcher* Buchstaben zu ihrer Formulierung. Und *deshalb* ist es egal, ob man das Schlussschema des Modus ponens als „A, A→B ergo B" oder als „p, p→q *ergo* q" notiert.

Erst an dieser Stelle tauchen unsere eingangs erwähnten Prinzipien wieder auf; etwa das „Prinzip des ausgeschlossenen Widerspruchs". Seine Formulierung ist recht schlicht: $\neg(A \wedge \neg A)$. Um seine Allgemeingültigkeit zu zeigen genügt der Dialog in Tabelle 2.

Das genannte Prinzip ist also ein allgemeingültiges Schema, das, weil es für beliebige (alle) A gilt, der Prädikatenlogik zweiter Stufe angehört. Das ist nicht weiter überraschend. Viel interessanter ist, dass sich dieses Prinzip nach einem dialogischen Logikaufbau beweisen lässt. Das haben wir ja eben getan. Damit verliert es seinen geheimnisvollen Status vollends: es ist nicht länger ein „Grundgesetz unseres Denkens" oder ein „Axiom" sonder ein wahrer Satz. Und bei genauem Hinsehen wird deutlich, dass dieser Satz sich nur aufgrund der Regeln für die sechs logischen Zeichen und die allgemeinen Spielregeln für Dialoge ergibt. Wenn man so will, kann man dafür auch sagen: „Wir haben die Logik gerade so gemacht, dass das Prinzip des ausgeschlossenen Widerspruchs gilt." Das haben wir natürlich deshalb getan, weil jedes vernünftige Argumentieren ohne (gegen) dies Prinzip unmöglich ist.

Klassische und andere Logik

Bisher haben wir keinerlei Hinweis darauf benötigt, welche Art von Logik wir eigentlich treiben. Das ist auch im Sinne eines methodischen Aufbaus ganz folgerichtig: wo noch keine Logiken voneinander unterschieden sind, kann man auf keine von ihnen verweisen. Vom Prinzip des ausgeschlossenen Widerspruchs wird man daher sagen können, es gelte „schlechthin". Anders ist das schon beim tertium non datur. Mit den bisherigen Regeln allein findet man dafür keine formale Gewinnstrategie. Will man dennoch, dass es gilt, muss man daher *die Regeln ändern*, genauer: die allgemeinen Spielregeln, nicht die Regeln für die logischen Zeichen. Das ist technisch nicht schwierig, gehört aber nicht mehr hierher (vgl. [2]). Die allgemein klassisch genannte Logik lässt sich dann charakterisieren als eine Variante, die an eine bestimmte Spielregel des Dialogs nicht gebunden ist. Man erhält dann nicht nur das tertium non datur, sondern das zusätzliche Ergebnis, dass die klassische Logik genau für den Spezialfall wertdefiniter Aussagen gemacht ist. Damit ist sie die Logik der Algorithmik: algorithmisch entscheidbare Prädikate sind wertdefinit.

Was sie hingegen nicht ist, weil sie der Situation nicht gerecht wird, haben wir schon gesehen: die klassische Logik ist nicht die Logik der KI, sie ist wohl auch nicht die Logik der Programmverifikation. Hier hat die konstruktive („intuitionistische") Logik ihr eigenes Bewährungsfeld, worauf Bauer (vgl. [1]) schon hingewiesen hat. Doch das sind Details, die über die Problematik der Grundbildung sicherlich hinausgehen. Gezeigt werden sollte hier nur ein Anknüpfungspunkt: der dialogische Logikzugang bildet einen solchen, an dem auch der oben erwähnte Gedanke einer Strukturierung durch Ordnungsideen, jetzt bezogen auf Spielregelvarianten, zu einem (Forschungs)programm ausgearbeitet werden könnte.

Literatur

1. Bauer, F.L.: Intuitionismus und Informatik. *Informatik-Spektrum* 4, S 284–287, (1999)
2. Inhetveen, R.: Logik. Eine dialog-orientierte Einführung. Leipzig 2003.
3. Kamlah, W, Lorenzen, P.: Logische Propädeutik. Vorschule des vernünftigen Redens. Stuttgart ^{3}1966

Zum Abschluss

Hiermit beschließen wir eine 6-teilige Artikelserie, die den Zweck verfolgte, die sprachlogischen Fundamente der Informatik für den Unterricht an allgemeinbildenden Schulen freizulegen. Sprachlogik in der Informatik ist wegen ihrer Unhintergehbarkeit einer Algorithmik und einer Gerätekunde methodisch voranzustellen, die beide von ihr abhängen. Es ist nicht einzusehen, dass sprachinteressierte und begabte Mädchen eine sprachbasierte Informatik scheuen sollten. Unsere Darstellung ist als eine stark verkürzte *Logische Propädeutik* [3] aufzufassen, zugeschnitten auf das Schulfach „Informatik".

Auf Vollständigkeit der Darstellung wurde kein großer Wert gelegt. Zwar entstand während der Abfassung der Serienbeiträge unter den Autoren eine heftige Debatte darüber, was nun zu einer Grundbildung unbedingt gehören muss und was nicht. Sollen z. B. Anwendungen der Logik im Hinblick auf die wichtige Teil-Ganzes-Relation mit ihren Komponentenbegriff (Mereologie) oder die Kontrollsphären-Lehre mit ihrem Transaktionsbegriff Gegenstand einer Grundbildung sein? Der Zeitdruck hat Diskussionen dieser Art unsanft beendet. Vollständigkeit ist in der Lehre Gott sei Dank nur eine Sekundärtugend. Zu den Primärtugenden gehört aber die Zwecksetzung, Informatik als Grundbildung nicht als ein Exzerpieren von Hochschulwissen mit anschließender Verdünnung aufzufassen. Informatik ist im Unterricht konstruktiv einzuführen, d. h. schrittweise, zirkelfrei und alles explizit machend. Die Informatik ist für die Didaktik keine Bezugswissenschaft. Das ist sie für ihre „Hauptkunden": Naturwissenschaften und Technik, Medizin und Wirtschaftswissenschaften. Vor einer nichtkonstruktiven Einführung in die Informatik ist zu warnen, weil die Folgen ähnlich sein werden wie vor rund 30 Jahren bei der Einführung der Mengenlehre. Schema und Ausprägung, Elementarsätze einer rationalen Grammatik, Abstraktion usw. sind nicht zu behandeln, als seien sie einer methodischen Einführung nicht bedürftig. Bevor die Informatik geboren wurde, gab es schon viele Essentials, die in einem Unterricht zu berücksichtigen sind. Keiner behauptet, dass mit der 6-teiligen Serie nun der „Stein der Weisen" gefunden wurde. Aber in den allbekannten Modus zu verfallen, der da lautet: „ignorieren und weitermachen!," ist angesichts der prekären internationalen Lage Deutschlands im Bildungsbereich unverantwortlich, was von allen Verantwortlichen zur Kenntnis genommen werden sollte. Und Verantwortung fängt immer ganz oben an und hört ganz unten auf.