universität wien

DIPLOMARBEIT

Titel der Diplomarbeit

Was ist ein Computer?

Verfasser

Manuel Schleiffelder

angestrebter akademischer Grad

Magister der Philosophie (Mag. phil.)

Wien, 2008

Studienkennzahl lt. Studienblatt:	A 296 332
Studienrichtung lt. Studienblatt:	Philosophie
Betreuer:	Ass.-Prof Dr. Wolfgang Pircher

[1]

[1] Meister der Spielkarten ('http://commons.wikimedia.org/wiki/Image:Vogel-Fünf.png') Lizenz: public domain.

Herzlicher Dank für ihre tatkräftige Unterstüzung in Form von Korrekturen und Zusprache gebührt vor allem Maria Krennmayr, David Unterholzner und Friedrich Penkner. Für die Betreuung der Diplomarbeit und für seine Geduld mit mir bedanke ich mich bei Herrn Prof. Wolfgang Pircher. Für die Unterstützung im DiplomandInnenseminar und auch danach bei Frau Prof. Elisabeth Nemeth und für die Förderung im Studium und am Institut bei Herrn Prof. Herbert Hrachovec. Danke LaTeX.

Inhaltsverzeichnis

1 Einleitung

1.1 Was ist ein Computer?

Eine philosophische Arbeit, die bereits im Titel eine 'Was ist ...'-Frage stellt, begibt sich in eine lange Reihe von Überblicksschriften. Alleine im Online-Katalog der Universitätsbibliothek Wien finden sich im Moment etwa zweihundert derartige Titel. Da gibt es: 'Was ist Ethik?', 'Was ist Sufismus?', 'Was ist Topologie?'. Diese Titel betreffen meist ganze Sachgebiete. Nur manche dieser Begriffe verweisen auch auf konkrete Entitäten. Etwa die Publikation 'Was ist ein Vogel?'[2]. Die Volltitelanzeige belehrt mich, dass es sich um ein Lehrbuch handelt, das im Untertitel als eine 'Einführung in Anatomie, Physiologie, Ökologie' ausgewiesen ist. Ich erhalte also vermutlich eine einführende Abhandlung über ein Objekt, dessen inneren Aufbau, dessen äußerliche Eigenschaften und seine Einordnung in der Umwelt.
Etwas äquivalent Umfassendes kann ich für 'Computer' nicht leisten. Ich gestehe ein, nicht alles, was mit Computern in Verbindung gebracht wird, zu erfassen zu können, und möchte deshalb einige Aspekte der gestellten Was-Frage herausgreifen, um sie im Hauptteil genauer zu untersuchen. Das betrifft vor allem die Frage nach Materialität und physischer Gebundenheit von Computern, die Frage nach Berechenbarkeitsbegriffen und Formalisierung, Fragen zur Analogie von Gehirn und Computer und daraus entstehenden Schwierigkeiten und Missinterpretationen.
Um den Problemabriss auf den Punkt zu bringen – den Leser in die Mitte der Dinge zu führen – möchte ich ein Beispiel vorführen, in dem meine Eingangsfrage einige Verwirrung stiftet. Ich stimme in den wenigsten Punkten

[2]Berger, Burkhard: *Was ist ein Vogel?* Hannover: Landbuch Verlag, 1991, LB-Naturbücherei

mit den im Beispiel erzielten Ergebnissen überein. Aber die wichtigsten Zusammenhänge und Begriffe zeichnen sich ab, und so gibt das Beispiel Hinweise auf die Stellen, an denen meine Untersuchung einsetzt.

1.1.1 Ein Beispiel zur Einführung

Oron Shagrir stellt mit dem Aufsatz 'Why we view the brain as a Computer' eine Frage, die der von mir gestellten sehr nahe kommt. Er will herausfinden, weshalb sowohl Theorien aus dem Bereich der 'Cognitive-Science' als auch der Neurowissenschaften sehr oft das Gehirn als eine Art Computer darstellen. Dazu muss er erst herausfinden – was auch mich interessiert – womit das Gehirn überhaupt verglichen wird: Was ist ein Computer? Seine erste Schlussfolgerung im Abstract ist etwas verwirrend: "But since we can view every physical system as a computer, it has been less than clear what this view amounts to."[3] Und wahrlich, es ist alles andere als klar, wie eine 'computational theory' erklärende Kraft haben soll, wenn der Beliebigkeit keine Grenzen gesetzt sind. Lassen sich keine Kriterien finden, um Computersysteme einzugrenzen? Der Ausblick in Shagrirs Abstract deutet – wenn auch nur als Vermutung – in eine Richtung, aus der, so glaube ich im Verlauf der Arbeit zeigen zu können, eine genauere Spezifizierung zu erwarten ist : "[...] in a nutshell, [computational theories] underscore correspondence relations between formal/mathematical properties of the electrical signals and formal/mathematical properties of the represented objects."[4]

Da sowohl das Gehirn als auch ein Rechner ('Computer') offensichtlich physikalische Entitäten darstellen, betrifft der Untersuchungsgegenstand

[3] Shagrir, Oron: Why We View the Brain as A Computer. Synthese, 153 2006, Nr. 3 S.1
[4] Shagrir S.1

auch die zugrundeliegende Materie. Oron Shagrir will einen neuen Ansatz zur Darstellung physikalischer Berechnung ('physical computation') geben, und schickt voraus: "[...] that, everything can be conceived as a computer, and that to be a computer is not a matter of fact or discovery, but a matter of perspective."[5] Er kommt zu diesem Schluss, indem er sich zuerst die Frage stellt, wie man physische Computer ('computing physical systems, such as desktops and brains') von anderen physischen Systemen ('that do not compute') unterscheiden könnte. Er will dazu alle uns bekannten Voraussetzungen, die einen Rechner ausmachen, untersuchen. Die erste Voraussetzung ist beschreibt er folgendermaßen: Jede Berechnung ist eine Art von Informationsverarbeitung ('computation is a species of information processing'). Das, was ein Computersystem tut, ist eine Art Informationsverarbeitung und 'Information' ist – laut Shagrir – immer auch gleichzeitig mit 'Repräsentation' verbunden. Weiters ist Repräsentation keineswegs objektiv, sondern reine Definitionssache: "[...] it is we who ascribe to them this representational force."[6] Damit scheitert die erste Voraussetzung als eine, eben nicht ausschließlich für Computersysteme zutreffende. Und es ergibt sich eine 'being-seen-as-Schlussfolgerung': "After all, every physical system can be interpreted as representing something. We can, for instance, take planetary systems, stomachs and washing machines to compute the solutions of the mathematical equations that describe their operations; that is, we can construe their states as representing certain numerical values."[7] Die zweite, in Betracht kommende Voraussetzung lautet: Ein Rechner operiert auf einem definierten Symbolsystem ('a computation operates solely on a symbol system, i.e., a system

[5] Shagrir S.1
[6] Shagrir S.5
[7] Shagrir S.5

of representations with combinatorial syntax and semantics'). Einerseits verwirft er dieses Kriterium als weitaus zu strikt, denn es würde alle Arten von Analog-Rechnern, die mit nicht-symbolischer Repräsentation ('non-symbolic representation') arbeiten, ausschließen. Andererseits gilt auch hierfür, dass dieses Symbolsystem durch alles Mögliche repräsentiert werden könnte. Somit ist auch Argument Nummer Zwei kein entscheidendes Kriterium. Als Drittes kommt das Formalitätskriterium ('formality condition: a process is computational only if it is formal'). Als Subkriterien für Formalität zählen hier: Mechanizität, Abstraktheit und Algorithmizität. Die ersten Beiden sind einfach abhandelbar: "After all, planetary movements, digestive processes, and wash cycles are also described by a set of mathematical equations, known as the laws of nature. And, much like digital computers, they can be seen as 'implementing' these equations, in the sense that their physical states mirror or otherwise correspond to the states of the describing equations."[8] Für sein drittes Formalitätskriterium – die 'Algorithmiziät' – gilt nach Shagrir dasselbe wie für die anderen Kriterien: "At some level of description, everything is algorithmic: every physical process can be seen as a discrete state-transition process."[9] Weiters gibt es Shagrir zufolge auch Funktionen, die nicht algorithmisch dargestellt werden können: "[...] there is the important class of neural networks, artificial and biological, that can be viewed computing, even though their dynamics are not 'digital' in any obvious sense."[10] Abschließend zitiert er Searles Aufsatz 'The Rediscovery of the Mind', der zu einem recht ähnlichen Ergebnis kommt: "[...] there is no way that computational cognitive science could ever be a natural science, because computation is not an intrinsic feature of the world. It

[8]Shagrir S.8
[9]Shagrir S.9
[10]Shagrir S.9

is assigned relative to observers."[11] Aber während Searle die Konsequenzen aus seinen Schlüssen zieht: "[... the claim that the brain is a computer] does not get up to the level of falsehood. It does not have a clear sense."[12] meint Shagrir, dass Searles Conclusio voreilig, beziehungsweise unfertig sei, denn "True, we do not discover that the brain is a computer, but we decide to so describe it" und fordert deshalb: "our task is to clarify what motivates us to apply the computational approch when studying the brain and cognitive functions."[13] Damit führt sich aber der Vergleich selbst in die Irre: "The computational approach is an explanatory strategy that seeks to explain a system's execution of semantic tasks."[14] Das soll wie folgt geschehen: "The explanatory force of this strategy arises from identifying correspondence relation between the computational structure of the system and certain mathematical properties of the states and objects that are being represented."[15] Grob gesagt: Es gibt eine Ähnlichkeitsrelation ('correspondence relation') zwischen der computerähnlichen Struktur eines Systems und den mathematischen Zuständen und Objekten, die das Zielsystem repräsentieren. Wenn etwas eine computerähnliche Struktur aufweist, wenden wir also den 'Computervergleich' als Erklärungsstruktur an. Die wichtige Frage lautet: Was ist mit dieser Analyse gewonnen? Polemisch gesagt: gar nichts. Denn erstens könnte im Sinne der Repräsentation neben Gehirnen auch die Sonne eine solche Struktur aufweisen und zweitens können wir nicht sagen, was ein Computer ist, da auch die Sonne als einer angesehen werden müsste ('being seen as'). Die Hauptfrage:

[11] Searle, John: *Is the Brain a Digital Computer?* Cambridge: MIT Press, 1992 S.212; zitiert von Shagrir S.10

[12] Searle s.225, vgl. Searle S.13

[13] Shagrir S.11

[14] Shagrir S.22

[15] Shagrir S.29

Was ist ein Computer? scheint in diesem Zusammenhang gänzlich unbeantwortbar.

1.1.2 Entstandene Fragen

Meine Polemik tut zwar dem Artikel unrecht, insofern er nicht beansprucht, eine allgemeingültige Antwort zu liefern, sondern ganz im Gegenteil, aus dem sehr spezialisierten Forschungsgebiet der Kognitions- und Neurowissenschaften kommt. Das heißt jedoch nicht, dass die Frage nach den Eigenheiten von Computersystemen aus einer anderen Perspektive nicht besser beantwortet werden kann. Und noch weniger bedeutet es, dass diese Fragen für andere Bereiche uninteressant oder irrelevant sind. Aus diesem Grund werde ich mich auf einer breiteren Argumentationsbasis dem Thema nähern. Ich unterteile die eingangs schon angedeuteten Unklarheiten in vier verschiedene Fragenkomplexe:
Das erste Kapitel untersucht die Was-Frage aus traditioneller Sicht. Aristoteles' Kategorienlehre, und die dazugehörige Einleitung von Porphyrius werden auf mögliche Antworten hinsichtlich der Kategorie 'Substanz' befragt. Denn die Frage danach, 'was etwas ist' wird dort mit dem Hinweis auf eine Gattung oder Art beantwortet. Dadurch ist Abgrenzung von anderen Dingen überhaupt erst möglich. Im Zentrum der Untersuchung steht die Frage, wie sich Computer zu- und einordnen lassen. Und obwohl sich die aristotelische Einteilung der Substanz in Arten und Gattungen für Computer nicht aufrecht erhalten lässt, stellt sich heraus, das der Begriff Substanz in mehrerlei Hinsicht zielführender und auch aussagekräftiger ist, als die Betonung des Materials, aus dem eine etwaige Rechenmaschine gebaut ist. Hierbei spielt vor allem die Frage nach einem Kriterium, Computer von

anderen Dingen zu unterscheiden, eine Rolle. Das gibt auch schon die Überleitung zum nächsten Abschnitt.
Im zweiten Kapitel werden jene Punkte, die Oron Shagrir als Kriterien zur Identifikation von Computersystemen verwirft, genauer unter die Lupe genommen. In einem kurzen Exkurs zur Geschichte der Formalisierung wird die Entwicklung von der Aristotelischen Logik über Freges Kalkülsprache bis hin zur Loslösung der Form eines Beweises von seinem Inhalt nachvollzogen. Sibylle Krämer schreibt: "Die Grundidee der Formalisierung besteht darin, das Manipulieren von Symbolreihen von ihrer Interpretation abzutrennen."[16] Somit kann, sofern die Manipulation in klare Regeln gefasst ist, auch eine Maschine diese Aufgabe übernehmen. Darauffolgend rücken die maßgeblichen Entdeckungen und Beweise aus der ersten Hälfte des 20. Jahrhunderts in den Mittelpunkt. Als allgemeines Modell für eine 'symbolische Maschine' wird die Turingmaschine eingeführt. Die wichtigste Frage aus dieser Perspektive lautet, inwiefern Algorithmizitäts- und Berechenbarkeitsbegriffe über die Beschränkungen, denen Computer unterliegen, Auskunft geben können. Taugen sie dazu, Computer von anderen Dingen zu unterscheiden?
Im dritten Kapitel soll nun auf Basis dieser Grundlagen näher auf die Unterscheidung zwischen analogen und digitalen Rechenmaschinen eingegangen werden. Diese erweist sich als maßgeblich für die Frage, wie berechnende und nicht-berechnende Dinge auseinander zu halten sind. Im Zuge dessen wird erörtert, ob und inwiefern Berechnungen über die allgemeinen Beschränkungen hinaus, denen Turingmaschinen – und damit alle bisherigen Rechenmaschinen – unterliegen, angestellt werden können. Spekulationen in diese Richtung sind auch als 'Hyperturing' oder

[16]Krämer, Sybille: *Symbolische Maschinen*. Darmstadt: Wissenschaftliche Buchgesellschaft, 1988 S.176

'Hypercomputation' bekannt. Die im vorangegangenen Kapitel erarbeitete Church-These in Kombination mit Turings Berechenbarkeitsbegriff wird noch einmal einer genaueren Analyse unterzogen. Dabei wird eine aus neueren Beiträgen stammende, spezielle Formulierung der Church-Turing-These behandelt, die versucht, über Naturgesetze die mathematischen Grundlagen von Rechenmaschinen an die physische Welt zu binden. Mit Referenz auf das erste Kapitel wird sich dieser Versuch als Sackgasse erweisen.
Im vierten Kapitel werden die Auswirkungen der Analogisierung von Gehirnen und Computern ins Auge gefasst. Da Maschinen plötzlich nicht mehr ausschließlich – von Menschen gesteuert – Arbeit verrichten, sondern dessen Aufgabe übernehmend, selber Information verarbeiten, ist eine Re-Evaluierung dessen, was Denken ausmacht, notwendig geworden. Alan Turings Papiermaschine belegt die eine Seite der Analogie:

> Es ist möglich den selben Effekt einer Rechenmaschine zu erreichen, indem man eine Liste von Handlungsanweisungen niederschreibt und einen Menschen bittet, sie auszuführen. Eine derartige Kombination eines Menschen mit geschriebenen Instruktionen wird 'Papiermaschine' genannt. Ein Mensch, ausgestattet mit Papier, Bleistift und Radiergummi sowie strikter Disziplin unterworfen, ist in der Tat eine Universalmaschine.[17]

Wenn auch Computer mittlerweile um vieles schneller geworden sind, so trifft das immer noch zu. Wie steht es aber mit der anderen Seite der Analogie? Ein Ausflug zu den wichtigsten Errungenschaften der Kybernetik und Gotthard Günthers trans-klassischen Maschinen soll Licht in die Sa-

[17] Turing, Alan M.; Dotzler, Bernhard. Kittler, Friedrich (Hrsg.): *Intelligence Service*. Berlin: Birkmann & Bose, 1987 S.91

che bringen. Abschließend folgen noch zwei Unterkapitel zu John Searles Intentionalismus und dessen Auswirkung auf obiges Beispiel.

1.1.3 Was hier nicht bearbeitet wird.

Die Analyse des vielfältigen Einsatzes von Computern in allen Lebensbereichen, die sozialen Seiten der Computernutzung sowie die damit einhergehenden gesellschaftlichen Veränderungen, müssen hier leider unter den Tisch fallen. Positiv gesagt: Wenn man zustimmt, dass der Computer im Bezug auf seine Nutzung[18] drei verschiedenen Bereichen zugeordnet werden kann, so wird hier nur seine Bedeutung als Werkzeug und Maschine gewürdigt. Die aktuelle Debatte, die Computer – im Sinne des dritten Nutzungsaspekts – als Medium interpretiert, nimmt ihren Ausgang, wo diese Untersuchung endet. Auch der Netzwerk- oder Vernetzungs-Aspekt von Computern baut auf den hier entwickelten Begrifflichkeiten auf und schließt in gewisser Weise die Ära der Vergleiche mit Denkmaschinen und Geistesprothesen ab.[19]

[18]Dennis Mocigemba unterscheidet drei Nutzungsmetaphern: Werkzeug – Maschine – Medium. Vgl. Mocigemba, Dennis: *Die Ideengeschichte derComputernutzung*. Dissertation Technischen Universität Berlin, Berlin, Deutschland, 2003.

[19]Vgl. Krämer, Sybille: Was haben die Medien, der Computer und die Realität miteinander zu tun? In Krämer, Sybille (Hrsg.): Medien Computer Realität. Band 1379, Frankfurt am Main: Surkamp Verlag, 1998

2 Hauptteil

2.1 Die Berechnungswesen.

Inwiefern unterscheidet sich die Was-Frage nach einem Computer von der Was-Fragen nach einem Vogel? Eine kurze Befragung der Kategorienlehre des Aristoteles darüber, wonach wir üblicherweise mit einer Was-Frage suchen, soll folgende drei Punkte herausarbeiten: Die Aussagekraft einer Was-Frage in einem relationalen Kosmos; das Substanzsein von Computersystemen und ein Vergleich der Begriffe Substanz und Material.

2.1.1 Die Kategorienlehre bei Aristoteles

Den Anfang soll nun ein kurzer Ausflug zu einer traditionellen Art der Behandlung von Was-Fragen machen. Eine Was-Frage referiert bei Aristoteles auf die Einteilung des Seienden, das im Sinne von 'Was ist etwas?' den Gattungen und Arten der Kategorie Substanz ('ousia') zugeordnet wird. Die Substanz, nicht zu verwechseln mit 'Materialität', bezieht sich auf das Wesen einer Sache, und lässt sich etymologisch erhellen insofern sich das Substantiv 'ousia' vom altgriechischen Partizip für 'seiend' herleitet. Für meine Untersuchung werde ich erst einige Bestimmungen aus dem Vorwort des Porphyrius[20] durchgehen, das als Einleitung in die Kategorienlehre des Aristoteles gedacht, und den meisten Ausgaben vorangestellt ist. Zur Einführung in das Verständnis der Unterteilung der Kategorien behandelt er die Begriffe Gattung, Art, Differenz, Proprium und Akzidenz. Um es noch einmal herauszustreichen: es handelt sich dabei

[20] Porphyrius lebte im 3. Jahrhundert n.Chr. und war ein Schüler Plotins und Lehrer des Jamblichus.(Angebl. Geb. 234 und Gest. 304 n.Chr) vgl. Vorrede des Übersetzers zu der Einleitung des Porphyrius. In: Aristoteles; Rolfes, Eugen (Hrsg.): *Kategorien / Lehre vom Satz*. Band 8/9, Philosophische Bibliothek. Hamburg: Verlag von Felix Meiner, 1962 S.9f

nicht um eine Erläuterung der einzelnen Kategorien, sondern er gibt lediglich Aufschluss darüber, wie sie ihrerseits unterteilt werden. In den Beispielen geht es aber doch hauptsächlich um die – eingangs erwähnte – Kategorie der Substanz. Sie genießt auch bei Aristoteles Vorrang vor den Kategorien der Quantität, Qualität, Relation, Ort, Zeit, Lage, Haben, Tun und Erleiden. Demnach ist die Frage nach dem Wesen einer Sache immer eine Frage nach deren Art oder Gattung im Rahmen der Kategorie der Substanz, nicht nach arbiträren Unterscheidungen ('Differenzen') oder den ihnen nur zukommenden Eigenschaften ('Akzidenzien'). "Von der Differenz aber und den gemeinsamen Akzidenzien unterscheidet die Gattung sich [...] dadurch, daß, wenn die Differenzen und die gemeinsamen Akzidenzien auch von vielem und der Art nach Verschiedenem ausgesagt werden, die Aussage doch nicht im Sinne einer Wesensbezeichnung, sondern im Sinne einer Qualitätsbezeichnung geschieht. Denn auf die Frage nach dem Was des Dinges, von dem sie ausgesagt werden, antworten wir mit der Gattung, nicht mit den Differenzen und Akzidenzien."[21]

Durch die Einteilung in Gattungen und Arten entsteht eine hierarchische Struktur. Eine Gattung bezeichnet eine Vielheit – sowohl verschiedene Arten als auch Individuen – und gibt deren Wesenheit an. Was etwas ist, oder im weiteren Sinne, der Stoff der ihm zugrunde liegt, wird ausgesagt (siehe weiter unten: 'zweite Substanz', ebenfalls nicht mit 'Materialität' zu verwechseln). Bei Porphyrius sowie auch bei Aristoteles werden zur Verbindung von Untergeordnetem und Übergeordnetem in der deutschen Übersetzung die Verben 'aussagen' und 'aufheben' verwendet. Etwas wird von einem Einzelding (der Zahl nach 'eines'; das kann sowohl ein Individu-

[21] Porphyrios: Einleitung in die Kategorien. 1b, zitiert nach der Übersetzung von Eugen Rolfes. In: Aristoteles; Rolfes, Eugen (Hrsg.): Kategorien / Lehre vom Satz. Band 8/9, Philosophische Bibliothek. Hamburg: Verlag von Felix Meiner, 1962, S.13

um, als auch auf höherer Ebene eine Art sein) ausgesagt, indem es ihm als Prädikat zukommt. Oder etwas wird nicht von einem Einzelnen ausgesagt, dann kommt ihm das Prädikat nicht zu. Das Prädikat 'Mensch' kann vom Einzelwesen 'Platon' wahrheitsgemäß ausgesagt werden: Platon ist ein Mensch. Man würde den Term 'aussagen' umgangssprachlich etwas unmißverständlicher mit 'etwas kann von sich sagen' übersetzen. Jedenfalls kann Porphyrius damit folgende Abgrenzung treffen: "Eigentümlich aber hat die Gattung, daß sie von mehrerem ausgesagt wird als die Differenz, die Art, das Proprium und das Akzidenz."[22] Also ist die Gattung allgemeiner und in der Hirarchie weiter oben angesiedelt. Der Weg in der Hierarchie von unten nach oben verläuft über das 'Ausgesagt-Werden' und der Weg von oben nach unten läuft über das 'Aufgehoben-Werden'. Nämlich in dem Sinne, als mit dem Aufheben der darüber liegenden Einteilung auch die in ihr enthaltenen kleineren Einteilungen oder Individuen aufgehoben werden. Wenn es keine Gattung für Sinnenwesen gibt, so kann es auch keine, denselben ja zugehörigen, Menschen geben. "Ferner enthält die Gattung die Differenz der Potenz nach; denn das Sinnenwesen ist teils vernünftig, teils unvernünftig; die Differenzen aber enthalten nicht die Gattungen. Ferner sind die Gattungen früher als die unter sie fallenden Differenzen und heben sie deshalb mit auf."[23] Porphyrius bringt das Beispiel von der, unter die Kategorie Substanz fallenden Gattung der Sinnenwesen, die sich zwar auch durch die Differenz von 'vernünftig' und 'unvernünftig' unterteilen lässt, in der sich aber erst durch das Proprium 'kann lachen', da es nur einer Art zukommt, die Art Mensch abhebt. Die einzelnen Individuen dieser Art sind dann: Sokrates, Platon usw. Als Gegenbeispiel wird in diesem Fall der für uns nicht mehr ganz greifbare 'En-

[22] Porphyrios: Einleitung in die Kategorien. 4b, In: Aristoteles S.25
[23] Porphyrios: Einleitung in die Kategorien. 4b, In: Aristoteles S.25

gel' angeführt, der zwar auch 'vernünftiges Sinnenwesen', aber offenbar nicht 'des Lachens fähig' ist.

> Ferner wird, wie gesagt, die Gattung auf die Frage, was etwas ist, dagegen die Differenz auf die Frage, wie beschaffen es ist, prädiziert. [...] Ferner ist die Gattung für jede Art eine, wie für Mensch Sinnenwesen, der Differenzen aber sind viele, wie vernünftig, sterblich, aufnehmendes Subjekt von Verstand und Wissenschaft, durch die er sich von den anderen Sinnenwesen unterscheidet. Endlich gleicht die Gattung dem Stoff die Differenz der Form.[24]

Die Art bezeichnet auch eine 'Vielheit', aber nur solche Ansammlungen, die sich zählen lassen; was sich der Zahl nach unterscheidet (z.B. Weiß als Art von Farbe, Dreieck als Art von Figur). Eine Art kann ebenso wie die Gattung als Antwort auf eine Was-Frage stehen, ist aber immer einer Gattung untergeordnet. Jedoch ist die Hierarchie nicht auf drei Stufen (Gattung – Art – Individuum) beschränkt. Eine Art kann für weitere Unterarten durchaus Gattung sein, und umgekehrt kann eine Gattung für übergeordnete Gattungen Art sein. "Das Individuum ist also in der Art enthalten und die Art in der Gattung. Denn die Gattung ist ein Ganzes und das Individuum ein Teil, die Art aber ist Ganzes und Teil zugleich. Aber der Teil ist eines anderen (gehört ihm), während das Ganze nicht eines anderen, sondern in anderen ist, in den Teilen nämlich."[25] Allerdings ist die mögliche Kette auch nicht beliebig lange. Nach unten hin, ist sie durch das spezifische unteilbare Individuum begrenzt, und nach oben hin durch die allgemeinste Gattung oder die Kategorie. "Man muss also beim Abstieg zum Spezi-

[24]Porphyrios: Einleitung in die Kategorien. 4b, In: Aristoteles S.25f
[25]Porphyrios: Einleitung in die Kategorien. 3a, In: Aristoteles S.18

ellsten mittels der Einteilung durch die Vielheit schreiten, dagegen beim Aufstieg zum Generellsten, die Vielheit in Eines zusammenfassen. Denn die Art und noch mehr die Gattung sammelt das Viele in einer Natur, das Besondere und Einzelne aber teilt umgekehrt das Eine immer in eine Vielheit, denn durch die Teilnahme an der Art sind die vielen Menschen einer, aber durch die Menschen im besonderen und einzelnen ist der eine und gemeinsame Mensch viele, da das Einzelne immer teilt, während das Gemeinsame sammelt und eint."[26]

Sowohl Gattungen und Arten als auch Akzidenzien entstehen aus einer Differenz. Etwa die Differenz 'des Lachens fähig' – 'des Lachens nicht fähig' unterscheidet die Art 'Mensch' von allen Nicht-'Menschen'. Eine artbegründende Differenz wird auch Proprium genannt. Es entspricht einer Eigentümlichkeit, die nur einer speziellen Art zukommt (deren Proprium es ist) und wird von den Individuen ausgesagt. Die Arten und Gattungen werden aber zusätzlich durch weitere nicht-artbildende Differenzen unterteilt: "Ferner ist die Differenz früher als die durch sie begründete Art: vernünftig hebt, wenn aufgehoben, Mensch mit auf, aber Mensch hebt, wenn aufgehoben, nicht vernünftig auf, da es Engel gibt."[27]

Die unterste Stufe der Differenzen bilden die Akzidenzien. "Akzidenz ist, was demselben Subjekt in gleicher Weise beiwohnen und nicht beiwohnen kann, oder: was weder Gattung ist, noch Differenz, noch Art, noch Proprium, aber immer in einem Träger subsistiert."[28] Sie bezeichnen 'wie' etwas ist, treten auf und verschwinden ohne Untergang des Subjekts, also ohne das Subjekt aufzuheben. Sie lassen sich in trennbare und untrennbare, 'an sich' (ganz oder gar nicht) oder mitfolgend (mehr oder minder) zukommen-

[26]Porphyrios: Einleitung in die Kategorien. 2b, In: Aristoteles S.17
[27]Porphyrios: Einleitung in die Kategorien. 5b, In: Aristoteles S.30
[28]Porphyrios: Einleitung in die Kategorien. 4a, In: Aristoteles S.23

de Differenzen einteilen. Eine Differenz ist spezifisch, wenn sie verschiedene Dinge unterscheidet, aber kein Proprium bzw. nicht artbegründend ist, d.h. wenn sie mehreren zukommt (Vernunft scheidet die Menschen von den Pferden, kommt aber auch Göttern oder Engeln zu). Ansonsten ist sie einfach.

Diese Unterscheidungen entstammen einem System, das zur Betrachtung und Beherrschung der Natur erdacht wurde. Sie lassen sich nicht so ohne Weiteres auf meine Fragestellung übertragen. Schon Porphyrius erwähnt, dass "von Natur aus Schiffahrt treiben"[29] als Differenz, Proprium oder Akzidenzium in dieser Einteilung wohl Schwierigkeiten bereiten wird. Tiefergehende Fragestellungen, die das Verhältnis von Natur- und Kulturerscheinungen im Rahmen der westlichen Philosophie beleuchten, seien hier beiseite gelassen.

2.1.2 Die Einteilung der berechnenden Dinge

Da sich die Arbeit im Folgenden wieder konkreter den Computern zuwendet, möchte ich ein paar Worte von Luciano Floridi vorausschicken, die in gewisser Weise auch mich und das Folgende betreffen: "to a man with a computer-hammer, everything looks like information-nail, and I'm afraid that in my case it does so foundationally."[30] Ich möchte also hervorheben, dass es mir nicht um eine Neubewertung der Naturphilosophie des Aristoteles geht. Im Gegenteil, es geht mir darum, Teile dieser Bestimmungen für meine Eingangsfrage fruchtbar zu machen.

Dem Vorausgegangenen entsprechend scheint nun klarer, welche Antwort

[29] Porphyrios: Einleitung in die Kategorien. 4a, In: Aristoteles S.22

[30] Floridi, Luciano: Two Approaches to the Philosophy of Information. Minds and Machines, 13 2003, Nr. 4 S.9

man auf eine Was-Frage, traditionell erwarten kann: Die Antwort besteht in der Angabe einer Gattung oder Art. Diese wiederum wird jeweils durch ein Proprium, eine artbildende Differenz bestimmt. Die Einteilung, die unter diesen Bedingungen vorgenommen wird, entspricht im Prinzip einer hierarchischen Datenstruktur mit kleinen Erweiterungen. Das Grundschema bildet die Welt als hierarchische Baumstruktur ab (vgl. Sinnenwesen – Mensch – Platon).

> Eine Datenstruktur B=(K,R) heißt (gerichteter Wurzel-) Baum, wenn R aus genau einer Relation besteht, die die folgenden Bedingungen erfüllt:
>
> - (1) Es gibt genau einen Knoten W, der keinen Vorgänger hat. W heißt die Wurzel des Baums.
> - (2) Jeder Knoten, mit Ausnahme der Wurzel W, hat genau einen Vorgänger.
> - (3) Für jeden von der Wurzel verschiedenen Knoten k gibt es einen Folge $W = k_0, k_1, ..., k_n = k, (n \geq 1)$ von Knoten, bei der k_i der Nachfolger von k_{i-1} ist $(1 \leq i \leq n)$.[31]

Durch eine solche Struktur lassen sich nur Eltern-Kind-Beziehungen (also 1:1 oder 1:n Relationen) abbilden. In unserem Fall entspricht dies der Struktur von Gattungen, Arten und Individuen. Mit Hilfe von weiteren Differenzen – die zur Baumstruktur parallele bzw. redundante Einteilungen zulassen – und zukommenden Akzidenzien, die die Individuen anreichern, wird das Schema erweitert. Ich will nicht darauf eingehen, wie praktisch, genial, richtig, oder nicht, das Aristotelische Schema ist, sondern ich will

[31] Schneider, Hans-Jochen (Hrsg.): *Lexikon Informatik und Datenverarbeitung*. München: R.Oldenburg Verlag, 1997 S.85

sehen, ob und wie man mit diesen Mitteln auch meiner Aufgabe beikommen kann.
Wenn ich jetzt das Gebiet der 'berechnenden Dinge' (in Anlehnung an die 'Sinnenwesen' könnte ich sie 'Rechenwesen' nennen) ebenfalls in ein hierarchisches Substanz-Schema einteilen möchte, wird bald klar, dass eine Baumstruktur hier aus verschiedenen Gründen nicht durchgehalten werden kann. In der Gattung der berechnenden Dinge gäbe es drei logische Unterarten: analoge Rechenmaschinen, digitale Rechenmaschinen und neuronale Netze.
Die Art der analogen Rechenmaschinen (deren Proprium etwa 'operiert mit reellen Werten' sein könnte), die ja wiederum Gattung ist, enthielte die Unterart der 'einfachen analogen Steuerungen'. Eines der Individuen könnte zum Beispiel ein bestimmter Wattscher-Fliehkraftregler sein. Allerdings wäre die Art der 'einfachen analogen Steuerungen' ein sehr bunter Haufen. Wenn wir nämlich behaupten, dass ein Fliehkraftregler eine Berechnung durchführt, ist es schwer, einem fallenden Apfel abzusprechen, dass er in diesem Moment die Gravitationskonstante berechnet. Genaueres zu diesem Problem findet sich im Kapitel über Analogrechner. Einer hierarchischen Einteilung kommt das jedenfalls nicht zugute.
Digitale Rechenmaschinen stellen uns vor ein weiteres Rätsel, denn eine Maschine für eine bestimmte Berechnung könnte ganz verschieden realisiert sein. Die selbe Berechnung könnte mittels einer elektrischen Schaltung, einer mechanischen Vorrichtung (man denke an Charles Babages Difference Engine), hydraulisch oder auch mit optischen Mitteln durchgeführt werden. Wiederum ist es scheinbar von der Zuschreibung abhängig, ob eine Vorrichtung nun 'berechnet' oder nicht. Möglicherweise beschreibt eine analoge Rechenmaschine – in einem gewissen Toleranz-

feld – die selbe Berechnung wie eine digitale Rechenmaschine. Bei komplexeren Digitalrechnern ist (oft) neben den Hardwarekomponenten auch noch Software für die Berechnung verantwortlich. Können verschiedene Softwareausstattungen in einem solchen Schema berücksichtigt werden? Die Analyse dieses Problems wird auf einen späteren Zeitpunkt verlegt. Die Einordnung in ein strenges hierarchisches Datenmodell ohne Redundanzen oder Mehrfachnennungen ist bei Computern jedenfalls nicht möglich.

Zumindest biologische, neuronale Netze lassen sich noch recht gut genealogisch unterteilen. Davon wären jedoch alle neuronalen Netze (und das betrifft die meisten künstlichen neuronalen Netze), die auch auf einem Digitalrechner simuliert werden können, auszunehmen. Umgekehrt kann ein Mensch (als Metonymie für Gehirn) mit einer Liste von genauen Instruktionen, Papier und Bleistift alle Funktionen berechnen, die auch eine universale Turingmaschine berechnen kann[32]. Wenn auch meist langsamer.

Am ehesten wird noch eine Baumstruktur sichtbar, wenn man zum Beispiel in der Domäne der modernen Digitalrechner einen bestimmten Hersteller als Art, die von diesem produzierten Typen als Unterarten und die einzelnen Geräte mit fortlaufender Seriennummer als Individuen betrachtet. Verschiedene Ausstattungen kommen als Akzidenzien hinzu. Das mag aber auch einfach damit zusammenhängen, dass auf diesem Gebiet technische Fortschritte in gewissen Aspekten genealogisch ablaufen. Wenn zum Beispiel neuere Entwicklungen auf Ältere aufbauen.

Da ich vorhin schon den Vergleich einer hierarchisch strukturierten Datenbank, die die Welt abbildet, bemüht habe, liegt es an der Hand, doch

[32]Vgl. Turing, A. M.: Computing Machinery and Intelligence. Mind, 59 1950, Nr. 236

auch flexiblere Modelle heranzuziehen. Etwa ein relationales Datenmodell. Ihre "Relationen besitzen die Möglichkeiten, komplexe Strukturen abzubilden. Es können Wiederholungsgruppen, hierarchische und netzartige Datenstrukturen modelliert werden. Also auch n:m strukturierte Beziehungen können direkt dargestellt werden."[33]
Der Erfinder der Relationalen Algebra, Edgar F. Codd, hat in den 1970'er Jahren das Datenbankwesen revolutioniert. Bis dahin waren das Netzwerkdatenmodell und das hierarchische Datenmodell als Standards im Einsatz. Eine relationale Algebra ist eine formale Sprache, mit der sich Abfragen über einem relationalen Schema formulieren lassen. Die Relationen, auf die sie sich gründet, sind im Prinzip Mengen, die sich als Tabellen (mit Zeilen und Spalten) darstellen lassen. Diese Relationen haben Attribute (in den Spalten) und Tupel (in Zeilen). Attribute wiederum sind einem bestimmten Wertebereich zugeordnet, der definiert, welche Werte ein entsprechender Zeileneintrag annehmen kann. Die formale Sprache arbeitet mit Operatoren, die eine Menge von Relationen verknüpfen können. Durch diese Operation wird eine neue Relation erzeugt. Neu erzeugte Relationen können ihrerseits wieder in Operationen eingebunden sein.[34] Es gibt verschiedene Grundoperationen, mit denen alle Datenbankabfragen formuliert werden können: SELECT, PROJECT, UNION, INTERSECTION, DIFFERENCE, CARTESIAN PRODUCT und JOIN. Zur genaueren Erläuterung der einzelnen Operationen verweise ich auf die Fußnoten,[35] da dies für meine vorliegende Zielsetzung nicht essentiell wichtig ist.

[33]Karagiannis, Dimitris; Endres, Albert. Krallmann, Herbert. und Schnupp Peter (Hrsg.): *Wissensbasierte Datenbanken*. Band 8.2, Handbuch der Informatik. München: R.Oldenburg Verlag, 1994 S.19

[34]Vgl. Codd, Edgar F.: *The relational model for database management*. Reading (Mass.): Addison-Wesley, 1990

[35]Siehe ('http://db.grussell.org/ch5.html', Son Okt 19 12:41:23 CEST 2008) oder ('http://aktuell.de.selfhtml.org/artikel/datenbanken/',Son Okt 19 12:41:23 CEST 2008)

Als kurzes Beispiel dient mir eine Tabelle mit den drei Attributen Indexnummer, Name, Geburtsjahr. Die Tabelle enthält bereits eine Relation, da sie in jedem Tupel (jeder Zeile) die Attribute in Relation zueinander setzt. Etwa die Tabelle (Menschen):

Index	Name	Jahr	...
1	Josef	1970	...
2	Martina	1981	...
3	Theresia	1975	...
4	Monika	1981	...
...	...	...	...

Wende ich auf sie die Relation: SELECT Name, Jahr FROM Menschen WHERE Jahr = '1981'; an, dann bekomme ich folgende Relation zurück,

Index	Name	Jahr
2	Martina	1981
4	Monika	1981

die ich wiederum mit anderen Relationen verbinden könnte.

Angenommen, ich überführte zum Beispiel die Begriffe aus Porphyrius' Einleitung in eine relationale Datenbank. Ich würde zwei Haupttabellen anlegen: Eine Tabelle, die Differenzen (bestehend aus fortlaufendem Index[INT[36]] und Name) auflistet, und eine Tabelle, die einen Eintrag für jedes Individuum (ebenso mit fortlaufendem Index[INT], Namen) enthält. Für Proprium und Akzidenz reicht jeweils eine Zuordnungstabelle[37]. Gattung und Art würde man als selbstreferentielle Tabelle mit den Attributen Index,

[36] Aus den natürlichen Zahlen (egl. sowie lat. 'integer')

[37] Attribute für Proprientabelle: [INT], Fremde Indexnummer zur Zuordnung zur entsprechenden Differenz[INT], Fremde Indexnummer zur Zuordnung zur gebildeten Art[INT] und evtl. eine Benennung), Attribute für die Akzidenzientabelle: [INT], Fremde Indexnummer zur Zuordnung zur entsprechenden Differenz[INT], Fremde Indexnummer zur Zuordnung zum Individuum[INT] und evtl. eine Benennung.

SelfIndex (der höheren Gattung), PropriumsIndex (des artbildenden Propriums) und evtl. Name anlegen. Angenommen, ich definiere weiters, dass die höchste Gattung (die Wurzel des Baumes, die ja per Definition nicht Art ist) in der Gattung-Art-Tabelle durch eine '0' in der Selfindex-Spalte beziehungsweise Nichtzuordnung zu einer übergeordneten Art bestimmt ist: Mit einer speziell geschleiften Abfrage kann dann jederzeit der Weg von einem speziellen Individuum zur allerhöchsten Gattung abgefragt werden. Also lässt sich das hierarchische Schema gut in eine relationale Datenbank überführen. Man beachte, dass es sich – zumindest Gattungen, Arten und Individuen betreffend – natürlich immer noch um ein streng hierarchisches Datenschema handelt.
Dies ist jedoch nicht die einzige Möglichkeit, obigen Sachverhalt in einem relationalen Datenbankmodell abzubilden. Andere Arten der Normalisierung[38] sind möglich. Eigentlich ist diese Überführung in ein relationales Schema nicht unbedingt notwendig. Sie bot sich aber als Möglichkeit zur Veranschaulichung an, dass ein rein hierarchisches Schema immer auch relational abgebildet werden kann.
Wenn ich nun dieses Schema für die 'Rechenwesen' erweitere, die streng hierarchische Ordnung aufgebe, zudem Redundanzen und Mehrfachzuordnungen möglichst vermeide, so verändert sich auch der Gehalt einer möglichen Antwort auf eine Was-Frage. Die Relation kann ich als Datenbankabfrage folgendermaßen formulieren: 'SELECT Index, Name, Baujahr FROM Computer RIGHT JOIN Substrat, Bauweise, Architektur, Funk-

[38] Die Normalisierung ist ein Prozess der Umgestaltung von Datenbankstrukturen in der relationalen Welt. Ziel ist die Vermeidung von zahlreichen Problemen, die sich aus der einfachen Speicherung 'in einer großen Tabelle' ergeben. Das Ergebnis der Normalisierung sind zahlreiche, kleinere Relationen, in denen weniger bis keine Abhängigkeiten zwischen einzelnen Feldern bestehen und Redundanzen nicht mehr auftreten. (Vgl. selfhtml.org 'http://aktuell.de.selfhtml.org/artikel/datenbanken/joins/', Fre Okt 17 11:40:59 CEST 2008)

tionsumfang ON Substrat. Subst = ’elektronisch’ AND Computer.Index = Substrat.CompIndex AND Bauart.Bau = ’digital’ AND Computer.Index = Bauart.CompIndex AND Architektur.Arch = ’VonNeumann’ AND Computer.Index = Architektur.CompIndex ...’ (vgl. Datenbankabfrage). Eine typische Antwort könnte ausformuliert lauten: ’Alle Individuen der Relation ’Computer’, die in der Relationstabelle ’Substrat’ als ’elektronisch’, in der Relationstabelle ’Bauart’ als ’digital’, in der Relationstabelle ’Architektur’ als ’VonNeumann’, ..., vermerkt sind usw.’. Also erfahre ich, wie ein Objekt in ein relationales Datenmodell eingebunden ist; es hat aber seine fixe Position in der Hierarchie der Arten und Gattungen verloren. Diese ist nur mehr als, in bestimmten Relation zu allen anderen Einträgen stehend, erfragbar. Und sie ist nur so gut definiert, wie die in Relation stehenden Objekte. Mit einem Wort: wurzellos. Alles hängt von den Ausgangsrelationen (Ausgangstabellen) ab. Sind diese veränderlich, verändert sich auch die Position des gesuchten Objektes

Der Philosoph mag den Anschein verzeihen, dass ich Aristoteles hier zu einem Datenbankprogrammierer degradiere. Dem ist nicht so. “Die Aristotelische Syllogistik ist zwar formal aber kein formalistisches System. Dabei sei unter einem ’formalistischen System’ ein System verstanden, in welchem die Richtigkeit von Ableitungen innerhalb des Systems nachgeprüft werden kann, ohne daß man die Bedeutung der in den Ableitungen benutzten Ausdrücke und Symbole in Rechnung stellen muß.”[39] Bei Aristoteles ist es wesentlich, dass die Bestimmungen innerhalb der Kategorie der Substanz einen gültigen Bezug zum ’absoluten’ Sein haben. Mit einer relationalen Datenbank, formal einer 4relationalen Algebra, geht dieser Bezug verloren. Wie die Relationen und ihre Werte als System gestaltet

[39] Krämer: *Symbolische Maschinen* S.75

sind, wird im Gegensatz zum Aristotelischen System, das wesentlich von der Lebenswelt bestimmt ist, nur mehr vom Systemgestalter festgelegt. Zur Bestimmung der Adäquatheit werden dann etwa empirische Methoden herangezogen. Ich behandle hier jedoch nur einen kleinen Ausschnitt aus der Kategorienschrift. Ein Verweis auf die Relationalität im Einteilungsschema legt zudem einen Vergleich mit Cassirers Relationsbegriffen nahe. Das geht aber über die Erfordernisse dieser Arbeit hinaus.
Ist die Eingangsfrage damit sinnlos geworden? Im Gegenteil. Zu ihrer Beantwortung müssen wir Antwort auf zwei Subfragen finden:

- (1) Inwiefern kann außerhalb einer hierarchischen Ordnung der Dinge eine Abgrenzung 'berechnender' von 'nicht berechnenden' Dingen gemacht werden? (Was berechnet? Was berechnet nicht?)

- (2) Referieren wir auf das einer Was-Frage zugrundeliegende Objekt als Substanz oder als Material?

Anders formuliert: ist ein Bezug auf die Kategorie der Substanz im Sinne der Aristotelischen Definition noch sinnvoll?

2.1.3 Substanz als Grundlage von Computersystemen

> For Aristotele, 'substances' are the things which exist in their own right, both the logically ultimate subjects of predication and the ultimate objects of scientific inquiry. They are the unified material objects, as well as the natural stuffs, identifiable in sense-experience, each taken to be a member of a natural species with its 'form' and functional essence. Entities in other categories – qualities, actions, relations, and so forth –

> are treated as dependent on, if not just abstracted aspects of, these independent realities.[40]

Somit ist Substanz mehr als nur ein bestimmtes Material. Man könnte vielleicht sagen, Substanz ist, worauf wir potentiell Referenz nehmen können und betrifft somit nicht nur differenzierbare materielle Objekte. Aristoteles unterscheidet zwischen erster und zweiter Substanz. “Jede Substanz scheint ein Dieses zu bezeichnen, und bei den ersten Substanzen ist es zweifellos und wahr, daß sie das tun. Das, worauf man hier hinweist, ist unteilbar und der Zahl nach eins.”[41] Aber referiert auch die zweite Substanz auf etwas Wirkliches? Aristoteles: “Bei den zweiten Substanzen aber wird zwar durch die Art der Benennung der Schein erweckt, als ob es ebenso wäre, wenn man vom Mensch oder Sinnenwesen spricht, aber es ist nicht wahr: vielmehr bezeichnet man in diesem Falle ein Qualitatives. Denn hier ist das Subjekt nicht eines wie die erste Substanz, sondern Mensch und Sinnenwesen wird von vielen Subjekten ausgesagt.”[42] Die Gattungen und Spezies haben, obwohl sie ’Qualitatives’ bezeichnen, aufgrund ihrer hierarchischen Einordnung hohen Aussagewert bezüglich der Frage ’Was ist etwas?’ und damit gewisse ontologische Selbstständigkeit. Deshalb sind sie von den Differenzen und Akzidenzien verschieden, die nur einem Zugrundeliegenden zukommen können. Für meine Untersuchung über Computersysteme wird jedoch mit dem Wegfall der hierarchischen Gliederung auch diese Sonderstellung als ’naturgegebene’ vorerst bedeutungslos.

Im Gegensatz zu den anderen Kategorien, wie zum Beispiel der Qualität

[40] Craig, Edward (Hrsg.): *Routledge Encyclopedia of Philosophy*. Band 9, New York: Routledge, 1998 S.205

[41] Aristoteles Cat. 3b, zitiert nach der Übersetzung von Eugen Rolfes. In: Aristoteles; Rolfes, Eugen (Hrsg.): Kategorien / Lehre vom Satz. Band 8/9, Philosophische Bibliothek. Hamburg: Verlag von Felix Meiner, 1962, S.49

[42] Aristoteles Cat. 3b, In: Aristoteles S.49

(die weißer, schöner, am schönsten und alle Abstufungen dazwischen sein kann), kann ich mit Aristoteles feststellen: “Die Substanz lässt also kein Mehr oder Minder zu.”[43] Denn Platon ist nicht mehr oder minder Platon und er ist auch nicht mehr oder minder Mensch beziehungsweise Sinnenwesen. “Aber die Substanz ist, obwohl der Zahl nach ein und das selbe, für Konträres empfänglich. So wird z.B. ein bestimmter Mensch, obwohl er einer und derselbe ist, bald weiß, schwarz, warm und kalt, schlecht und gut.”[44]

Quantität ist teils diskret, wenn sie eine Ordnung betrifft und teils kontinuierlich, wenn sie eine Lage betrifft. Wohl auch deshalb, weil Ordnungsangaben besser in natürlichen und Maßangaben genauer in reellen Zahlen ausgedrückt werden können. “In der Metaphysik V, 13.1020a 7ff. steht zuerst ihre Definition [...] : ’Quantitativ heißt, was so in Bestandteile zerlegbar ist, daß jeder davon, zwei oder mehrere, seiner Natur nach ein Eines und Dieses sein kann. Menge ist ein Quantitatives, wenn es zählbar, Größe, wenn es meßbar ist. Man nennt aber Menge, was potentiell in Nichtstetiges, Größe was in Stetiges zerfällt.”[45] Sowohl eine diskrete als auch eine kontinuierliche Quantität kann, sofern sie nicht schon eines ist, in mehrere Substanzen zerfallen. Das gilt sowohl für eine Menge (drei Steine von fünf Steinen wegnehmen) als auch für eine Größe (fünf Zentimeter von einem Stock abschneiden). Jedoch gilt dadurch für sie, ähnlich wie für die Substanz: “Die Quantität scheint aber kein Mehr und Minder zuzulassen, z.B. das zwei Ellen Lange nicht: das eine ist nicht mehr zwei Ellen lang als das andere. Auch gibt es kein Mehr und Minder bei der Zahl: so ist die Drei im Vergleich zu der Fünf nicht mehr Drei und die Fünf im Vergleich zu der Drei

[43] Aristoteles Cat. 4a, In: Aristoteles S.50
[44] Aristoteles Cat. 4a, In: Aristoteles S.50f
[45] Aristoteles Kommentar. S.84

nicht mehr Fünf. Auch kann man nicht sagen daß eine Zeit mehr Zeit sei als die andere, und spricht überhaupt bei keinem der genannten Dinge von einem Mehr und einem Minder. Mithin läßt auch die Quantität kein Mehr und Minder zu."[46] Eine weitere wichtige Eigenschaft des Quantitativen für meine Untersuchung ist die Folgende: "Am meisten ist es dem Quantitativen eigentümlich, daß es gleich und ungleich genannt wird. Denn jedes der angeführten Quantitativa wird gleich und ungleich genannt. So nennt man z.B. den Körper gleich und ungleich und ebenso die Zeit."[47]

Diese Ausführungen sind – jedenfalls in der Form meiner verkürzten Wiedergabe – was die 'wahre' Ordnung der Wirklichkeit betrifft, je nach präferierter Konzeption womöglich anzweifelbar. Allerdings wage ich zu behaupten, dass die Begriffe der Substanz, der Quantität und Qualität in der Konzeption des Aristoteles ein angemessenes Werkzeug für die Behandlung meiner Eingangsfrage darstellen. Wie oben bereits erwähnt, ist, von praktischen Gesichtspunkten (wie Herstellungsverfahren, Genauigkeit des zugrundeliegenden physischen Phänomens bei einem Analogrechner; erreichbare Schaltgeschwindigkeiten bei einem Digitalrechner etc.) abgesehen, weder das Material aus dem ein Computer hergestellt wird, noch die verwendete Technik um diskrete Zustände herzustellen, von Einfluss auf die Funktionalität. Und so gibt es gute Gründe, eine Einheit, der die Möglichkeit von zumindest zwei diskret unterschiedenen Werten (im Falle eines Digitalrechners) als ein und Derselben (z.B. Speichereinheit) zukommt, Substanz zu nennen. Denn es "[...] gilt die Substanz aus dem Grunde für das aufnehmende Prinzip des Konträren, weil sie selbst Konträres aufnimmt. Sie nimmt Gesundheit und Krankheit, Weiße und Schwärze auf, und man läßt sie für die konträren Gegensätze insofern

[46] Aristoteles Cat. 6a, In: Aristoteles S.55f
[47] Aristoteles Cat. 6a, In: Aristoteles S.56

empfänglich sein, als sie jedes Derartige selbst aufnimmt. Also muß der Substanz eigentümlich sein, daß sie, wiewohl der Zahl nach ein und dasselbe, für Konträres auf Grund ihrer eigenen Veränderung empfänglich ist."[48] Die Tatsache, dass wir einen Digitalcomputer ebensogut als Binärrechner (also zur Basis zwei) wie auch auf Basis des Zehnersystems (oder überhaupt anderen[49]) operieren lassen können, tut obiger Angabe keinen Abbruch – der Unterschied ist rein quantitativ. Wir gruppieren einzelne Substanzen zu größeren Einheiten. Also kann ich ein Byte aus 8 Bit basteln; ein Wort lassen wir aus zwei Byte bestehen usw. Wenn ich wie oben, nicht den Mengenaspekt sondern den Größenaspekt der Quantität betrachte, sehe ich, dass Selbiges auch für einen Analogrechner gilt. Welche der beiden Kontraritäten beziehungsweise welchen kontinuierlichen Größenwert eine Substanz angenommen hat, ist dann eine Frage der Qualität. "Unter Qualität (Beschaffenheit) verstehe ich das, vermöge dessen man so oder so beschaffen heißt."[50] Eine Qualität ist aber nichts Eigenständiges, sondern sie ist auf ein Zugrundeliegendes angewiesen. Sie enthält wiederum Kontraria und lässt in bestimmten Fällen auch ein 'mehr oder minder' zu. "Sodann wäre es auch, wenn ein und dasselbe relativ und qualitativ zugleich wäre, keine Ungereimtheit, daß das nämliche zu zweien Genera gehörte."[51] Weder erste noch zweite Substanzen sind relativ.[52] Relatives muss sich immer auf etwas Bekanntes beziehen; was keinen Bezug nötig hat, kann auch nicht relativ sein. "So ist eben alles relativ, dem eben das, was es begrifflich ist, im Vergleich zu einem anderen

[48] Aristoteles Cat. 4b, In: Aristoteles S.51f
[49] Vgl. Algorithmus im Kapitel über Berechenbarkeit 2.2.3
[50] Aristoteles Cat. 8b, In: Aristoteles S.63
[51] Aristoteles Cat. 11a, In: Aristoteles S.69
[52] Vgl. Aristoteles Cat. 8a-9a, In: Aristoteles S.61 – 63

oder in irgendeinem sonstigen Verhältnis zu anderem beigelegt wird."[53]
Es ist ersichtlich, dass für eine Berechnung, sofern zumindest irgendein Token[54], das von anderen unterscheidbar sein soll, wohl nicht in seiner Materialität fest bestimmt ist, aber in jedem Fall als eine Substanz da sein muss. "Die einzigen Bedingungen dafür, als manipulierbare Objekte algorithmischer Verfahren geignet zu sein, ist, daß die betreffenden Objekte deutlich voneinander abgesetzt sind, denn das Operieren besteht darin, räumlich oder zeitlich geordnete Dinge in neue Konfigurationen zu bringen."[55] Noch einfacher könnte man die Unterscheidung vielleicht so formulieren: Die Substanz eines bestimmten Apfels besteht darin, daß er genau einer ist, dass er wirklich ist, dass er unabhängig von uns da ist, dass wir auf ihn referieren können. Die Materialität eines Apfels versuchen wir mittels Molekülen (Wasser, Zucker, Vitamine, ...) Atomen, Quarks etc. zu begreifen; für eine vollständige Beschreibung, etwa zum Aufzählen aller Moleküle (was ja höchstens eine hinreichende Beschreibung genannt werden kann) eines bestimmten Apfels reichen unsere Möglichkeiten nicht aus. Außerdem wäre das Aufzählen der Atome für die meisten Anwendungsfälle, die Äpfel betreffen, bedeutungslos. Die Substanz eines bestimmten Bits besteht in einem Schalter, der zwei konträre Zustände annehmen kann. (Auf Werkzeuge zur Realisierung eines solchen werde ich noch zurückkommen.) Die Materialität eines Bits jedoch ist ebenso schwer angebbar wie die eines Apfels; zwar für Rechnengeschwindigkeit und Verarbeitungsaspekte möglicherweise ausschlaggebend, aber (von spekulativen Randproblemen abgesehen; siehe Hyper-

[53] Aristoteles Cat. 6b, In: Aristoteles S.57
[54] Engl. Zeichen oder Marke; kleinste sinngebende Einheit in einer Programmiersprache. Auch verwendet als Bezeichnung für ein Staffelholz für Abfolgen in Netzwerkumgebungen.
[55] Krämer: *Symbolische Maschinen* S.158

computing) für das Durchführen einer Berechnung auch ebenso bedeutungslos.

2.1.4 Substanz vs. Materialität

Bei Aristoteles dient die Kategorie der Substanz dazu, 'Seiendes' zu bezeichnen. Die anderen neun Kategorien bezeichnen Differenzen, die nicht selbständig und unabhängig von einem Substrat existieren können: "Die Substanz ist das erste und vornehmste und auch allein das eigentlich Seiende, weil sie an und für sich ist, während das andere, die Akzidenzien der neun anderen Kategorien, nur insofern sind, als sie sich an der Substanz und dem Seienden als Bestimmungen finden."[56] Nun ist, wie bereits erwähnt, Substanz kein Ausdruck für zugrundeliegende Materialität, sondern für Zugrundeliegendes im Sinne der hierarchischen Unterteilung der Kategorie. Aristoteles unterscheidet erste und zweite Substanz: "Von ersten Substanzen im Gegensatz zu zweiten Substanzen spricht Aristoteles darum, weil das einzelne Wirkliche, das selbstständig ist, wie Sokrates oder das bestimmte Tier, die und die bestimmte Pflanze, nur Subjekt ist und nie von einem anderen als Subjekt ausgesagt wird, während die zweiten Substanzen, die Arten und Gattungen, von den ersten Substanzen ausgesagt werden, wie z.B. in dem Satze: Sokrates ist ein Mensch, Sokrates ist ein sinnliches Wesen."[57] Eine Ansammlung von mehreren Bits zum Beispiel in einem Byte wäre demnach keine zweite Substanz, sofern ein digitaler Binärrechner dieser Sichtweise entsprechend überhaupt aus lediglich einer Art Substanz bestünde. Denn ein Byte besteht aus einem Quantum (in diesem Fall acht) von Bits, wird aber nicht von diesen aus-

[56] Aristoteles Kommentar. S.82
[57] Aristoteles Kommentar. S.82

gesagt. Zwar ist ein Bit, insofern es verwendet wird um eine Berechnung anzustellen, immer auch materialisiert; der Materialaspekt ist aber nicht dasjenige was es zum Bit macht.

> In Aristotele's view, matter is not a substance or an element. His own term for matter was reserved for one aspect of individual substances – particular objects like Sokrates – which he saw as having both 'matter' and 'form'. Matter has the potential to receive form; it is never found in absence of form, being inseperable from it.[58]

Somit ist Materie für Aristoteles eine Potentialität, der durch die Form Eigenschaften zukommen können; sie selber hat keine Eigenschaften. Er unterstellt seinen Vorgängern, etwa den Protagonisten der milesischen Schule, dass sie auf der Suche nach elementaren Bestandteilen, aus denen die Dinge gemacht seien, eine einzige materielle Substanz als Grundlage erkoren (Monismus). Zum Beispiel sieht Thales von Milet (624-546 v. Chr.) das Wasser, Anaximander (ca. 610-547 v. Chr.) das stofflich unbestimmte Apeiron und Anaximenes (585-525 v. Chr.) die Luft als den materiellen Grund- oder Urstoff alles Seienden an. Parmenides (ca. 540-475 v. Chr.) war der Ansicht, dass, wenn es nur eine einzige zugrundeliegende Substanz gäbe, auch keine Veränderung möglich wäre. Und nach dieser Zurückweisung der frühen Monismen, entstanden pluralistische Ideen, die die Annahme von mehr als einer Ursubstanz propagieren. Etwa Empedokles (494-434 v. Chr.), der eine Vier-Elemente-Lehre von Feuer, Erde, Luft und Wasser aufstellte. Graduell am weitesten entfernt von den Vorstellungen der Monisten, der Materie als Kontinuum, ist der antike Atomismus[59].

[58] Craig, Edward (Hrsg.): *Routledge Encyclopedia of Philosophy*. Band 6, New York: Routledge, 1998 S.192f

[59] vgl. Craig: *Routledge Encyclopedia of Philosophy* S.193

Leukipp, Demokrit und später Epikur vertraten Ansichten, denen zufolge die Welt aus kleinsten, diskreten und unteilbaren Teilen, den Atomen besteht. Eher durch einen Glücksfall stellten die Atomisten eine Hypothese auf, die mit leichten Abwandlungen heute noch aktuell ist. “Leukip glaubte, eine Theorie gefunden zu haben, die in Einklang mit der Sinneswahrnehmung stand und weder das Entstehen und Vergehen noch die Bewegung und die Vielfalt der Dinge ausschloß.”[60] Ein weiteres Problem stellte die Ansicht dar, dass in einem völlig ausgefüllten Raum keine Bewegung möglich wäre, was aber mit Parmenides’ Argumenten gegen das Nichtsein im Widerstreit stand. Bertrand Russell fasst Parmenides’ Auffassung so zusammen: “Sagt man, es gebe die Leere, dann ist der leere Raum nicht Nichts; daher ist es also auch kein leerer Raum.”[61] Der Ausweg besteht endlich darin, zwischen Materie und Raum zu unterscheiden, somit den Raum als ein Behältnis zu verstehen, das teilweise mit Materie ausgefüllt ist. “Aristotele rejected atomism on several grounds, including its dismissal of final causes in nature. Despite a revival of atomism by Lucretius, Aristoteles rejection of the doctrine persisted among his medieval followers.”[62].

Max Jammer beschreibt in seinem Buch ’Der Begriff der Masse in der Physik’ das Fehlen eines Begriffs für die Quantität der Materie (vgl. Masse) in der antiken Philosophie. Körper werden zwar als etwas, dem eine Größe, eine Gestalt, Widerstand und Schwere zukommt gesehen, aber eher im Sinne einer zukommenden Eigenschaft. “Diese Attribute einschließlich der Größe und der Gestalt sind, wie Plotin wiederholt erklärt, für diese antiken Philosophen Formen und nicht das Substrat, das Formen annimmt. Die

[60] Russell, Bertrand: *Philosophie des Abendlandes*. Wien: Europa Verlag, 2001 S.90
[61] Russell S.91
[62] Craig: *Routledge Encyclopedia of Philosophy* S.194

elementare Materie, die Materie an sich, kann nicht auf quantitative Weise beschrieben werden. Das steht in diametralem Gegensatz zur Newtonschen und zur modernen Physik. Denn diese findet ihre Aufgabe darin, die qualitativen auf quantitative Aspekte zurückzuführen, und sie sind Quantität keineswegs als eine besondere Art von Qualität (bzw. Form)."[63] Die Idee des spezifischen Gewichts existiert zwar schon in den Schriften des Archimedes, ist als Begriff aber nie von ihm verwendet, geschweige denn definiert worden. "Der Begriff der Dichte in seiner exakten Form als das Verhältnis von Masse und Volumen stammt aus neuerer Zeit, beginnend bei Leonhard Euler."[64]

Auch die frühe, moderne Wissenschaft im 17. Jarhundert, wesentlich durch Descartes geprägt, lehnt Ansichten, nach denen es mehrere verschiedene Typen von materieller Substanz gibt, ab: "... there is only one universal type of matter, with universal essential properties or 'attributes'. All material bodies [...] are to be understood in terms of the motion of this fundamental matter. [...] All action of matter must be caused by an external agent."[65] Descartes unterschied materielle Substanz, die sich im Raum ausdehnt, streng von dem, was er Geist oder mentale Substanz nennt, die wesentlich nicht-räumlich ist und im Gegensatz zur materiellen Substanz selbst Bewegung initiieren kann. Demnach ist, wie später bei Newton, Materie wesentlich eine passive Substanz. Allerdings, wo für Descartes Materie etwas Kontinuierliches, mit dem Raum Identisches ist, identifiziert Newton "atoms as contingent occupants of space and consisting of physically (rather than logically) indivisible atoms occupying only a small proportion

[63] Jammer, Max: *Der Begriff der Masse in der Physik*. Darmstadt: Wissenschaftliche Buchgesellschaft, 1974 S.25

[64] Jammer S.29

[65] Craig: *Routledge Encyclopedia of Philosophy* S.194

of the total space available in the universe."[66] Im weiteren Rückblick zeigt sich noch eine wichtige Veränderung: Während für Aristoteles ein Körper nur in Ruhelage als 'inert'[67] gilt, sind das für Newton und Descartes auch Körper in gleichförmiger Bewegung:

> [...] they agreed that it is of the nature of matter to persist in its inertial state of rest or uniform motion unless acted upon by an external force: that it is inert, dead, passive, blindly continuing in its present state, acting only when acted upon. [...] From most of the seventeenth-century thinkers, whether atomists or plenists, the basis of action in the universe was the impenetrability or hardness of matter, and action always took place by contact.[68]

Leibniz lehnte das Cartesianisch-Newtonsche Konzept der passiven Materie ab. "He argued that the merely passive, whether space, time or matter, whether atomistic, plenum, purely space, cannot account for the unity of things, their individuation, change, resistance or impenetrability, or action."[69] Reine Passivität kann nicht erklären, wie Dinge aufeinander einwirken und Undurchdringlichkeit ('impenetrability') ist selbst eine Form von Aktivität. Leibniz löst das Problem von Aktivität und Passivität, der Descarteschen Wechselwirkung von Geist und Materie durch das idealistische Konzept der Monade.

> Demgemäß glaubte er an eine Unzahl von Substanzen, die er 'Monaden' nannte. Jede von ihnen besäße einige Eigenschaften eines physikalischen Punktes, allerdings nur bei abstrakter

[66] Craig: *Routledge Encyclopedia of Philosophy* S.194
[67] träge; vgl. Trägheitsgesetz bei Newton
[68] Craig: *Routledge Encyclopedia of Philosophy* S.194
[69] Craig: *Routledge Encyclopedia of Philosophy* S.194

> Betrachtung; in Wirklichkeit ist jede Monade eine Seele. Das ergibt sich naturgemäß, wenn man die Ausdehnung als Attribut der Substanz verwirft; was allein als mögliches Wesensattribut übrigbleibt, schien das Denken zu sein. So kam Leibniz dazu, die Realität der Materie zu leugnen und durch eine unendliche Familie von Seelen zu ersetzen.[70]

Für Rene Descartes ist die Vorstellung eines leeren Raumes ebenso sinnlos wie die des Glücks ohne ein es erlebendes Wesen, da er wie die antiken Philosophen Ausdehnung als Attribut der Materie ansah. “Auch Leibniz glaubte, allerdings aus etwas anders gearteten Gründen, an den völlig ausgefüllten Raum, behauptete jedoch, der Raum sei nur ein System von Beziehungen.” Um diese Frage entstand eine bekannte Kontroverse zwischen Newton, mit seiner Konzeption des absoluten Raumes und Leibniz. Erst in der Neuzeit verhalf Einstein mit der Relativitätstheorie Leibniz zum Sieg.[71]

Durch die Entstehung der experimentellen Naturwissenschaft letztlich, löste sich der physikalische[72] vom ontologischen Begriff der Materie ab. Die

[70]Russell S.592

[71]Vgl. Russell S.92

[72]vgl. Materie: grundlegender physikalischer Begriff, der im Laufe der Physikgeschichte unterschiedliche Interpretationen erfuhr. Im Weltbild der klassischen Physik ist Materie eine von der Energie abgegrenzte meß- und berechenbare Größe, der Newtons Unterscheidung der trägen und schweren Masse materieller Körper zugrunde liegt. In der Speziellen Relativitätstheorie muß der Materiebegriff durch die Erkenntnis der endlichen Lichtgeschwindigkeit in der Elektrodynamik revidiert werden; insbesondere stellen sich Masse und Energie als äquivalent heraus. Die Äquivalenz von träger und schwerer Masse wird zur Grundlage der Allgemeinen Relativitätstheorie. Der Materiebegriff der Quantenmechanik untscheidet sich durch den Welle-Teilchen-Dualismus, die Heisenberg’sche Unschärferelation und das Superpositionsprinzip wesentlich von der klassischen Mechanik, und eine rein realistische Deutung der Quantenmechanik muß gewohnte Vorstellungen der Materie aus der klassischen Physik aufgeben. In den Quantenfeldtheorien wird der Begriff des Materiefeldes eingeführt, mit dem die Wechselwirkungen von Elementarteilchen beschrieben werden. Die Thermodynamik des 19. Jahrhunderts beschäftigte sich mit Materie zunächst unter dem Blickwinkel der Wärmelehre und leitete

Bedeutung des Ersteren speist sich wesentlich aus Bestimmungen, die durch Experimente, sowie den dazugehörigen Beobachtungen und Messungen im Hinblick auf das Gesamtsystem physikalischer Begriffe gewonnen werden. “Galileo Galilei zählt als primäre Qualitäten der Materie arithmetische (Zählbarkeit), geometrische (Gestalt, Größe, Lage, Berührung) und kinematische Eigenschaften (Beweglichkeit) auf; auch die Möglichkeit einer natürlichen Trägheit der Materie faßt er bereits ins Auge. Diese wird aber erst von J. Kepler im Zusammenhang der Erklärung der Planetenbewegung systematisch herangezogen”[73]. Lavoisier, einer der Väter der modernen Chemie stellt ein empirisch anwendbares Konzept für ’Elemente’ auf: Ein Material muss als eigenständiges Element behandelt werden wenn: “(a) with any of the means available to us, we cannot break it down into further constituents”, zweitens: “(b) through such experiments, we always find members of the same class of substances as constituents” und drittens: “(c) we find that other substances can be reconstructed by combinations of one or more of these ultimate breakdown products”[74]. Damit wurde eine Einteilung der ’toten’ Materie in chemische Elemente möglich. Die Möglichkeiten, mit denen Lavoisier und seine Zeitgenossen im 19. Jahrhundert auf Materie einwirken konnten, werden heute im Sinne von ’low-energy processes’ als ’chemisch’ im Unterschied zu den Kräften (’nuclear’ forces), die die Atomkerne selbst zusammenhalten, bezeichnet.

Auch die Theorien über die Mikrostruktur der Materie, die seit

aus der Äquivalenz von Wärme und Arbeit die beiden Hauptsätze der Thermodynamik ab. Der statistische Zugang L. Boltzmanns untersucht die Entstehung von Ordnung im thermischen Gleichgewicht. Die moderne Thermodynamik des Nichtgleichgewichts liefert den Ansatz für die Erklärung von Selbstorganisationsformen der Materie. Kilian, Ulrich. und Weber, Christine (Hrsg.): *Lexikon der Physik*. Band 3, Heidelberg: Spektrum Akademischer Verlag, 1999 S.450f

[73]vgl. Ritter, Joachim. und Gründer, Karlfried (Hrsg.): *Historisches Wörterbuch der Philosophie*. Band 5, Basel: Schwabe & CO AG Verlag, 1980 S.922

[74]vgl. Craig: *Routledge Encyclopedia of Philosophy* S.195

> der Akzeptierung von Daltons Atomtheorie beständig weiter ausgebaut und u.a. auf die elektrischen und elektrochemischen Erscheinungen sowie das Licht (Äthertheorie) ausgedehnt werden, bilden zunächst nur die Extrapolation der klassischen Vorstellungen in den Mikrobereich. Die Materie besteht danach aus kleinsten Teilchen – Atomen und Molekülen -, die als klassisch-mechanische Körper vorgestellt werden. Um die Wende zum 20. Jh. beginnen sich jedoch die Ergebnisse und Hypothesen zu mehren, die eine Auflösung der klassischen Grundvorstellungen bedeuten.[75]

Die spezielle sowie die allgemeine Relativitätstheorie, die nicht-euklidische Geometrie, die Thermodynamik und andere Entwicklungen des frühen 20. Jahrhunderts brechen die alten Vorstellungen auf. “While the concept of the indestructibility and conservation of mass is central to classical theories of mechanics and chemistry, relativity theories recognize that mass varies with frame of reference, and with velocity as measured in that reference-frame, and further, that mass is interconvertible with energy.”[76] Die Relativitätstheorie bringt das Ende der Newtonschen Konzeption der klassischen Mechanik, der absolute Zeit und des absoluten Raumes. Aber nicht nur, denn “bedenkt man die oben genannte Identität von Masse und Energie, so wird deutlich, daß von den klassischen ’ontologischen’ Differenzen keine einzige übriggeblieben ist – eine Feststellung, die nicht zuletzt für die philosophische Erkenntnistheorie ein Problem enthält.”[77] So wie die Relativitätstheorie im Großen bringen Feldtheorie und Quantenmechanik im Kleinen, also bezüglich der kleinsten bekannten Bausteine

[75] Ritter S.922
[76] Craig: *Routledge Encyclopedia of Philosophy* S.195
[77] Ritter S.923

der Materie (den Elementarteilchen), neue Erkenntnisse.

> The sharp contrasts between inert, passive permancence and dynamic activity, ultimate constituency and change, matter and force, fail to appear in quantum-theoretical analogues of classical matter. Indeed, the merely approximate analogy of fermions and bossons to classical concepts of matter, the activity of the quantum vacuum, the character of quantum fields as superpositions of possibilities and many other features of the quantum world, together with the existence of dark matter, all conspire to dictate the reformulation of many traditional problems of philosophy, such as the free will problem, the doctrines between actuality and possibility, and the topic of matter itself.[78]

Diese notwendigerweise extrem verkürzte Darstellung der Geschichte von Substanz und Materie – soll vorerst ausreichen, um zu zeigen, dass ich hier nicht voraussetzungslos meiner Forschungsfrage nachgehen kann.
Materie ist in den verschiedenen historischen Facetten ihrer Beschreibung immer vorrangig dies: eine Beschreibung, die möglichst dem aktuellen Wissensstand gerecht wird. Also eine Frage nach der Qualität (Wie ist etwas?). Und sie ist, im Gegensatz zur Substanz, vor allem nicht per Definition 'operational'. Substanz hat schon bei Aristoteles die Funktion etwas zu bezeichnen und kann zum Beispiel in einem Satz als logischer Operator fungieren. Die neuere Forschung, vor allem in Richtung Quantencomputer versucht eben dies aufzuholen, indem sie die – zumindest bis jetzt – kleinsten und ursächlichsten Phänomene in einer speziellen Laborsituation operational macht. Und zwar indem sie in einem Molekül (etwa Chloroform) die Überlagerung von Spins (Superposition) ein Qbit opera-

[78] Craig: *Routledge Encyclopedia of Philosophy* S.196

tional machen. Die Relaxationszeiten, sowie die Messungen liefern aber Warscheinlichkeitswerte und müssen mittels Fehlerkorrekturmaßnahmen an die Bedürfnisse angepasst werden.[79] Es ist demnach wiederum nicht Materie die Grundlage für die Berechnung sondern wesentlich 'hergestellte' Substanz. Das gilt sowohl für das Qbit als auch für einen traditionellen Rechner auf Siliziumbasis, der etwa durch einen Taktgeber und durch als Bit fungierende Schalter operational wird.

[79] Vgl. Görnitz, Thomas: *Quanten sind anders*. München: Spektrum Akademischer Verlag, 2006

2.2 Der Computer als symbolische Maschine

In diesem Kapitel stehen mathematische Aspekte und speziell der Maschinencharakter von Computern im Mittelpunkt. Die verkürzte Wiedergabe, der für meine Arbeit interessantesten Punkte aus Sibylle Krämers Arbeit über 'Symbolische Maschinen',[80] nimmt den meisten Raum ein. Den Einstieg bildet ein kurzer Ausflug in die Geschichte der Formalisierung, der Fokus dieses Kapitels liegt aber auf den entscheidenden Jahren zu Anfang des 20. Jarhunderts, der Entwicklung des Berechenbarkeitsbegriffes, dem damit verbundenen Algorithmenbegriff und der maschinellen Ausführung formalisierter Vorgänge. Begriffe wie 'Unvollständigkeitssatz', 'Entscheidungsproblem' und 'Halteproblem' werden eingeführt. Auch der Aufbau und die Funktionsweise von Turingmaschinen werden besprochen.

2.2.1 Aristoteles, Leibniz, Frege

Vom Weg zur Entwicklung von formalisierten Kalkülen möchte ich drei Stationen herausgreifen: Aristoteles und die Einführung von Variablen, Leibniz und die Anerkennung des formalen Beweises und letztlich Freges Einführung einer universellen Begriffschrift und der damit einhergehenden Neudefinition der Verbindung von Mathematik und Logik. Dabei muss nicht extra erwähnt werden, dass viele wichtige Entdeckungen und herausragende Persönlichkeiten dazwischen existieren, die hier nicht genannt werden.[81]
Der erste wichtige Schritt – noch lange vor den Entdeckungen der genann-

[80] Krämer: *Symbolische Maschinen*

[81] (Genaueres siehe: Krämer: *Symbolische Maschinen* oder, speziell die Entwicklungen um Leibniz im 17.Jarhundert betreffend, ihre Habilitation: Krämer, Sybille; Patzig, Günther. Scheibe, Erhard. Wieland Wolfgang (Hrsg.): *Berechenbare Vernunft*. Band 28, Quellen und Studien zur Philosophie. Berlin, New York: Walter de Gruyter, 1991.

ten Persönlichkeiten – auf dem Weg der Ablösung der Zeichen vom Bezeichneten, ist die Verwendung gegenständlicher Hilfsmengen beim Zählen. Steine, Stäbchen oder Finger dienen als Substitute. Dabei wird analog die Anzahl der Dinge auf die Hilfsmenge übertragen. Die dabei übertragene Struktur, entspricht der Aneinanderreihung von diskreten, homogenen Einheiten. Sie sind aber noch keine Zahlen (haben noch nicht die Bedeutung von 'Zahlen'), sondern einfache Anzahlen. Sie leiten zu rudimentären Bildungsgesetzen von 'natürlichen Zahlen' über. Die Ablösung ist erst dann wirklich vollzogen, wenn an die Stelle von analogischen Anzahlen Zahlzeichen treten. In Europa vollzog sich dieser Wandel relativ spät. Erst zwischen dem 14. und 15. Jahrhundert, mit der Einführung der arabischen Zahlen, wurde die Rechenbretttechnik durch das Ziffernrechnen abgelöst.[82] Im antiken Griechenland fand noch eine andere, wichtige Ablösung statt: Aristoteles wird durch die Idee, mit Termini schematisch zu operieren, also Variablen-Zeichen als Leerstellen einzusetzen, zum Schöpfer der modernen Logik. Er setzt für die Variablen Buchstaben ein, die logischen Operatoren belässt er als Worte (Verben und Prädikate). Aber in seinem System können die Variablen nur für Terme stehen, nicht für ganze Ausdrücke. "Die Variablenzeichen stehen nicht mehr für ein einzelnes Objekt, sondern für unspezifische Objekte eines wohlbestimmten Variabilitätsbereiches. [Und um ein wenig vorzugreifen:] Aristoteles und die stoischen Logiker führten die Variablen in die Logik, Vieta und Descartes führten sie in die Mathematik ein, auf Frege geht der Gebrauch der gebundenen Variablen zurück."[83] Aristoteles trennt aber streng zwischen dem Gebrauch von Wörtern und dem Rechnen mit Zahlen. Er erklärt gemischte Operationen, mit Wortzeichen und Zahlzeichen für ungültig, in-

[82]Vgl. Krämer: *Symbolische Maschinen* S.7f
[83]Krämer: *Symbolische Maschinen* S.182

dem er auf die Vieldeutigkeit der Wortzeichen verweist.
Erst Leibniz sieht, dass “formale logische Schlussweisen und das Rechnen analoge Prozeduren sind“[84]. Sowohl beim Rechnen als auch beim Schlussfolgern sind ’Wahrheitsbeweise’ auf ’Richtigkeitsbeweise’ reduzierbar. Leibniz kann zeigen, dass ein mathematischer Beweis ausschließlich Kraft seiner Form richtig ist, denn er findet nur auf dem Papier und im Bezug auf die Zeichen statt. Er wird unabhängig von der Sache abgehandelt. “Formales Denken [...] beruht auf der Möglichkeit, das Operieren mit Gedanken zu ersetzen durch das Operieren mit Zeichenmustern, so daß alle Regeln, nach denen der Aufbau und die Veränderung der Zeichenmuster sich vollzieht, keinen Bezug mehr nehmen auf den Inhalt der Gedanken, sondern nur noch auf die Strukturen der Muster selbst.”[85]
Sibylle Krämer fasst Leibniz’s Versuche, den Kalkül zum philosophischen Erkenntnisgewinn zu instrumentalisieren unter dem Terminus ’Das Leibnizprogramm’ zusammen. Dieses gliedert sich in drei Aspekte: (1) Die Idee einer Universalwissenschaft, (2) die Idee einer universellen Kalkülsprache, und (3) die Idee logischer Kalküle im engeren Sinne.[86]

> Das Leibnizprogramm ist der erste Versuch, die epistemische Systemrationalität des axiomatisch-deduktiven Theorieaufbaus mit der technischen Handlungsrationalität algorithmischer Erzeugungsprozeduren zu verbinden. Durch den Aufbau einer ’ars characteristica’, einer universalen Kalkülsprache der Wissenschaft, die beliebige Sachverhalte und ihre Beziehungen formal auszudrücken gestattet, und einen ’calculus ratiocinator’, der die Folgerungsbeziehungen zwischen den Aussagen

[84] Krämer: *Symbolische Maschinen* S.101
[85] Krämer: *Symbolische Maschinen* S.102
[86] Vgl. Krämer: *Symbolische Maschinen* S.100ff

> der 'ars characteristica' als Formations- und Transformationsregeln eines Kalküls faßt, soll es möglich werden, alle wahren Sätze automatisch herzuleiten sowie über die Wahrheit jedes vorgelegten Satzes zu entscheiden. [...] Im Leibnizprogramm wird die maschinenmäßige Erzeugbarkeit eines wissenschaftlichen Satzes zum Kriterium seiner Wahrheit.[87]

Von Leibniz kommt die Idee, ein formalisiertes logisches System zu kreieren, aber erst George Boole schafft eine systematische Ausarbeitung der symbolischen Logik. Später knüpft Gottlob Frege mit seiner 'Begriffsschrift' an zwei Ideen von Leibniz an. Nämlich (1) "dass das Rechnen Beweisfunktion zu erfüllen habe." und (2) "dass dies nur auf der Grundlage einer künstlichen Zeichensprache möglich sei, die ideographisch verfahre, also nicht mehr gesprochene Wörter sondern Begriffe verschriftliche"[88]. Dieser konstruierende Ansatz verhalf ihm zur Überwindung der traditionellen 'Subjekt-Prädikat-Struktur' von logischen Urteilen. Die Variablen der symbolischen Logik sind nicht mehr nur Platzhalter für sprachliche Ausdrücke, sondern sie fungieren als Grundzeichen von Kalkülen. "Damit werden die Variablen nicht durch Abstraktion gewonnen, sondern sind Resultat eines operativen Verfahrens, welches die Konstruktion von Kalkülen zur Voraussetzung hat. Die logische Geltung von Aussagezusammenhängen wird damit zu einer Frage ihrer Herleitbarkeit innerhalb des formalen Systems."[89] Das führte zur Einführung von Quantoren und der damit verbundenen Unterscheidung zwischen gebundenen und freien Variablen. Die Orientierung der logischen Analyse am umgangssprachlichen Sprachgebrauch ist damit Geschichte.

[87] Krämer: *Symbolische Maschinen* S.179f
[88] Krämer: *Symbolische Maschinen* S.132
[89] Krämer: *Symbolische Maschinen* S.136

Frege betont in seinem Aufsatz 'Über die wissenschaftliche Berechtigung einer Begriffsschrift' den Vorteil der Schriftlichkeit gegenüber der flüchtigen mündlichen Sprache für die wissenschaftliche Betrachtung und deren Überprüfbarkeit. Im Bereich der schriftlichen Form ist wiederum ein Unterschied zwischen den aufgeschriebenen Lautfolgen der Umgangssprache und einer arithmetischen Formelsprache zu machen, in der ein einfaches Urteil in einer Zeile untergebracht werden kann. Die zweifache, plane Ausdehnung der Schreibfläche wird genützt: "Wenn aus zweien ein dritter folgt, trennt man den dritten durch einen horizontalen Strich, der mit 'folglich' übersetzt werden kann, von den beiden ersten."[90] Sobald die logische Verknüpfung komplexer sein soll, ist es notwendig, auf Worte zurückzugreifen. Deshalb kommt der arithmetischen Formelsprache, der die Ausdrücke dafür fehlen, noch nicht vollends der Name 'Begriffsschrift' zu. "Gerade umgekehrt ist es bei der von Leibniz herrührenden Bezeichnungsweise logischer Beziehungen, die in neuerer Zeit von Boole, R. Graßmann, St. Jevons, E. Schröder und anderen erneuert worden ist. Hier hat man zwar die logischen Formen, obwohl nicht ganz vollständig; es fehlt aber der Inhalt."[91] Frege meint, dass jeder Versuch, an die Stelle der Buchstaben ganze Ausdrücke oder Gleichungen zu setzen, zu Unübersichtlichkeit oder sogar Vieldeutigkeit der entstehenden Formeln führen würde. Deshalb fordert er für eine Begriffschrift:

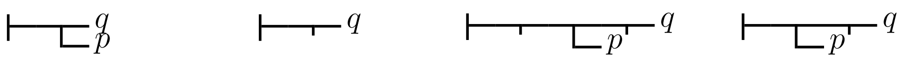
> Sie muß für die logischen Beziehungen einfache Ausdrucksweisen haben, die, an Zahl auf das Notwendigste beschränkt, leicht und sicher zu beherrschen sind. Diese Formen müssen geeignet sein, sich mit einem Inhalte auf das Innigste zu verbin-

[90] Vgl. Frege, Gottlob; Patzig, Günther (Hrsg.): *Funktion, Begriff, Bedeutung*. Band 1144, Kleine Vandenhoeck-Reihe. Göttingen: Vandenhoeck & Ruprecht, 1975 S.95f

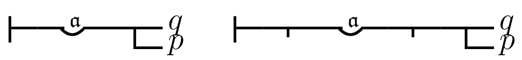
[91] Frege S.96

den. Dabei muß solche Kürze erstrebt werden, daß die zweifache Ausdehnung der Schreibfläche für die Übersichtlichkeit der Darstellung gut ausgenützt werden kann. Die Zeichen von inhaltlicher Bedeutung sind weniger wesentlich. Wenn die allgemeinen Formeln einmal vorhanden sind, können jene leicht nach Bedürfnis geschaffen werden. Wenn es nicht gelingt einen Begriff in seine letzten Bestandteile zu zerlegen, kann man sich mit vorläufigen Zeichen begnügen.[92]

So besteht die Begriffsschrift in einer graphischen, sich in zwei Dimensionen erstreckenden Notation, in der die Formeln (Konditional, Verneinung, Konjunktion und Disjunktion) durch senkrechte und waagrechte Verbindungen und Quantoren durch eine Höhlung an der Waagrechten bezeichnet werden. Im Text mit dem Titel 'Funktion und Begriff' führt Frege die zentralen Begriffe seiner 'Begriffsschrift' noch einmal sukzessive, anhand von Beispielen ein. Ein senkrechter Strich am Beginn markiert eine Behauptung. Da die Begriffsschrift ihre Terme im horizontalen Verlauf der Linie staffelt kommt sie ohne Klammern aus. Dies sind die aussagenlogischen Grundelemente[93]:



$(p \to q)$	$(\neg q)$	$(p \wedge q)$	$(p \vee q)$
Das Konditional	Die Verneinung	Die Konjunktion	Die Disjunktion



$\bigvee$	$\bigwedge$
Der Allquantor	Der Existenzquantor

[92]Frege S.96f
[93]Vgl. Frege S.34ff

Neben George Boole und Ernst Schröder ist Gottlob Frege einer von jenen Mathematikern, die durch die Aufarbeitung der Logik den Grundstein zur Erforschung der Grundlagen der Mathematik legten. “Um ein Bild zu benutzen: Frege wollte nicht – wie die algebraischen Logiker – ein weiteres Stockwerk bauen am Hause der Mathematik, sondern die Fundamente dieses Hauses sicherstellen, indem er zeigte, daß nicht die Logik als ein Zweig der Mathematik, sondern die Mathematik als ein Zweig der Logik zu gelten habe; die Gesetze der Arithmetik also auf diejenigen der Logik zurückzuführen seien.”[94] Durch die Kalkülisierung der Aussagenlogik, kann die Logik als eigenständiges axiomatisch deduktives System, ohne die Mathematik – und damit sich selber – voraussetzen zu müssen, bestehen.[95]

Frege war damit Begründer eines neuen mathematikphilosophischen Logizismus, demzufolge sich die Sätze der Arithmetik auf logische Wahrheiten zurückführen lassen. In seinem späteren Werk ’Die Grundgesetze der Arithmetik’ sollte ein solches formales System erarbeitet werden, es scheitert aber an der Russellschen Antinomie, einem Einwand, den ihm Bertrand Russell per Brief übermittelt. Frege, der sein Lebenswerk gescheitert sieht, zieht sich resigniert zurück, er hat jedoch eine Basis geschaffen, auf der Andere aufbauen konnten und können.

2.2.2 Wendepunkt im Formalisierungsstreben

Einen Wendepunkt in der Geschichte der Rechenmaschinen und damit auch einen Schwerpunkt dieser Untersuchung markiert die Generation um Alonzo Church, Alan Turing und Kurt Gödel. Formalistische Unternehmun-

[94] Krämer: *Symbolische Maschinen* S.131
[95] Vgl. Krämer: *Symbolische Maschinen* S.137

gen, die die ganze Welt mit Hilfe der Logik zu fassen versuchten (Weltformel), scheiterten an Gödels Unvollständigkeitssatz und Church's These, beziehungsweise Turings Mechanisierung berechenbarer Zahlen.
Gödel zeigte, dass jedes formalisierte System der Arithmetik wesentlich unvollständig ist. Es gibt innerhalb eines widerspruchsfreien Axiomensystems immer Sätze, die aus ihm selbst heraus nicht bewiesen oder widerlegt werden können. Zu diesen Sätzen gehört auch immer der Satz: 'Das formale System ist widerspruchsfrei.' Wenn man jetzt das Axiomensystem so erweitert, dass auch dieser Satz entscheidbar (also bewiesen oder widerlegt werden könnte) wird, so steht man notgedrungen vor einem 'stärkeren' Axiomensystem, das erst wieder des selben Beweises bedarf. Der von Gödel dazu konstruierte Beweisgang konnte unentscheidbare Sätze in der 'Principia Mathematica' freilegen.
Die drei, zwischen 1910 und 1913 von Bertrand Russell und Alfred North Whitehead herausgegebenen Bände der 'Principia Mathematica' sind das Ergebnis des Versuchs, alle mathematischen Wahrheiten aus einem wohldefinierten Satz von Axiomen abzuleiten. Darin wird das Programm Freges, wie es im Hilbertprogramm vorgeschlagen worden war, neu in Angriff genommen. David Hilberts Ziel war es, ähnlich wie Frege, die von Leibniz angestrebte Axiomatisierung der ganzen Wissenschaft, zumindest auf dem Teilgebiet der Mathematik umsetzen zu können. Die Widersprüche in Freges Mengentheorie (siehe Russellsche Antinomie) wurden durch eine Typentheorie gelöst. Es konnte gezeigt werden, dass sich im Prinzip die ganze Mathematik aus dem Formalismus der Principia Mathematica entwickeln lässt. Gödels Beweis zeigt lediglich, dass das Axiomensystem nicht abgeschlossen ist.

Sibylle Krämers zeigt in ihren Ausführungen[96], dass Kurt Gödel auf einen Gedanken zurückgreift, der als ’Richardsche Antinomie’ bekannt ist. Der Mathematiker Jules Antoine Richard stellte eine fiktive Reihenfolge der möglichen arithmetischen Eigenschaften der natürlichen Zahlen auf. Diese Eigenschaften werden als Sätze in einer beliebigen Sprache formuliert und wiederum jeweils einer natürliche Zahl zugeordnet, um damit eine Reihenfolge festzusetzen. Das Ordnungskriterium, ob einem Satz eine hohe oder niedrige Zahl zugewiesen wird, stellt dabei die Buchstabenanzahl der jeweiligen Sätze dar. Die Zahl 1 wird dem kürzesten Satz zugewiesen. Die Sätze lauten etwa: ’n ist eine Primzahl’ oder ’n ist das Produkt aus vier und sechs’. Da nun die Sprache beliebig gewählt ist, kommt es vor, dass der zugeordneten Zahl die Eigenschaft wirklich zukommt, und manchmal auch nicht. Nun definiert Richard, dass alle Zahlen denen die zugeordnete Eigenschaft nicht entspricht, ’Richardsche Zahl’ genannt werden. Mit anderen Worten: es kommt einer natürlichen Zahl die Eigenschaft zu, eine ’Richardsche Zahl’ zu sein, wenn die ihr vorher zugeordnete Eigenschaft nicht auf sie zutrifft. Damit ist es aber auch möglich diese neu gewonnene Eigenschaft in obige Reihe einzuordnen als: ’n ist eine Richardsche Zahl’. Aber wenn irgendeine natürliche Zahl n ebendiese Eigenschaft zugeordnet bekommt, ergibt sich die Antinomie: “n ist dann und nur dann eine Richardsche Zahl, wenn n keine Richardsche Zahl ist: Der Satz ’n ist eine Richardsche Zahl’ ist zugleich wahr und falsch.”[97] Das Problem im Richardschen Beispiel besteht darin, dass die Eigenschaft, eine Richardsche Zahl zu sein, nicht zu den arithmetischen Eigenschaften gehört, und daher einer Metasprache – Sätzen über die Arithmetik – angehört. Dieses Problem tut hier nichts zur Sache, denn Gödel entwickelt aus dieser

[96]Vgl. Krämer: *Symbolische Maschinen* S.146ff
[97]Krämer: *Symbolische Maschinen* S.148

Idee einen arithmetischen Kalkül, in dem er jedem Elementarzeichen, jeder Formel und jedem Beweis aus der Arithmetik genau eine bestimmte natürliche Zahl, eine 'Gödelzahl' zuweist. Damit lässt sich jeder logisch-mathematische Ausdruck des Kalküls einer endlichen Folge von natürlichen Zahlen zuordnen. "Die Gödelisierung ist also ein Verfahren, das es ermöglicht, durch eindeutige Zuordnung einer natürlichen Zahl zu einem formalen Ausdruck eines Systems, formale Systeme innerhalb der Zahlentheorie darzustellen und abzuhandeln."[98]

Gödel konstruiert für seinen Beweis eine arithmetische Formel $\mathfrak{F}$, die ihre eigene Unbeweisbarkeit besagt ('Die Formel $\mathfrak{F}$ ist nicht beweisbar'). Diese wird wiederum, ähnlich wie bei der Richardschen Antinomie, diesmal aber ohne ungültige Vermengung von Arithmetik und Metasprache, einer Zahl z zugewiesen und lautet: 'Die Formel mit der Zahl z ist nicht beweisbar'. Wenn jetzt die Formel $\mathfrak{F}$ innerhalb des Systems beweisbar wäre, wäre auch ihr Gegenteil bewiesen. Entweder ist ein solches Axiomensystem widersprüchlich (sowohl ein Ausdruck als auch seine Verneinung sind ableitbar), oder das Axiomensystem ist widerspruchsfrei und die Formel $\mathfrak{F}$ eben darin nicht entscheidbar. Damit zeigt Gödel, dass die Arithmetik prinzipiell unvollständig ist. Selbst wenn man weitere Axiome einführt, ließen sich wiederum formal unentscheidbare Sätze finden.[99]
Es ist aber zu beachten: "Gödel gelangt in seinem Beweisgang zu dem metamathematischen Satz: 'Wenn die Arithmetik widerspruchsfrei ist, ist sie unvollständig'."[100] Das besagt nicht, dass die Widerspruchsfreiheit der Arithmetik ausgeschlossen ist. "Unmöglich ist lediglich der Beweis, der auf

[98] Krämer: *Symbolische Maschinen* S.150
[99] Vgl. Krämer: *Symbolische Maschinen* S.151f
[100] Krämer: *Symbolische Maschinen* S.152

formale Operationen innerhalb des arithmetischen Kalküls abgebildet werden kann. Dies aber wäre ein Beweis, der sich nur der 'systemeigenen' Mittel bediente."[101]

In der Folge stellt Church einen Beweis für die Unentscheidbarkeit des allgemeinen Prädikatenkalküls auf. 1936 gelang es ihm zu zeigen, dass die Menge aller allgemeingültigen Ausdrücke im Prädikatenkalkül erster Stufe nicht entscheidbar ist. "Es gibt also kein mechanisches Verfahren, um die Zugehörigkeit einer Formel zur Klasse der gültigen Formeln der Prädikatenlogik erster Stufe zu bestimmen."[102] Etwa zur selben Zeit gelangte Alan Turing mit seinem Modell der Turingmaschine zu einem ähnlichen Ergebnis. Turings Theorem zeitigte ein Konzept für eine universale Rechenmaschine, die jeden Algorithmus – also eine formalisierbare Vorschrift, die auch ein Mensch abarbeiten könnte – berechnen kann. Die Formalisierung eines Systems besagt, dass alle Operationen innerhalb des Systems in Form von Regelanweisungen angegeben werden können, die auch von einer Maschine ausgeführt werden könnten. Diese Regeln entsprechen dem Programm, das eine Maschine abarbeitet. Die Werte, auf welche die Regeln anzuwenden sind, heißen Daten. Alles, wofür ein Algorithmus aufgestellt werden kann, kann durch eine Maschine simuliert werden. Nach den Beweisen von Gödel und Church ist klar, dass "keine Maschine gebaut werden kann, die alle wahren Sätze eines formalisierten Systems mechanisch herzuleiten gestattet."[103] Und zwar deshalb, weil es aufgrund des Unvollständigkeitstheorems prinzipiell nicht möglich ist. Hier kommen zwei weitere Begriffe ins Spiel, die eigentlich nur zwei unterschiedliche Formulierungen des selben Grundproblems in verschiedenen Bereichen

[101] Krämer: *Symbolische Maschinen* S.153
[102] Krämer: *Symbolische Maschinen* S.153
[103] Krämer: *Symbolische Maschinen* S.155

darstellen.
Hinter dem Begriff 'Entscheidungsproblem' steht die Frage, ob es möglich ist, einen Algorithmus anzugeben, der in einem Axiomensystem die Allgemeingültigkeit von beliebigen Ausdrücken feststellen kann. Anders formuliert: zu entscheiden, ob ein in den Begriffen des Systems (oder der Theorie) formulierter Satz innerhalb des selben Systems bewiesen werden kann oder nicht. Gödels Unvollständigkeitstheorem besagt, dass genau dies nicht möglich ist. Es gibt in einem geschlossenen Axiomensystem immer Aussagen, die nicht aus ihm selbst heraus entscheidbar sind.
Das 'Halteproblem', der zweite Begriff, ist eine auf Rechenmaschinen umgelegte Formulierung des Entscheidungsproblems. Diese fragt ob es möglich ist, einen Algorithmus zu definieren, der für eine beliebig gegebene Turingmaschine berechnen kann, ob sie jemals anhält (siehe Kapitel 2.2.4). Wie Alan Turing gezeigt hat, ist auch das – äquivalent zu Obigem – nicht möglich.
Für die Generation um Turing, Church und Gödel waren Arbeiten über Computersysteme noch von einer primär mathematisch-logische Herangehensweise geprägt und die Möglichkeiten für Berechnungen waren für sie primär – abgesehen von der Zeit- und Speicherplatzbeschränkung – an innerlogische Beschränkungen gebunden. Die von ihnen bewiesenen Grenzen des Berechenbaren sind zwar allgemeiner Natur, schließen das Gebiet aber keineswegs ab und es gibt einige AutorInnen, die unter dem Stichwort 'Hypercomputation' die Meinung vertreten, dass Berechnungen über die Limitierungen hinaus denen eine Turingmaschine unterliegt, möglich sind. Obwohl die Grenze des Berechenbaren, nicht zuletzt durch technische Neuerungen, ständig in Veränderung begriffen ist, gibt es bis jetzt noch keine Evidenz für 'Hypercomputation'. Das Kapitel 2.3.2

wird sich diesen Spekulationen kurz widmen. Hier bleibt abschließend noch zu sagen: “Die Bedeutung der Überlegungen von Gödel, Church und Kleene besteht vielmehr darin, daß sie uns zeigen, daß das Tun des Mathematikers niemals vollständig auf das Berechnen zurückzuführen ist.”[104]

2.2.3 Algorithmus und Berechenbarkeit

Obwohl ich den Begriff schon mehrmals verwendet, und einige Einschränkungen aufgezeigt habe, ist keineswegs so klar, was ein Algorithmus ist. Sybille Krämer schreibt dazu: “Mitte der Dreißiger haben mehrere Mathematiker, teilweise unabhängig voneinander, formale Definitionen des Begriffs ’Algorithmus’ im Sinne des Begriffs der berechenbaren Funktionen vorgeschlagen: Herbrand und Gödel entwickelten das Konzept der allgemein-rekursiven Funktionen; Church und Kleene präzisierten die Berechenbarkeit mit der λ-Definierbarkeit und Kleene führte den Begriff der partiell rekursiven Funktionen ein; Turing entwickelte den Begriff der automatischen Maschine, heute Turingmaschine genannt. In ihrem formalen Aufbau sehr verschieden, haben sich diese Definitionen mathematisch als äquivalent erwiesen.”[105] Weitere Formulierungen folgten später, zum Beispiel von Markov und Post. Ich werde weiter unten noch zu den formalen Definitionen zurückkommen. Zuerst möchte ich auf eine praxistaugliche Beschreibung (intuitiver Algorithmenbegriff) eingehen. Ein Algorithmus hat nach Sibylle Krämers Auflistung vier Eigenschaften: Elementarität, Determiniertheit, Allgemeinheit und Endlichkeit. Er ist insofern ’elementar’, als er in kleinste, bestimmte Grundoperationen zerlegbar ist und diese in ei-

[104]Krämer: *Symbolische Maschinen* S.157
[105]Krämer: *Symbolische Maschinen* S.158

ner eindeutigen Abfolge von Schritten abgearbeitet werden können. Er ist insofern 'determiniert', als die Regeln so eindeutig vorgeschrieben sind, dass derjenige, der sie befolgt, kein Wissen zu ihrer Bearbeitung benötigt, sondern die Regeln einfach nur stur abarbeitet. Er ist insofern 'allgemein', als er nicht auf eine bestimmte Materialität festgelegt ist, und auch nicht für nur ein Problem, sondern für eine ganze Klasse von Problemen eine Lösung darstellt. Er ist insofern 'endlich', als er in einer endlichen Folge von Buchstaben niedergeschrieben werden kann, und nach endlich vielen Schritten zu einem Ergebnis kommt. Damit ist jede Handlungsfolge, die diese vier Merkmale aufweist, auch algorithmisierbar.
Im Prinzip kann ein Algorithmus jegliche Art von Objekten manipulieren. Das ergibt sich schon aus den Bedingungen für ein solches Verfahren überhaupt: "Die einzigen Bedingungen dafür, als manipulierbare Objekte algorithmischer Verfahren geeignet zu sein ist, daß die betreffenden Objekte deutlich voneinander abgesetzt sind, denn das Operieren besteht darin, räumlich oder zeitlich geordnete Dinge in neue Konfigurationen zu bringen."[106] Im Normalfall repräsentieren diese Objekte ein Alphabet, eine Zeichenreihe auf der operiert werden kann. Eigentlich reicht ein Alphabet, das ein einziges Symbol beinhaltet. Auf dieser (unären) Basis (etwa x) können, abgesehen von der leeren Menge, die Worte x, xx, xxx, xxxx, ... erzeugt werden, die wir mit den natürlichen Zahlen in Verbindung setzen können. Eine ähnliche Abbildung in die natürlichen Zahlen haben wir oben bei der Gödelisierung schon angetroffen. So ist es im Grunde egal, welches Alphabet wir verwenden, "denn zu jedem Alphabet A kann durch Gödelisierung ein Alphabet A' gebildet werden, so daß ein zu dem ursprünglichen Algorithmus isomorpher Algorithmus entsteht. Das aber

[106] Krämer: *Symbolische Maschinen* S.158

heißt: Alphabete, auf die Algorithmen anwendbar sind, sind immer arithmisierbar."[107]

Mit dieser Darstellung von Algorithmizität habe ich nun eine Vorschrift, um maschinenverarbeitbare Regeln aufzustellen. Es handelt sich dabei aber nicht um ein Kriterium für Berechenbarkeit. Man kann sich eine Maschine, die einen Algorithmus ausführt als eine 'Blackbox' vorstellen. Als eine Maschine, die einen Satz von Eingangswerten, in einen Satz von Ausgangswerten aufgrund von fixierten Regeln überführt. Dabei ist der Algorithmus die rekonstruierte Vorschrift, wie die Elemente der Menge A der Menge B zugeordnet werden. Eine solche Zuordnung nennt man auch eine 'Funktion' mit Definitionsbereich und Wertebereich. Wenn es einen Algorithmus gibt, der die Werte der Funktion in endlich vielen Schritten ermitteln kann, so ist er gleichzeitig eine äquivalente Definition der Funktion, und die Funktion ist damit berechenbar. "Immer wenn es zu einer Funktion einen (abbrechenden) Algorithmus gibt, so handelt es sich um eine berechenbare Funktion."[108] Es gibt aber auch Funktionen, die nicht mit algorithmischen Mitteln berechnet werden können. Damit haben wir die Möglichkeit festzustellen, dass eine bestimmte Funktion berechenbar ist, indem wir einen angemessenen Algorithmus finden, der die entsprechenden Werte berechnet. Eine solche Vorgangsweise ist jedoch bloßes Ausprobieren und führt nicht zu einer allgemeinen Bestimmung. "Wenn wir sagen, eine Funktion, zu der es einen Algorithmus gibt, nennen wir berechenbare Funktion, berechenbare Funktionen aber sind programmierbar, so kann diese Aussage im strengen Sinn nicht mathematisch bewiesen werden, da der Begriff der berechenbaren Funktionen eher ein intuitiver,

[107]Krämer: *Symbolische Maschinen* S.161
[108]Krämer: *Symbolische Maschinen* S.164

nicht aber mathematisch präziser Begriff ist."[109]

Der Begriff der 'berechenbaren Funktionen' setzt die mathematische Präzisierung äquivalenter Begriffe wie 'algorithmisierbar' und 'berechenbar' voraus. Ihnen kommen wir durch die Theorie der rekursiven Funktionen und die Turingmaschine näher. "David Hilbert sprach 1926 die Vermutung aus, daß jede primitiv-rekursive Funktion berechenbar sei"[110]. Primitiv-rekursive Funktionen sind intuitiv berechenbar. Sie werden durch einfache Grundfunktionen und eine einfache Rekursionsschleife gebildet. Man kann sie auch als Funktion sehen, die sich mit dem Ergebnis des ersten Durchlaufes als neuem Eingangswert erneut selbst aufruft, bis sie bei Erreichen des Abbruchkriteriums nach endlich vielen Rekursionen anhält. Krämer bringt als Beispiel den Euklidischen Algorithmus zum Auffinden des größten gemeinsamen Teilers (ggT), aber auch ein schriftliches Ausführen der Grundrechnungsarten kann als rekursive Funktion angesehen werden. Die Klasse der primitiv-rekursiven Funktionen enthält aber nicht alle 'intuitiv berechenbaren' Funktionen. Die Ackermannfunktion etwa wurde (ebenfalls 1926) von Wilhelm Ackermann konstruiert, um Hilberts Vermutung zu widerlegen. Man spricht bei Funktionen dieser Art von μ-rekursiven Funktionen oder partiell-rekursiven Funktionen.

> Was Hilbert für den Zusammenhang von Berechenbarkeit und primitiv-rekursiven Funktionen vermutete, trifft in Bezug auf die μ-rekursiven Funktionen zu. Alle bisher bekannten berechenbaren Funktionen sind μ-rekursiv, und es besteht daher die begründete Vermutung, daß alle μ-rekursiven Funktionen berechenbar sind.[111]

[109] Krämer: *Symbolische Maschinen* S.164
[110] Krämer: *Symbolische Maschinen* S.166
[111] Krämer: *Symbolische Maschinen* S.168

Von der Klasse der μ-rekursiven Funktionen konnte später gezeigt werden, dass sie äquivalent zu den von Herbrand, Gödel und Kleene entwickelten, durch Gleichungssysteme definierten, allgemein-rekursiven Funktionen sind. Diese zeigten sich dann wiederum äquivalent zu der 1936 von Kleene und Church entwickelten Präzisierung der Berechenbarkeitsbegriffes mittels λ-Definierbarkeit (λ-Kalkül). Und diese wiederum, zum Berechenbarkeitsbegriff der Turingmaschine. Es stellte sich heraus, dass die aus unterschiedlichen Ausgangspositionen formulierten Begriffe für Berechenbarkeit übereinstimmen. Daraufhin formulierte Alonzo Church eine These, die besagt: "Der intuitiv gegebene, allgemeingebräuchliche Begriff der berechenbaren arithmetischen Funktion ist identisch mit dem exakt definierten Begriff der allgemein-rekursiven Funktion."[112] Sie wurde später als 'Church's-Thesis' bekannt. Es handelt sich dabei nicht um eine bewiesene Behauptung, sondern eher um eine fundierte Meinung. Denn der eher vage Begriffe der 'intuitiven Berechenbarkeit' entzieht sich der mathematischen Definierbarkeit. Dennoch wird sie in der Erfahrung insofern bestätigt, als sich alle bisher berechenbaren Funktionen als allgemein-rekursiv erwiesen.

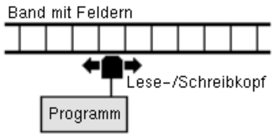
2.2.4 Die Turingmaschine

Die letzte, zwar bereits erwähnte, aber noch nicht behandelte Präzisierung des Berechenbarkeitsbegriffes ist die Turingmaschine. Sie ist die aufgrund ihrer Mechanizität wahrscheinlich greifbarste 'Zugangsweise'. Alan Turing entwirft seine Maschine anhand einer genauen Analyse der Schritte, die ein menschlicher Rechner (sofern das Verfahren bereits klar ist) ausführt.

[112] Krämer: *Symbolische Maschinen* S.169

Das Herunterbrechen der Schritte, die ein Mensch mit Bleistift und Papier ausführt, auf elementare Operationen, ermöglicht es, diese in einer einfachen Maschine zu implementieren: “Wir können einen Mann, der gerade eine reelle Zahl berechnet, mit einer Maschine vergleichen, die nur über eine endliche Zahl von Zuständen q_1, q_2, ..., q_R verfügt, die ihre ’m-Zustände’ heißen sollen.”[113] Sie arbeitet – analog zum Papier, das dem Menschen zur Verfügung steht – auf einem Endlosband, das in quadratische Felder aufgeteilt ist. In diesen Feldern kann jeweils ein Symbol stehen. Zu jedem Zeitpunkt ist genau ein Feld r, das ’abgetastete Feld’ mit dem Symbol $\mathfrak{S}(r)$, in der Maschine. Dieses ’abgetastete Symbol’ ist das einzige, dessen sich die Maschine ’direkt bewusst’ ist. Aber durch den Wechsel ihrer inneren m-Zustände kann die Maschine die vorherigen Symbole effektiv erinnern. Ein effektives Erinnern bedeutet nicht, dass die Maschine Information in einem Speicher abgelegt hat. Die vorhergehenden Symbole haben den Wechsel der m-Zustände beeinflusst und sind daher implizit, im aktuellen m-Zustand erinnert. Der ’Zustand’ der Maschine, der das mögliche weitere Verhalten bestimmt, besteht aus dem m-Zustand q_n und dem ’aktuellen Symbol’ $\mathfrak{S}(r)$. Die Maschine selbst besteht aus einem Schreib/Lese-Kopf, der die folgenden Operationen ausführt: aktuelles Feld lesen, aktuelles Feld über-schreiben, um ein Feld nach links fahren, um ein Feld nach rechts fahren, anhalten (siehe Grafik). Nachdem eine Operation ausgeführt wurde, kann in einen anderen m-Zustand gewechselt werden. Im Fall einer einfachen Turingmaschine – wie im folgenden Bild – bilden alle m-Zustände zusammen das (fix implementierte) Programm.[114]

[113] Turing: *Intelligence Service* S.20
[114] Vgl. Turing: *Intelligence Service* S.20ff



[115]

Die Maschine arbeitet nach einem endlich langen Programm. Die Abarbeitung ist in einzelne elementare Teilschritte (Teiltafel) aufgespalten. Als Beispiel wähle ich eine Turingmaschine, die auf einem binären Alphabet arbeitet. Jeder m-Zustand erhält eine eindeutige Identifikationsziffer (StateId) und eine an den aktuellen Bandwert (Rread) geknüpfte Bedingung. Dann je nach Bandwert eine Schreiboperation (Print; P0, P1), eine Kopfbewegung (Move; L, R) und die Nummer des nächsten auszuführenden m-Zustandes (Next). Dabei kann sich ein m-Zustand durchaus immer wieder selbst aufrufen, bis seine Bedingung erfüllt ist. Die Turingmaschine arbeitet so lange, bis sie die Abbruchbedingung (Stop; S) erreicht. Es gibt allerdings Konfigurationen mit denen sie diese nie erreicht (siehe Halteproblem). Die Inschrift am Band ist dann das Ergebnis der Berechnung. Ein beliebtes Beispielprogramm sieht folgendermaßen aus: Der Kopf geht solange nach rechts, bis er auf einer Bandstelle eine Reihe von '1'en antrifft, und verändert diese, beginnend mit '1', auf abwechselnd '0' und '1'. Bis die ursprüngliche Reihe von '1'en zu Ende ist, also eine '0' auftaucht. Dann geht die Maschine in den Zustand Stop(S).
Hier die Tafel mit den m-Zuständen der Beispielmaschine:

[115]Quelle: ('http://de.wikipedia.org/w/index.php?title=Bild:Turingmaschine.png'); Lizenz: GNU-FDL; Die Okt 21 11:03:41 CEST 2008

StateId	Read	Print	Move	Next
q1	0	P0	R	1
	1	P1	R	2
q2	0	P0	R	4
	1	P0	R	3
q3	0	P0	R	4
	1	P1	R	2
q4	0	P0	S	0
	1	P1	S	0

Angenommen, die Ausgangskonfiguration des Bandes sieht folgendermaßen aus: Es gibt sechs Felder mit der Inschrift ’0’, dann sieben Felder mit der Inschrift ’1’ und ein Feld mit der Inschrift ’0’. Die einzelnen Felder sind zur besseren Adressierbarkeit in der Erklärung mit einem Index (1 – 14) versehen.

0_1	0_2	0_3	0_4	0_5	0_6	1_7	1_8	1_9	1_{10}	1_{11}	1_{12}	1_{13}	0_{14}

Ich starte also mit dem Lesekopf auf dem Feld mit dem Index 1, im Zustand q_1. Der Kopf liest an dieser Stelle ’0’ (Read 0). Also springt die Maschine in die obere Spalte von Zustand q_1 und führt aus, was da steht. Sie schreibt demnach ’0’ (Print P0; in diesem Fall tut sie nichts, da im aktuellen Feld bereits ’0’ steht), fährt mit dem Kopf ein Feld nach Rechts (Move R). Danach bleibt die Maschine im Zustand q_1 (Next 1) und wiederholt den Vorgang so lange (fährt im Prinzip immer weiter nach rechts) bis sie im Feld mit Index 7 auf eine ’1’ stößt. Dann (Read 1) führt sie die untere Spalte von Zustand q_1 aus: Fahre nach rechts und wechsle in Zustand q_2. Im Feld 8 findet sie wiederum eine ’1’ (Read 1) und führt die untere Spalte von Zustand q_2 aus: Schreibe ’0’ auf Feld 8, fahre nach rechts und wechsle in Zustand q_3. Im

Feld 9 findet sie wiederum eine ’1’ (die aber stehen bleiben soll), daher führt sie aus: Schreibe ’1’, fahre nach rechts und wechsle wieder zurück in Zustand q_2. Das geht solange, bis sie in Feld 14 auf eine ’0’ trifft. Sowohl aus Zustand q_2 als auch aus Zustand q_3 würde die Maschine daher in den Zustand q_4 wechseln und hält. Das Endergebnis sieht dann so aus:

0_1	0_2	0_3	0_4	0_5	0_6	1_7	0_8	1_9	0_{10}	1_{11}	0_{12}	1_{13}	0_{14}

Die Auflistung aller Teilschritte mit dem jeweils zugehörigen Bandwert in einer speziellen Notation, auch Turingtafel genannt, definiert eine Tunringmaschine vollständig. “Die Identität zwischen einer Turingtafel und einer Turingmaschine zeigt, daß der tatsächliche Mechanismus der Turingmaschine für das mathematische Modell unerheblich ist. Das einzige, was interessiert, ist die Struktur der Folge von Konfigurationen bzw. Maschinenwörtern, die die Zustände beschreiben, welche die Maschine sukzessive annimmt und die in der Turingtafel aufgeschrieben sind.”[116] Eine Turingmaschine erfüllt damit alle oben angeführten Bedingungen für einen Algorithmus. “Turing stellt nun die These auf, die in Umkehrung dazu sagt, daß jede durch einen Algorithmus definierbare Funktion auch von einer Turingmaschine berechnet werden könne.”[117] Diese These ist ebenso wie jene von Church nicht beweisbar, wird aber dadurch gestützt, dass sich alle anderen Formalisierungen des Algorithmenbegriffes als äquivalent erwiesen haben. Jeder Algorithmus nach obiger Definition lässt sich durch eine geeigneten Turingtabelle beschreiben.

Im Fall des obigen Beispiels ist ein fixes Programm implementiert. Damit ist die Turingmaschine also für eine bestimmte Anwendung maßgeschneidert und wird, egal was auf dem Band steht, immer auf der rechten Seite

[116]Krämer: *Symbolische Maschinen* S.174
[117]Krämer: *Symbolische Maschinen* S.174

nach der ersten ’1’ suchen. Wenn auf der rechten Seite keine ’1’ am Band steht, wird sie ewig nach rechts weiterfahren. Sie wird also nie halten.
Um die Notation zu verkürzen, kann man mehrere Kopfbewegungen in einen m-Zustand verpacken. Damit lässt sich die Anzahl der m-Zustände meist beachtlich reduzieren. Um die Tabellen für komplexere Aufgaben weiter zu vereinfachen, werden oft verwendete Operationen (kopieren oder vergleichen von Symbolfolgen; tilgen aller Symbole einer Form, etc.) zusammengefasst, um als Funktionen in das Tabellen-Gerüst einzugehen.[118]

2.2.5 Universale Turingmaschinen

Eine weitere Sichtweise auf den Berechenbarkeitsbegriff liefert Alan Turings Aufsatz ’On Computable Numbers, with an application to the Entscheidungsproblem’. Er beschreibt darin die berechenbaren Zahlen als “diejenigen reellen Zahlen, deren Dezimalausdrücke mit endlichen Mitteln errechnet werden können.”[119] Wenn der Dezimalausdruck einer Zahl, in endlicher Zeit von einer Maschine niedergeschrieben werden kann, ist sie berechenbar. Es gibt relativ wenige definitiv nicht-berechenbare Zahlen, die auch Verwendung finden. Aber obwohl die Klasse der berechenbaren Zahlen sehr vielschichtig ist, ist sie im Vergleich zur überabzählbaren Klasse der reellen Zahlen doch relativ klein. Und da die Anzahl der möglichen sie berechnenden Turingmaschinen abzählbar ist, sind die berechenbaren Zahlen ebenfalls in die natürlichen Zahlen abbildbar, und daher auch abzählbar (vgl. Gödelzahlen).
Wie lässt sich die Abzählbarkeit der Turingmaschinen zeigen? Folgender-

[118]Vgl. Turing: *Intelligence Service* S.24ff
[119]Turing: *Intelligence Service* S.19

maßen wird vorgegangen: “Eine berechenbare Folge γ wird durch die Beschreibung einer Maschine zur Berechnung von γ bestimmt [...] und es ist tatsächlich der Fall, daß jede berechenbare Folge in Terms [sic] einer solchen Tabelle beschrieben werden kann.”[120] Turing schafft eine neue Notation, indem er alle nacheinander folgenden Zeileneinträge der Maschinentabelle (s.o.), durch Semikolons getrennt, in eine Zeile schreibt. Die Notation wird weiter vereinfacht, indem zum Beispiel der oben erwähnte Marker für den m-Zustand q_4 durch den Buchstaben ’D’ für Zustandsmarker, gefolgt von vier Mal dem Buchstaben ’A’, repräsentiert wird. Ebenso wird mit den anderen Symbolen und Operationen verfahren. Diese Art der Beschreibung einer Turingmaschine nennt Alan Turing: ’standard description’. Sie besteht nur aus den Buchstaben ’A, C, D, I, R, N’. “Wenn wir schließlich noch ’A’ durch ’1’, ’C’ durch ’2’, ’D’ durch ’3’, ’L’ durch ’4’, ’R’ durch ’5’, ’N’ durch ’6’ und ’;’ durch ’7’ ersetzen, werden wir schließlich zu einer Beschreibung der Maschine kommen, die die Form einer arabischen Ziffer hat.”[121] Diese Zahl heißt dann Beschreibungszahl (’description number’). Die Maschine mit der Beschreibungszahl n kann damit als $\mathfrak{M}(n)$ bezeichnet werden.

> Jeder berechenbaren Folge entspricht mindestens eine Beschreibungszahl. Während keiner Beschreibungszahl mehr als eine berechenbare Folge entspricht. Die berechenbaren Folgen und Zahlen sind daher abzählbar.[122]

Damit ist es möglich, eine Maschine $\mathfrak{U}$ zu konstruieren, sodass, wenn auf deren Band als Inschrift die Beschreibungszahl (’description number’, kurz

[120] Turing: *Intelligence Service* S.29
[121] Turing: *Intelligence Service* S.30
[122] Turing: *Intelligence Service* S.30

D.N.) irgendeiner anderen Turingmaschine $\mathfrak{M}$ steht, die Maschine $\mathfrak{U}$ dieselbe Folge wie $\mathfrak{M}$ berechnet. $\mathfrak{U}$ ist eine universelle Turingmaschine wenn sie jede beliebige Beschreibungszahl ausführen kann. Die vollständige Tabelle will ich hier auslassen[123].

> Die Bedeutung der universalen Maschine ist klar. Wir brauchen nicht unzählige unterschiedliche Maschinen für unterschiedliche Aufgaben. Eine einzige wird genügen. Das technische Problem der Herstellung verschiedener Maschinen für verschiedene Zwecke ist ersetzt durch die Schreibarbeit, die Universalmaschine für diese Aufgaben zu programmieren.[124]

Universale Turingmaschinen (UTM) können auf verschiedene Weise vorkonfiguriert sein. Die aus den m-Zuständen bestehende Konfiguration definiert den Code, mit dem die Maschine gefüttert werden kann. Die UTM verfügt über keinen weiteren Speicher. Analog der 'Von Neumann-Architektur', der die meisten aktuellen Computersysteme entsprechen, befinden sich sowohl das auszuführende Programm als auch die zu bearbeitenden Daten im selben Speicher. Im Falle der UTM liegen sie auf dem Endlosband. Somit kann sie mit einem speziell präpariertem Band (das ein Programm, eben die D.N. einer speziellen Turingmaschine enthält) jede beliebige (mögliche) Turingmaschine simulieren.[125]

[123] Eine solche finden sie zum Beispiel: Turing: *Intelligence Service* S.33ff
[124] Turing: *Intelligence Service* S.88
[125] Ein illustriertes Beispiel sich unter: (http://www.frank-busse.de/texte/utms.php'), Mit Okt 22 12:06:09 CEST 2008

2.3 Analoge und digitale Rechenmaschinen

2.3.1 Analoge Computer

Das, was wir Analogcomputer nennen, sind physikalische Phänomene, die einem 'idealen' funktionalen Zusammenhang sehr nahe kommen. "In einem Analogrechner wird jede Zahl durch eine physikalische Größe dargestellt, deren Wert, gemessen in einer vorbestimmten Einheit, gleich der betreffenden Zahl ist. Die zur Darstellung verwendete physikalische Größe kann ein Rotationswinkel einer Scheibe, ein elektrischer Strom oder eine (gegen einen Bezugspunkt gemessene) elektrische Spannung usw. sein. Damit die Maschine rechnen, d.h. mit diesen Zahlen nach einem vorher festgelegten Plan operieren kann, ist es notwendig, Bausteine (oder Bauelemente) zu finden, die mit diesen dargestellten Größen die mathematischen Grundoperationen ausführen können."[126]

Analogrechner sind kontinuierliche, physische Prozesse, die die Quantität eines Eingangswertes instantan in eine Quantität als Ausgangswertes konvertieren. "Digital computation abstracts away from the concrete properties of a physical substrate, whereas these properties are often crucial for the specific computational purpose of an analog device."[127] In einem Analogrechner gibt es keinen Unterschied zwischen Software und physischer Hardware. Während Digitalrechner in der Regel seriell arbeiten, wird die Funktionsweise von Analogrechnern als parallel dargestellt, da die verschiedenen Komponenten simultan arbeiten. Analogrechner werden bedient, indem die Werte als physikalische Quantitäten angelegt und

[126]Neumann, John von (Hrsg.): *Die Rechenmaschine und das Gehirn*. München, Wien: R.Oldenburg Verlag, 1980 S.16

[127]Fitz, Hartmut: Church's Thesis And Physical Computation. In Adam Olszwski, Jan Woleǹski, Robert Janusz (Hrsg.): Church's Thesis After 70 Years. Band 1, Heusenstamm: Ontos Verlag, 2006 S.194

wieder gemessen werden (Ladung, Spannung, etc.). Aufgrund der Annahme, dass die physische Welt Kontinua, die in ihrem Verlauf einer irrationalen Zahl entsprechen, zur Verfügung stellt, sind Funktionen von rellen Zahlen das fundamentale Konzept von Analogrechnern. “While digital computers represent binary quantities by discrete states such as open relays, analog computers allow representing reals by physical quantities. It has been conjectured that this enhanced representational capacity also results in amplified computational power.”[128] Bei vielen Anwendungen bieten Analogrechner einen essentiellen Geschwindigkeitsvorteil. Ein Analogrechner kann durchaus sehr komplexe Formen annehmen und aus speziellen Logik-Bausteinen zusammengesetzt sein, die auf der Basis von reellen Werten operieren. Diese können zum Beispiel als Multiplizier-Einheit, Addier-Einheit oder Integrator-Einheit usw. verwirklicht sein, sind aber an die Verwirklichung als physische Entitäten gebunden (vgl.GPAC[129], siehe auch Jerzy Mycka[130]) und somit mit Unschärfen versehen. Sowohl bei der Messung (die zur Evaluierung eines Ergebnisses notwendig ist) oder aber durch Unregelmäßigkeiten in den physischen Bauelementen, kann der Analogrechner nur mit mehr oder weniger – in einem Toleranzbereich – reellen Werten rechnen. Im Gegensatz zu den diskreten Werten einer digitalen Rechenmaschine, die in ihrer physischen Realisierung als Computer exakt dieselben, eben diskreten, Werte darstellt, haben wir beim Analogrechner die Vorstellung einer ’Reihe von idealen reellen Werten’(zum

[128] Fitz S.194

[129] General Pourpose Analog Computer. Wurde von Shannon als mathematisches Modell ’Differential Analyzer’ eingeführt. Dann von Vannevar Bush am MIT entwickelt und schließlich 1931 gebaut. Die fundamentalen Prinzipien wurden erstmals von Lord Kelvin 1876 beschrieben.

[130] Mycka, Jerzy: Analog Computation and Church’s Thesis. In Adam Olszwski, Jan Woleǹski, Robert Janusz (Hrsg.): Church’s Thesis After 70 Years. Band 1, Heusenstamm: Ontos Verlag, 2006 S.332)

Beispiel ein Polynom von reellen Werten), die in der physischen Realisierung an natürliche Schwankungen und Wechselwirkungen gebunden – und daher unvollkommen – ist. Damit ist ein Analogrechner ein physisches Phänomen, das einen funktionalen Zusammenhang (in manchen Fällen auch eine ganze Reihe solcher) besonders genau darstellt, und darum, umgekehrt sehr effektiv zur Bestimmung von Werten verwendet werden kann. Korrekt wäre von einer analogen Rechenhilfe zu sprechen, die jeweilige Funktion wird aber nicht im Verlauf berechnet, sondern ist in diesen Fällen immer vorweggenommen.

2.3.2 Hypercomputation

Jerzy Mycka analysiert in 'Analog Computation and Church's Thesis' Fälle, in denen Computer über die Limitierung von Turingmaschinen hinaus Berechnungen anstellen könnten. Man spricht dann von 'Hypercomputation'. Und zwar wäre das in solchen Fällen möglich, wo eine Turingmaschine durch die Limitierung von Zeit und Speicher gewöhnlich an ihre Grenzen stößt. Ein populäres Beispiel dafür ist das Halteproblem[131]. Ein Gedankenspiel, das dessen Lösung beansprucht, propagiert beschleunigte Turingmaschinen ('accelerated Turingmachines', vgl. Zeno's Paradox). Eine solche Turingmaschine hat auf ihrem Band ein bestimmtes Feld 'x', in dem zu Anfang eine '0' steht, und in welches sie, falls sie terminiert, eine '1' schreibt. Sie verrichtet den ersten Berechnungsschritt in einer Zeiteinheit, den zweiten Schritt in der Hälfte der Zeit, den dritten Schritt in einem Viertel der Zeit, und so weiter. Somit ist spätestens nach zwei Zeiteinheiten klar: Wenn im Feld 'x' eine '1' steht, ist der Algorithmus terminiert, wenn

[131] Eine Folgerung aus Gödels Unvollständigkeitssatz; Alan Turing hat bewiesen, dass es keinen von einer Turingmaschine berechenbaren Algorithmus gibt, der für jede beliebige Turingmaschine vorausberechnen kann, ob sie anhält.

dort eine ’0’ steht, terminiert der Algorithmus nicht. Da im Gegensatz zu diesem diskreten Prozess ein Analog-Computer jedoch in endlicher Zeit ’quasi’ ein ganzes Kontinuum von Werten abarbeiten kann, gibt es Vorschläge, ihn als Schlüssel zur ’Hypercomputation’ zu sehen. Das ist aber nur möglich, wenn der reale physische Prozess ein ideales Kontinuum[132] implementiert.

Die Idee der Hypercomputation ist jedoch nicht neu, sondern so alt wie die Idee der Turingmaschine selber. Alan Turing führt 1939 in ’Systems of logic based on ordinals’[133] als Gedankenexperiment eine Maschine mit den Eigenschaften eines Orakels ein. Man kann sich das Orakel als Turingmaschine vorstellen, die ein Problem (das Problem, dessen Orakel es ist) in einem Schritt löst (vgl. Blackbox). Solche Orakelmaschinen werden eingesetzt, um zum Beispiel nicht-deterministische Systeme deterministisch zu modellieren. So ist es auch möglich, ein spezielles Orakel für das Halteproblem einzuführen. Wie Alan Turing zeigen kann, trifft das Halteproblem auch auf Halteorakelmaschinen zu. Obwohl das Orakel für eine beliebige ’normale’ Turingmaschine entscheiden kann, ob sie hält, kann sie nicht entscheiden, ob eine beliebige Turingmaschine, die mit einem äquivalenten Orakel ausgestattet ist, hält. Sie bräuchte dazu wieder ein eigenes Orakel. So entsteht eine Hierarchie von Orakeln, äquvalent zu einer ’arithmetischen Hierarchie’.

[132] Weiter unten wird noch erwähnt, dass vollständige Linearität eines physischen Phänomens für Hyperturing-Berechnungen ausreichend ist; es muss nicht absolut Ideal sein, ideal linear reicht schon aus. Es ist jedoch nicht so, dass ich glaube, dass dies leichter in einem physischen Phänomen zu finden wäre.

[133] Turing, Alan M.: Systems of logic based on ordinals. Proceedings of the London Mathematical Society. Second Series, 45 1939

2.3.3 Status der Church-Turing-These

Die Church-Turing-These stellt eine Kombination der Thesen von Alonzo Church und Alan Turing dar, wurde jedoch niemals in der Form von einem der beiden Namensgeber formuliert; sie ist ein Schulkonstrukt[134]. Ich möchte, bevor ich mich theoretischen Versuchen zu physischen Fundierung der Church-Turing-These widme, noch einmal den Status dieser – nach Sibylle Krämer – 'fundierten Meinung' überprüfen. Eine deutsche Fassung würde in etwa lauten:

> Die Klasse der von einer Turingmaschine berechenbaren Funktionen ist genau die Klasse der intuitiv berechenbaren Funktionen.

Die englische Formulierung bei Hartmut Fitz lautet folgendermaßen:

> "A function is effectively computable iff [d.h. 'if and only if'; Anm. des Autors] it is Turing machine computable" (CTT).

In der englischsprachigen Literatur wird 'intuitiv berechenbar' meist mit 'effectively computable' übersetzt. Um die vage Definition dieses Begriffes rankt sich eine breite Diskussion. Die Church-Turing-These ist, ebenso wie ihre Mutterthesen, nicht beweisbar. Ich will es noch einmal in anderen Worten wiedergeben: Wenn eine bestimmte Funktion als Algorithmus niedergeschrieben, respektive formalisiert und von einer Turingmaschine berechnet werden kann, so ist sie 'berechenbar'. Um herauszufinden, ob eine Funktion intuitiv berechenbar ('effectively computable') ist oder nicht, müssen wir ausprobieren. Es gibt keine mathematische Beschreibung für die Klasse der 'intuitiv berechenbaren Funktionen'. Es fand sich aber bisher auch kein Gegenbeispiel im Sinne einer Funktion, die sich zwar als

[134] Fitz vgl. S.176

'berechenbar' aber nicht von einer Turingmaschine berechenbar erwiesen hätte.
Alle bisherigen Vorschläge für einen Berechenbarkeitsbegriff (λ-Kalkül, μ-rekursive Funktionen, Turingmaschine, etc.) haben sich als äquivalent und haltbar erwiesen, allerdings stellen sie kein hartes Kriterium dar. Sie schließen jedoch alle bisher bekannten 'berechenbaren Funktionen' mit ein.
Falls die These allerdings wahr ist – und es wurde bisher noch kein Gegenbeispiel gefunden – folgt: Es kann keine andere Art von Rechner geben, die mehr als eine Turingmaschine, und damit mehr als bisherige Computer mit höheren Programmiersprachen, berechnen kann. Jeder gängige Desktoprechner zusammen mit einer höheren Programmiersprache ist im Funktionsumfang einer universellen Turingmaschine äquivalent und kann jeden möglichen Algorithmus berechnen. Vorausgesetzt, es gibt genug Speicherplatz und wir haben genügend Zeit, das Ergebnis abzuwarten. Aufgrund des Halteproblems (siehe oben: Entscheidungsproblem) ist es nicht möglich, eine Maschine so zu programmieren, dass sie für jeden Algorithmus vorhersagen kann, ob er jemals anhält.
Sollte sich die Church-Turing-These als falsch erweisen, respektive ein Gegenbeispiel gefunden werden, so ist es möglich, einen Rechner zu bauen, der fähig ist, Berechnungen über die Limitierungen der Turingmaschine hinaus anzustellen. Man spricht dann von 'Hypercomputation'. Eine solche Berechnung bestünde zum Beispiel in der Lösung des Halteproblems.
Die Definition des Algorithmenbegriffs stellt kein Berechenbarkeitskriterium, sondern eine Vorschrift zur Erstellung von maschinenverarbeitbaren Regeln dar. Andreas Blass und Yuri Gurevich schreiben in 'Algorithms: A

Quest for Absolute Definitions'[135]: "[...] it is often assumed that the Church-Turing thesis settled the problem of what an algorithm is. That isn't so. The thesis claryfies the notion of computable function. And there is more, much more to an algorithm than the function it computes. The thesis was a great step towards understanding algorithms, but did not solve the problem what an algorithm is."[136]

Um noch mals eine, von Sibylle Krämers Darstellung leicht abweichende Definition für Algorithmen zu geben, möchte ich hier noch Donald E. Knuth's Vorbemerkungen aus 'The Art of Computerprogramming' heranziehen: "The NOTION of an algorithm is basic to all of computer programming, so we should begin with a careful analysis of this concept."[137] Neben einer Untersuchung über die geschichtliche Entwicklung des Begriffs, gibt er folgende Anweisungen: Zusätzlich zu der Anforderung, dass es sich um einen endlichen Satz von Regeln handelt, der in einer Reihe von Operationen ein spezifisches Problem löst, hat ein Algorithmus folgende fünf Eigenschaften:

- Endlichkeit ('finiteness'): Ein Algorithmus muss immer nach einer endlichen Anzahl von Schritten terminieren. Eine Prozedur, die eventuell nicht terminiert, nennt er Berechnungsmethode ('computational method'); ein Beispiel wäre ein Computerprogramm, das in einer Endlosschleife mit der Umgebung interagiert ('reactive process').

- Bestimmtheit ('definiteness'): Jeder Berechungsschritt eines Algorithmus' muss präzise und unzweideutig definiert sein.

[135] Blass, Andreas; Gurevich, Yuri: Algorithms: A Quest For Absolute Definitions. In Adam Olszwski, Jan Woleǹski, Robert Janusz (Hrsg.): Church's Thesis After 70 Years. Band 1, Heusenstamm: Ontos Verlag, 2006 S.24

[136] Blass S.24f

[137] Knuth, Donald Ervin: *Fundamental Algorithms*. Band 1, The Art of Computer Programming. Boston: Addison-Wesley, 1997 S.1

- Eingangswerte ('input'): Ein Algorithmus hat null oder mehr Eingangswerte, die ihm vor dem Start gegeben sind, oder dynamisch ausgelesen werden während der Algorithmus läuft.

- Ausgangswerte ('output'): Ein Algorithmus hat einen oder mehrere Ausgangswerte, die in spezieller Relation zu den Eingangswerten stehen.

- Effektivität ('effectiveness'): Von einem Algorithmus wird generell erwartet, dass er seine Aufgabe 'effektiv' durchführt, und die Operationen im Prinzip auch in einer finiten Zeitspanne mittels Papier und Bleistift durchführbar wären. Eine Berechnung mit reellen Werten ist demnach ausgeschlossen.[138]

Diese Angaben entsprechen einer guten Richtlinie für praktische Programmiertätigkeit, erfüllen aber nicht die Kriterien einer mathematischen Definition. Knuth gibt deshalb zusätzlich eine formale Definition einer Berechnungsmethode ('computational method') an, die obige Eigenschaften beinhaltet[139]. "There are many other essentially equivalent ways to formulate the concept of an effective computational method (for example, using Turing machines)."[140] Eine ähnliche Formulierung gibt A.A.Markov in seinem Buch 'The Theory of Algorithms' und allen gemeinsam ist, dass sie dem oben eingeführten Berechenbarkeitsbegriff äquivalent sind.
Aber das umfasst nicht alles, was unsere Computer tun. Deshalb geben manche Autoren weiter gefasste Definitionen für Algorithmen. Explizit ausgenommen wird zum Beispiel das Kriterium der Endlichkeit ('finiteness'), da eine potentiell endlose Programmschleife ('main loop') grundlegend für

[138] Vgl. Knuth S.4ff
[139] Vgl. Knuth S.7f
[140] Knuth S.8

moderne, interaktive Betriebssysteme ist. Unser Verständnis von Algorithmen hat sich bis heute auch in anderer Hinsicht weiterentwickelt. Blass und Gurevich nennen folgende Bereiche: 'Interactive algorithms', wie zum Beispiel die Berechnung von Zufallswerten, die auf 'Seeds' (Mausbewegung oder andere Variable aus dem Environment) angewiesen sind; asynchrone Algorithmen, deren Timing von der Umgebung beeinflusst wird; interaktive Algorithmen können zum Beispiel auch nur partiell rekursive Funktionen berechnen; und es kommen noch Algorithmen, die generell auf Benutzereingaben warten, hinzu. Auch 'Computing with abstract structures' sowie Parallelalgorithmen, verteilte Systeme und 'non-discrete Computations' sind nicht direkt von Turings Analysen berücksichtigt worden[141]. Außerdem gibt es eine Weiterentwicklung der Turingmaschine als 'pointer-machine', wie die Kolmogorov-Uspensky-Maschine ("contrary to Turings tape whose topology is fixed, Kolmogorov's 'tape' is reconfigurable."), oder Knuth-Schönhage-Maschinen, die dynamisch auf ihren Speicher zugreifen. "Dima Grigoriev proved that Turing machines cannot simulate Kolmogorov machines in real time"[142]

Wie immer man dazu stehen mag, die Definitionen für Algorithmen bleiben in gewissen Grenzen wandelbar.

2.3.4 Die Physikalische Church-Turing-These

Jerzy Mycka, der einen Weg finden will um die 'fundierte Meinung' der Berechenbarkeit zuzuführen, schreibt, dass Alan Turing nicht in erster Linie das Potential von Maschinen beschreiben will, sondern mit dem Wort 'computor' die idealisierte menschliche Tätigkeit herausstreicht: "the con-

[141] Vgl. Blass S.31ff
[142] Blass S.36

struction of Turing Machines encloses rather the possibilities of 'an ideal mathematician' activity."[143] In der Hoffnung, die Formulierung 'intuitiv berechenbar' ('effectively computable') in der Church-Tunring-These, welche – wie oben gezeigt – mehr den Charakter einer fundierten Meinung hat, mathematisch handhabbar zu machen, möchte Mycka einen Schritt weiter gehen: "Namely, physical devices are closer to our idea of computation than any kind of human activity. With respect to this observation some modification of Church's thesis [...] can be proposed."[144] Daher formuliert er eine physikalische Version der Church-Turing-These (PCT): "Everything that can be computed by any physical process can be calculated by a Turing machine as well."[145] Welche Schlüsse lassen sich aus einer solchen These ziehen? Falls sich die Aussage beweisen lässt, besagt sie, dass alles, was überhaupt berechnet werden kann, turingmaschinen-berechenbar ist. Damit wäre 'Hypercomputation' ausgeschlossen. Aber was ließe sich sonst daraus folgern? Stellen somit alle physischen Prozesse Berechnungen ('computations') an? Ist es möglich, dass eine Turingmaschine alle physischen Prozesse unserer Welt simuliert? Wer entscheidet, ob ein physischer Prozess gerade eine Berechnung durchführt, oder nicht? Berechnet alles vor sich hin? Oder benötigen wir einen Homunkulus, der diese Aufgabe übernimmt? Das beschreibt im Wesentlichen den Status, den wir schon bei Searles und Shagrirs Argumenten feststellen konnten. Falls sich ein Gegenbeweis findet, und die These widerlegt werden kann – also etwa ein Beispiel von Hypercompuation gefunden wird – ist die Folge ungewiss: Es gibt 'ideale' Prozesse, die zwar irgend anderweitig, aber nicht mit einer Turingmaschine berechnet werden können? Diese Aussage entspricht je-

[143] Mycka S.348
[144] Mycka S.348
[145] Mycka S.348

doch wiederum der Definition von Hypercomputation.
Durch die Formulierung einer PCT lässt sich folgender Vorteil erhoffen: Wenn sich physische Prozesse – etwa in Form von Naturgesetzen – mathematisch erfassen lassen, so kann der schwammige Teil ('intuitiv berechenbar') der Church-Turing-These durch eine in den Naturgesetzen fundierte Formel ersetzt werden. Damit wäre ein Beweis theoretisch denkbar und der Gültigkeitsbereich der Church-Turing-These endlich festgelegt. Das soll im Folgenden genauer untersucht werden.
Hartmut Fitz präsentiert in seinem Artikel 'Church's Thesis and physical Computation' typische Ansichen über berechnende physische Objekte. Etwa schreibt David Deutsch 1997:

> Computers are physical objects, and computations are physical processes. What computers can or cannot compute is determined by the laws of physics alone and not by pure mathematics. (Deutsch 1997, p98)[146]

Oder wie 1985 Charles H. Bennett an Rolf Landauer schreibt:

> A computation, whether it is performed by electronic machinery, on an abacus or in a biological system such as the brain, is a physical process. [...] We are looking for general laws that must govern all information processing, no matter how it is accomplished. Any limits we find must be based solely on fundamental physical principles [...](Bennett an Landauer 1985, p48)[147]

Laut Fitz artikulieren Deutsch, Bennett, Landauer und auch andere ("See also Lloyd[2000], Estesi and Nèmeti[2002], Hoarth[2004] – and many more"), eine weit verbreitete Ansicht. Philosophische Überlegungen werden

[146]Zitiert von Fitz S.178
[147]Zitiert von Fitz S.178

in diesen Belangen als irrelevant angesehen, denn, wie die fundamentalen Gesetze des Elektromagnetismus oder der Thermodynamik, sind die Eigenschaften eines Computers – als physisches System – Gegenstand einer wissenschaftlichen Tatsachenfeststellung. Im Gegensatz dazu waren Rechenmaschinen zur Zeit von Turing und Gödel noch eine rein mathematisch-logische Aufgabe, was aber – nach Ansicht der Genannten – nicht zuletzt daran lag, dass im Gegensatz zu heute kaum physische Computer zur Verfügung standen[148]. Im Folgenden will ich fünf wichtige Punkte aus der diesbezüglichen Darstellung von Hartmut Fitz herausgreifen.

Spezifizierung der Implementierung Wie können wir spezifizieren, was es heißt, dass ein physisches System eine Berechnung ('computation') durchführt? Beziehungsweise wie können wir feststellen, ob ein physischer Zusammenhang eine Berechnung implementiert? Eine häufige Feststellung lautet: "A physical system implements a given computation when the causal structure of the physical system mirrors the formal structure of the computation (Chalmers 1994, p.392)"[149] Um zu einem Vergleich herangezogen werden zu können, muss die formale Struktur einer Berechnung ('formal structure of computation') aber erst spezifiziert werden. Dazu ist ein Beobachter ('observer') notwendig. Wie wir schon bei Searle gehört haben, erklärt die Beschreibung der Struktur jedoch nicht das kausale Verhalten eines Systems. Für eine kausale Erklärung wird in den Naturwissenschaften die Angabe der Gesetze des Ablaufes, zusammen mit der Spezifizierung dessen, was den Prozess initiiert (bzw. auslöst), gefordert. Dadurch ist aber noch nicht viel gewonnen, da die Zuschreibung

[148]Vgl. Fitz S.178
[149]zitiert nach Fitz S.181

der Zustände des Systems (initialisiert, ... terminiert) wieder von außen erfolgen muss. “But because we are bio-mechanical and thus physical systems, an explication of physical computation based on human intervention eats its tail.”[150]

Empirische Unzugänglichkeit Es mag sein, dass die PCT falsch oder richtig ist, oder auch, dass sie nur zum jetzigen Zeitpunkt noch nicht als falsch angesehen wird. Ist sie jedoch überhaupt nicht falsifizierbar, ist ihre Aussagekraft mehr als zweifelhaft. Wir könnten sie empirisch falsifizieren, indem wir in der Natur einen Berechnungsvorgang (’computational process’) entdecken, den eine Turingmaschine nicht simulieren kann. Dafür können für uns durch Messung zwar zugängliche aber ansonsten zufällige oder erratische Prozesse jedoch nicht herangezogen werden (s.u. Trivialisierung). Also versucht man es mit einer ’Blackbox’, die eine nicht berechenbare Funktion (’noncomputable function’) berechnet (vgl. Turings Orakel), für wiederholte Eingaben aber immer gleiche Ergebnisse produziert. Die Berechnung nimmt dafür immer die gleiche Zeit in Anspruch. Allein dadurch bin ich aber noch nicht überzeugt, dass unsere Box auch wirklich eine nicht-rekursive Funktion berechnet, und es entsteht zusätzlich das Problem, dass ein endlicher Datensatz diese Eigenschaft nicht bestätigen kann: “All observational data is finite but non-recursiveness is an inherently infinitistic property.”[151] Somit scheint es keinen Weg zu geben, einen nicht-rekursiven Prozess in der Natur durch einfache Beobachtung zu entdecken. Deshalb gibt es auch keine Möglichkeit, die PCT empirisch zu falsifizieren. Hartmut Fitz hegt noch die Vermutung, dass es

[150] Fitz S.185
[151] Fitz S.188

eventuell in Verbindung mit logisch-mathematischen Einsichten möglich ist: Angenommen, ich habe eine nicht-rekursive Theorie, die ein System mit einem messbaren Ausgang adäquat beschreibt und das Halteproblem berechnet. So stelle ich – zum Beispiel in einem Experiment – mit nur endlich vielen Datensätzen fest, dass sich die Annahme in diesen Fällen bestätigt. Aber ich würde damit nicht beweisen können, ob sie tatsächlich für alle möglichen Eingaben das Halteproblem berechnet. “As a mathematical object (or a physical instantation thereof, for that matter) a Turing machine is finite but it is an 'idealization which brings in infinity [and zero].”[152]

Erkenntnisproblem: Feststellung von Nicht-Computation Wissenschaftliche Erkenntnis im Sinne der 'scientific method' kann als selbst-korrigierende Prozedur, die sukzessive, verlässlich und effektiv nach Wissen strebt, beschrieben werden. Aufgrund der induktiven Unterdetermination wissenschaftlicher Untersuchungen gibt es jedoch keine universalen Gesetze und kein unumstößliches Wissen für alle Zeiten. Den Hypothesen werden Widerlegungen und Nachbesserungen zugestanden. Zum Beispiel darf eine verifizierbare universale Hypothese als 'wahr' angenommen werden, solange keine belegbaren Widersprüche auftreten. Ebenso sind Annahmen falsifizierbar, beziehungsweise dürfen als 'falsch' angenommen werden, solange es nicht möglich ist, ein positives Beispiel zu produzieren. In beiden Fällen lässt sich mit dieser Methode in absehbarer Zeit Wissen gewinnen: “A statement which is verifiable and falsifiable in this sense is decideable in the limit.”[153] Die PCT ist in erkenntnistheore-

[152]Fitz S.189
[153]Fitz S.191

tischer Hinsicht nicht äquivalent zu den eben aufgeführten wissenschaftlichen Theorien. Wie wir gesehen haben, ist sie mit einer endlichen Anzahl von Datensätzen nicht verifizierbar. Sie zu falsifizieren ist gleichermaßen schwierig. Ob der Prozess X eine 'Hypercomputation' darstellt, kann, ebenso wie die PCT selbst, nicht mit endlichen Mitteln entschieden werden. Umgekehrt ist Nicht-Berechenbarkeit ('non-computability') keine mathematische Entität, sondern nur eine theoretische Entität – wie Elektronen in der Physik – die durch Beobachtung festgelegt wird. Eventuell stellt sich ein Sachverhalt später noch als berechenbar heraus. Aber welche Art von Theorie ist notwendig, damit 'Nicht-Berechenbarkeit' feststellt werden kann? "Accordingly, it is perfectly conceiveable that non-computability in nature, the falsehood of PCT, is a theoretical consequence of an observationally accessible theory."[154]. Und eine solche Theorie müsste von einer mathematischen Beschreibung des Sachverhalts verschieden, das heißt, eine wesentlich nicht-mathematische Theorie sein, die trotzdem mathematisch überprüfbar bleibt. Sowohl 'Hypercomputation' als auch die PCT sind isolierte Statements, haben keine Verbindung zu überprüfbaren Annahmen und erfüllen im Grunde keine Aufgabe in dem Sinn, irgendwelche beobachtbaren Phänomene zu erklären.

Analogrechner: Absolute Präzision

> As in the case of accumulator machines, asynchronous networks and ARNNs [analog recurrent connectionist networks], which serve only as examples here, their computational capacities derive from the representational power of the continu-

[154] Fitz S.193

> um, and are not due to the procedural capacities of the proposed mechanisms. The alleged 'super-Turingness' originates from some non-computable real which figures in the computational process itself, as input, clock frequency, or design constant. Thus, in violating PCT these models essentially depend on the existence of exact real-valued physical quantities.[155]

Es darf bezweifelt werden, ob ein physikalisches Gerät mit absoluter Präzision ('infinite precision') arbeiten kann. Die exakten reellen Quantitäten sind von einem zugrundeliegenden physischen Phänomen abhängig (siehe oben: thermisches Rauschen, Gitterschwingungen, etc.), der ideelle Wert im mathematischen Modell entspringt jedoch einer Zuschreibung: "States of the system are represented as real vectors only in the corresponding mathematical model but these representations are purely ascriptional". Obwohl es berechtigte Zweifel gibt, ob absolute Präzision hierfür notwendig ist – "Siglemann and Sonntag [1994] have shown that unbound linear precision suffices for their ARNNs tho become Super-Turing"[156] – ist unklar, wie absolute lineare Präzision verwirklicht werden soll? Weiters sind wir mit den fundamentalen Einschränkungen der jeweiligen Messung konfrontiert. "Measured quantities are always rational, and in a bounded range, measurement can only distinguish finitely many states of a physical system."[157] Somit wäre ein solches System eventuell durch eine Turingmaschine, die eine hinreichende Näherung berechnet, simulierbar (siehe unten: Turing-Approximation-Hypothesis (TAH)).
Fitz ortet tiefgreifende, erkenntnistheoretische und ontologische Probleme:

[155] Fitz S.197
[156] Fitz S.197
[157] Fitz S.198

> According to the doctrine of scientific realism, objects exist externally to us and independently of the mental. [...] We believe that objects exist if our best scientific theories claim or entail that they exist. On the other hand, no plausible variant of realist doctrines demands that we should accept our best scientific theories at face value. Rather they are transitory stages in scientific development which increasingly approximate truth. Likewise, mathematical descriptions which are an integral part of our best scientific theories, are taken to be true only approximately on most reasonable accounts of theory realism. Thus it is doubtful whether there is any meaning to a scientific statement asserting the existence of analog hypercomputation relying on infinite precision. Optimists about such observationally inaccessible devices have an obligation to clarify the semantic relation between approximate scientific theories and a world of real-valued quantities before they are justified in asserting that their existence entirely depends on our best scientific theories.[158]

Das mathematische Kontinuum ist seit Leibniz (oder eigentlich seit Aristoteles) integraler Bestandteil unserer wissenschaftlichen Methoden, mit deren Hilfe wir natürliche Phänomene im Rahmen von Zeit und Raum erklären. Das Kontinuum ist jedoch nur eine hilfreiche theoretische Konstruktion, kein ontologisches Faktum. Die Einteilung der Erscheinungen in kontinuierliche und diskrete ist keine empirische Frage, da sowohl Dichte als auch Unzählbarkeit die menschlichen Erkenntnisfähigkeiten übersteigen. “How we model time and space is therefore a largely pragmatic decisi-

[158] Fitz S.198f

on; there is no experimentally accessible fact of the matter. The idea of hypercomputational infinite precision devices seems to vanish behind this pragmatic veil."[159] Es hat den Anschein, als ob Analogrechner nicht die Beschreibung physischer Berechnungen ('physical computations') vereinfachen, sondern im Gegenteil in inflationäre Erklärungen münden.

Mit Bezug auf Dreyfuss' Analyse 'What Computers Still Can't Do: A Critique of Artificial Reason' stellt Fitz eine 'Turing approximation Hypothesis' (TAH) auf: "The input-output behavior of any analog computer can be approximated by a Turing machine to any desired degree of accuracy."[160] Nach unserem Wissenstand hat eine Messung immer nur einen bestimmten Genauigkeitsgrad. Somit müsste lediglich eine Turingmaschine existieren, die den Analogrechner bis zu dieser Genauigkeit hin simulieren kann. Darüber hinaus könnten wir, da keine Messdaten vorliegen, nichts aussagen. So wäre die Turingmaschine eine rekursive Beschreibung des Analogrechners. Sie könnte aber keinesfalls jeden beliebigen Grad an Genauigkeit einer nicht-rekursiven Funktion berechnen und ist somit darauf angewiesen, die Messdaten anzunähern. Es ist nicht entscheidbar, ob der Analogrechner tatsächlich eine nicht-rekursive Funktion berechnet, oder ob wir die rekursive Funktion, die diese Werte beschreibt, noch nicht gefunden haben.

Trivialisierung: Alles ist ein Computer Es gibt eine weitere Interpretation der PCT: Alles kann als Computer, jeder physische Zusammenhang als 'eine Berechnung implementierend' angesehen werden. Als Konsequenz daraus wäre die Bezeichnung 'Computer' überflüssig. Es gibt nach Fitz zwei Gegenstrategien: Entweder zu zeigen, dass es physische Syste-

[159] Fitz S.199
[160] Fitz S.201

me gibt, die nicht konsistent als 'eine Berechnung durchführend' klassifiziert werden können. Oder, objektive physische Kriterien zur Unterscheidung von berechnenden und nicht berechnenden Systemen zu finden. "The prospects for the first strategy seem bleak from the start because of our capacity to interpret temporal changes in almost any observable feature of a physical system as a meaningful computational sequence."[161] Die zweite Strategie stimmt nicht viel hoffnungsvoller. Der Ausweg, den schon Searle gewählt hat, bedeutet, dass Computation immer nur relativ zu einem Beobachter möglich ist: 'Computational states are not discovered within the physics, they are assigned to the physics'(s.o.). Dazu meint Fitz:

> Computation cannot be a purely ascriptional, observer-dependent notion and at the same time PCT be a matter of fact, an open question of contemporary physics. Under such a conception of computation, PCT is still a meaningful, empirical project, but our epistemic procedures on which the ascription of computational content is based become a central issue of that project – indeed the very object of investigation.[162]

Im Gegensatz dazu, gibt Jerzy Mycka am Ende seines Papiers eine Evaluation physikalischer Verhältnisse, unter denen Hypercomputation möglich wäre: "The above results do not entitle us to accept the thesis that hypercomputability is possible in our world. They show, however, that the possibility of crossing the Turing machines barriers is, in the light of some physical theories, real. [...] The boundaries of computability become valid only for a stated physical theory."[163] Mycka präsentiert zwei physikalische

[161] Fitz S.204
[162] Fitz S.206
[163] Mycka S.350

Theorien, in deren Rahmen ’Hypercomputation’ in obigem Sinne möglich wäre: Zum einen die Newtonsche Mechanik und zum anderen Einsteins allgemeine Relativitätstheorie. Im Fall der Newtonschen Mechanik wissen wir bereits, dass sich die Natur nicht daran hält. Aber für die Einsteinsche Relativitätstheorie gilt: “The one which is regarded to be valid nowadays, allows conducting an infifnite number of operations in limited time of some precisely chosen observer.”[164] Abgesehen davon, dass ein solches Setting nicht leicht herzustellen ist, hätten wir gänzlich andere Sorgen, sollte dies jemals in Wirklichkeit durchführbar werden. Nicht nur Schachspielen wäre dann von all seinen Reizen befreit.

[164] Mycka S.350

2.4 Die Kybernetische Gleichung

In diesem Kapitel soll in Anlehnung an Erich Hörls Beitrag 'Das kybernetische Bild des Denkens'[165] untersucht werden, wie es zur Überblendung der Vorstellung vom menschlichen Denken mit der Vorstellung einer Symbolischen Maschine kam. Dieser Kurzschluß wird, je nach Perspektive, als heilsame Entwicklung oder als Irrweg beschrieben. Heilsam gegen dogmatische Vorschreibungen der Philosophietradition was den Begriff des Denkens betrifft. Als Fehlleistung in Bezug auf den Status, der obiger Gleichung gerade in Kognitivistischen Theorien immer noch eingeräumt wird.

2.4.1 Heidegger und die Sache des Denkens

Erich Hörl zieht in seinem Aufsatz 'Das kybernetische Bild des Denkens'[166] Bilanz über das vergangene, in seinen Worten 'noographische Jahrhundert' (Noologie kommt von Gr. 'noein', Dt. 'denken'). Er identifiziert das 'Denken' als Angelpunkt der wichtigsten philosophischen Auseinandersetzungen nach 1900: "Die Sache des Denkens war eine, wenn nicht die große Obsession des zwanzigsten Jahrhunderts."[167] Auf der einen Seite steht Heideggers Programm zum Abbau der bisherigen logozentrischen Überlieferung:

> Daß Heideggers Diagnostik zur Sache des Denkens auf einen letztlich seit Aristoteles geschichtlich gleichbleibenden Sinn von Logik und Mathematik vertraute und in seiner Forderung nach einem 'anderen', nichtrechnenden Denken alles auf die Karte

[165] Hörl, Erich: Das kybernetische Bild des Denkens. In Hager, Michael. und Hörl, Erich (Hrsg.): Die Transformation des Humanen. Band 1848, Frankfurt am Main: Surkamp Verlag, 2008 S.163 – 195
[166] Hörl S.163 – 195
[167] Hörl S.165

> der Dichtung setzte, gegen das 'mathem' das 'poem' in Stellung brachte und es mit der neuen Aufgabe des Denkens verband, zeigt vielleicht den blinden Fleck seines Programms, das Insistieren einer gewissen philosophischen Romantik und einer unhaltbaren gegenmathematischen und gegenlogischen Reaktion, ja, wenn man so will, einer bestimmten gegenphilosophischen, weil zutiefst deutschen Tradition.[168]

Und auf der Anderen Seite steht die 'denkende' Maschine:

> "Das Denken – das eben markiert die Zäsur – sollte nunmehr zuallererst als eine Schrift von Kalkülen erscheinen, seit Turing als universale Maschine, die alle anderen Maschinen nachahmen, später hieß es: simulieren kann – eine Maschine, deren Modell einen Mathematiker bei der Arbeit verkörperte, der Mathematiker als Ensemble von Radiergummi, Bleistift, Papier und strikter Disziplin."[169]

Da nun das Denken in Form von Symbolen angeschrieben werden konnte, und diese auch Maschinen durchlaufen können, schien das Denken umgekehrt als Schrift neuronaler bzw. zerebraler Prozesse lesbar zu werden. Durch die überlieferte Gegenüberstellung des Denkens einerseits, und der Materialität der Schrift andererseits, geriet die rasante Entwicklung von Wissenschaft und Technik ins Wanken. Dass der klare Gegensatz zwischen dem inneren lebendigen Denken des Bewußtseins und dem sturem, äußerlichen und toten Rechnen immer mehr an Kontur verlor, nagte seit längerem an der Selbstbehauptung der Philosophen gegenüber den mathematisch-technischen Fortschritten der Neuzeit.[170] Am Anfang des

[168]Hörl S.165
[169]Hörl S.168
[170]Vgl. Hörl S.167

20. Jahrhundert entsteht dann eine große Auseinandersetzung darüber, ob eine symbolische Maschine denkt, beziehungsweise umgekehrt 'das Denken eine symbolische Maschine darstellt'. Oder ob nicht doch, darüber hinaus, ein Vorteil des menschlichen Denkens in – nicht formalisierbarer – Intuition und Anschauung besteht. "Spätestens seit dem neukantianischen und dann um 1900 beginnenden phänomenologischen Abwehrkampf gegen die experimentalpsychologischen, logisch-mathematischen und zuletzt anthropologischen Übernahmeversuche war klar, daß die Zukunft der Philosophie untrennbar mit dem epistemischen Schicksal der Sache des Denkens selbst verbunden war."[171] Erich Hörl ortet vier verschiedene Reaktionsweisen in der Philosophie. Erstens wurde die Aufgabe des Denkens in der Philosophie von 'bewußtseins- und subjektphilosophischen' Voraussetzungen befreit, indem dieser Teil der dogmatischen Überlieferung aufgegeben wurde. Zweitens in Gestalt der technisch-naturwissenschaftlich überhöhten maschinenfaszinierten 'Philosophy of Mind'. Drittens Rückzugsgefechte, bei welchen im Namen von 'Intentionalität, Intuition, Leiblichkeit oder Gestalt, background und personal knowledge' ein lebendiger Sinn gegen die algorithmisch sturen Maschinen in Stellung gebracht wird. Und viertens ein neu geprägter Assoziationismus, "der unter dem Stichwort eines pensee-cervaux oder Hirndenkens statt einer sauberen Schrift des Denkens 'wuchernde enzephalische Strukturen' sich abzeichnen sah."[172]

[171] Hörl S.168
[172] Vgl. Hörl S.169

2.4.2 Transklassische Maschinen

> Mit dem Auftauchen transklassischer, nicht mehr Arbeit, sondern Information prozessierender Maschinen geschah der Eintritt in eine 'neue Stufe der technischen Welt', ja, zog eine 'neue Seinsart der Technik' herauf, in der es zum Kurzschluß kam zwischen geistigen und technischen Prozessen.[173]

Das alte, selbstverständliche Primat des menschlichen Denkens, das sich selbst das Nächste ist, hat ein Ende. Das kybernetische Bild des Denkens, die Analogisierung von Rechenmaschinen und Gehirnen, stellt die Anhänger einer cartesianischen Auslegung des Denken vor einen Scherbenhaufen. Das Denken wurde eine Angelegenheit von 'Netzwerken idealisierter Neuronen', und so musste das, was die Überlieferung Geist und Bedeutung nannte, direkt aus den physikalischen Kausalstrukturen kommen. Es konnte also nicht nur ein Effekt der Bearbeitung von Symbolen nach bestimmten Regeln sein, sondern mußte als direkte Folge der materiellen Verkörperung logischer Maschinen, sowohl in natürlichen als auch künstlichen Systemen auffindbar sein. Erich Hörl zitiert Jean-Pierre Dupuy, der meint, dass der Kybernetiker durchwegs behauptet, dass "die Gesetze der Physik berechenbar und deshalb Computermodelle und Formalismen zur präzisen Beschreibung und Bestimmung physikalischer Phänomene perfekt geeignet sind." Und umgekehrt, dass "Logik in Materie verkörpert und ausgedrückt ist, in natürlichen wie in künstlichen Systemen."[174] Obwohl dieser Standpunkt viele Schwierigkeiten in sich birgt und uns weiter unten noch beschäftigen wird, besteht das Verdienst eines solchen kybernetischen Bildes im Aufbrechen alter Dogmen, "im unlearning dessen,

[173] Hörl S.170
[174] Hörl S.171

was bislang als fraglos und gewiß galt"[175]. Es gibt keine Grenze, keine klare Unterscheidung mehr zwischen dem Denken und – mit einem Wort Husserls – dem sinnlosen Rasseln einer Maschine.

2.4.3 Das Primat des Neuralen

Für den Mathematiker George Boole liegen die Gesetze, denen der Geist selbst unterworfen ist, (noch) im Dunkeln. Sie sind für den Geist selber maximal wissbar, aber weder versteh- noch hinterfragbar. Es ist lediglich möglich herauszufinden, welche ideale Ausdrucksform für die Wissenschaft die 'letzten Denkgesetze' möglich machen, beziehungsweise vorschreiben. Diese ideale Ausdrucksform soll in einer reinen Kalkülschrift gefunden werden.[176]

Warren McCulloch hat sich hundert Jahre später die Erforschung der Materialität der bis dahin undenkbaren Geistesvorschriften zur Aufgabe gemacht. Bereits 1917, also mit 18 Jahren, hat der spätere amerikanische Neurophysiologe und Kybernetiker sein Programm vor Augen: "What Is a Number, that a Man May Know It, and a Man that He May Know a Number?"[177] Seine Forschung stellt die Frage nach den physikalischen Zusammenhängen der Welt, die die Physikalität des Wissenden, der sie denkt, miteinbezieht. Die Modellierung des physiologischen Substrats des Wissenden steht im Mittelpunkt. "Das Nervensystem galt als 'logische Maschine par excellence', die – in Anlehnung an Hans Blumenberg – vorahmte, was die symbolischen Maschinen immer nur nachahmten. McCulloch war auf der Suche nach einer 'realistischen Logik', die auf der Urlogik des

[175] Hörl S.171
[176] Vgl. Hörl S.172
[177] McCulloch, Warren S.: What Is a Number, that a Man May Know It, and a Man, that He May Know a Number? In Embodiments of Mind. Cambridge: The MIT Press, 1988

Nervensystems basierte."[178] Wenn sich für jeden denkbaren Gedanken das adäquate, formal äquivalente neuronale Netz, das ihn schaltet, angeben ließe, schwände der Unterschied zwischen 'erzeugten' und gezeugten' Rechenmaschinen. Hörl betont, dass McCulloch eine 'Nerventheorie des Wissens' vorschwebte und zitiert eine Stelle aus dem Artikel 'Mysterium Inquitatis of Sinfol Man Aspiring into the Place of God', hier der ganze Absatz aus dem Original:

> Russell has already noted that the explanation of mind has become more materialistic only as our matter has become less material. So we seem to be groping our way toward an indifferent monism. Everything we learn of organisms leads us to conclude not merely that they are analogous to machines but that they are machines. Man-made machines are not brains, but brains are a very ill-understood variety of computing machines. Cybernetics had helped to pull down the wall between the great world of physics and the getto of the mind. [179]

Von einer Subjekt-Objekt-Relation kann keine Rede mehr sein, wenn das Denken zum subjektlosen neuronalen Ablauf wird.
Warren Mcculloch versuchte, gemeinsam mit Walter Pitts, ein Kalkül für Nervenaktivitäten zu entwickeln. Sein Team am MIT untersuchte unter anderem das Auge eines Frosches. Und sie machten die Entdeckung, dass die Information, die das Auge an das Gehirn weitergibt, nicht bloß in den Daten eines visuellen Bilsd besteht, sondern wesentlich vorstrukturiert ist. Ein weiterer Protagonist der 'Epochengleichung von Rechenmaschine und Gehirn' war John von Neumann. Er geht in seiner Formulierung des kyber-

[178] Hörl S.174
[179] McCulloch S.163

netischen Programms noch weiter, indem er behauptet, dass eine Untersuchung des Nervensystems neue Erkenntnisse in den Grundlagen von Mathematik und Logik bringen wird. Analog zu dem Ablauf, dass heutige Computer meist in einer höheren Programmiersprache programmiert werden, die dann durch ein Assembler-Programm in die entsprechenden Maschinenbefehle übersetzt wird, nahm von Neumann an, dass es eine Art Primärsprache des zentralen Nervensystems gibt, und unsere Kenntnisse von Mathematik und Logik wesentlich sekundär sind.[180]
Mit Bezug auf Georg Klaus, einen Kybernetiker aus der ehemaligen DDR, der die Kybernetik im Anschluss an Freud als vierte narzistische Kränkung des Menschen bezeichnet, formuliert Erich Hörl: “Nach der kosmologischen, der biologischen und der psychoanalytischen Desillusionierung des Menschen erschien die Kybernetik als Erschütterung seiner lang dauernden noologischen Illusion.”[181] Bevor die Kybernetik später selbst als Traum von einer fundierten, undogmatischen Wissenschaft in die Geschichte einging, konnte sie durchaus auch als Teil der selbstregulierenden Kräfte der Wissenschaften interpretiert werden, der “die Philosophie zur Neufassung der nach- oder nichtmetaphysischen Aufgabe des Denkens zwang. Sie zeigte für einen Moment die selbstdislozierende Kraft wissenschaftlicher Praxis und deren subversives Potential”[182]

2.4.4 Gotthard Günthers transklassischen Spekulationen

Gotthard Günther ist ein deutscher Philosoph und Logiker der 1937 vor den Nazis über Italien nach Südafrika flüchtete und schließlich 1940 in

[180] Vgl. Hörl S.177f
[181] Hörl S.181
[182] Hörl S.182

die USA emigrierte. Ab 1960 verband ihn eine Freundschaft mit Warren McCulloch und seine Publikationen wurden von Martin Heidegger mit Interesse verfolgt.
In dem von Erich Hörl zitierten Artikel 'Die zweite Maschine'[183] beschreibt Günther die Entwicklung von den ersten Werkzeugen, die die frühen Menschen verwendet haben mögen – in einem Sinne wie es die Primaten immer noch tun – bis hin zur zweiten Maschine, die nicht mehr Arbeit produziert, sondern deren Arbeit im prozessieren von Information besteht. Im Gegensatz zu den auch von Tieren werkzeughaft benützen Gegenständen der Natur, die allenfalls den Funktionsbereich des eigenen Körpers ausweiten, steht das intendierte, für einen Zweck hergestellte menschliche Werkzeug. Durch die Ablösung von Natur und Geist besitzt es eine eigene Existenzform. Ähnlich wie bei Georg Simmel das Geld eine Distanz zwischen dem Menschen und seinen Zwecken schafft, oder sich bei Cassirer das Werkzeug zwischen den Ansatz des Willens und das Ziel stellt, so ermöglicht erst das Werkzeug den Übergang von der magischen zu einer erklärbaren technischen Weltsicht.[184] In der Geschichte der Technik vollzieht sich ein Ablösungsprozeß. Ein in der Natur gewachsenes Stück Holz wird als Hebel verwendet. Auf der Seite des Subjekts, führt die Verselbständigung des Werkzeugs zur Maschine. Die menschliche Kraft, die noch beim einfachen Hebel notwendig war, wird beim Mühlrad durch die Kraft des Wassers ersetzt. Das drückt sich auch in unserem Sprachgebrauch aus: "Ein Werkzeug wird gehandhabt. Eine Maschine wird bedient."[185]

[183]Günther, Gotthard: Die 'zweite' Maschine. In Das Bewußtsein der Maschinen. Baden-Baden und Krefeld: Agis-Verlag, 1963
[184]Vgl. Cassirer, Ernst; Orth, Ernst W. und Krois, John M. (Hrsg.): *Symbol, Technik, Sprache*. Band 372, Philosophische Bibliothek. Hamburg: Felix Meiner Verlag, 1985 S.61
[185]Günther: *Die 'zweite' Maschine* S.182

Halbautomatische Maschinen, wie zum Beispiel das Auto, bedürfen noch eines menschlichen Bedieners, steuern aber gewisse Prozesse selbständig (etwa den Öldruck oder den Zündfunken). Andere Maschinen, wie das elektrische Thermostat bedürfen nicht einmal mehr der Bedienung. Der bisherige Maschinentyp, verrichtet seine Arbeit durch die Bewegung seiner mechanischen Teile. Seine Funktion beruht weitgehend auf dem archimedischen Hebelprinzp und der zugehörige Prototyp ist der menschliche Körper. Dem gegenüber steht für Günther die trans-klassische, nicht-archimedische Maschine. Sie arbeitet ohne klassisch mechanisch bewegliche Teile. Ein Transformator oder ein Transistor etwa wären dieser Art zuzuordnen. "Die Arbeitsweise der klassischen Maschine folgt dem Vorbild des arbeitenden Armes (samt Hand). Die Idee der trans-klassischen Maschine aber erwächst aus den technischen Forderungen, einen Mechanismus zu entwickeln, der nach der Analogie des menschlichen Gehirns arbeitet."[186]

Eine Maschine des archimedischen Typs liefert 'Arbeitsvorgänge', eine nicht-archimedische Maschine soll imstande sein, die Arbeitsvorgänge der klassischen Maschine zu lenken. "Die Idee der kybernetischen Maschine zielt also auf die konstruktive Verwirklichung eines Mechanismus, der Daten aus der Außenwelt aufnimmt, sie als Information weiterverarbeitet und dieselbe in Steuerungsimpulsen an die klassische Maschine weitergibt."[187]

Der Zweck der 'ersten', klassisch archimedischen Maschine ist es, 'Arbeit' zu leisten, und neben diese ist die Idee der 'zweiten' Maschine getreten, die 'Information' produziert.

Dazwischen verstreut findet sich bei Günther – wie bei McCulloch – die Hoffnung, dass sich jene informationsverarbeitenden Mechanismen, die

[186] Günther: *Die 'zweite' Maschine* S.184
[187] Günther: *Die 'zweite' Maschine* S.186

auch dem Gehirn zugrundeliegen, als examinierbar erweisen: Trans-klassische Maschinen “sind technische Zwischengebilde, die sich dadurch von dem archimedischen Maschinentyp unterscheiden, daß in ihnen der Mechanismus in subatomare Bereiche verlegt worden ist. Insofern folgen sie in der Tat einem nicht-klassischen Prinzip.”[188] Nach Günther sind die bisherigen Rechenmaschinen noch keine echten trans-klassischen Konstruktionen, da sie die Information nur transformieren und nicht – wie das Gehirn – selber produzieren. Die Eingrenzung auf ’bisherige’ Rechenmaschinen betrifft in Günthers Fall natürlich das Erscheinungsdatum des Artikels 1963. Es ist allerdings delikat, an welcher Stelle er die Unterscheidung ansetzt: “Selbstproduktion von Information, gleichgültig, ob sie partiell oder total ist, aber setzt Bewußtsein voraus. [...] Damit konzentriert sich das Problem der nicht-archimedischen Maschine auf das des ’mechanical brain’, d.h. des mechanischen Bewußtseins.”[189]

Die Konstruktion eines höheren, dem des Menschen ähnlichem Bewusstseins, schließt Günther aus Gründen einer nicht- hintergehbaren Sprachhierarchie aus.[190] Vom Tier leitet er ab, dass einfaches Bewußtsein ohne Selbstbewußtsein jedoch möglich, und durchaus auch maschinell realisierbar sei. Die Grundlage für die Realisierung eines einfachen Bewußtseins bilden ein Speicher und eine einfache logische Sprache. Mit der Konstruktion eines Regelkreises (einer einfachen Rückkoppelung) sind alle Bedingungen für ein einfaches Bewusstsein erfüllt. Ein Sensor liefert Information an die Steuerung. Diese verarbeitet die Eingangsdaten und gibt die Ausgangsdaten an einen Aktor weiter. Der Aktor greift ins Geschehen ein, wodurch der Sensor mit einem veränderten Zustand konfrontiert ist. Der

[188] Günther: *Die ’zweite’ Maschine* S.186f
[189] Günther: *Die ’zweite’ Maschine* S.187
[190] Vgl. Günther: *Die ’zweite’ Maschine* S.188f

Regelkreis ist sich – innerhalb gewisser Grenzen – des gemessenen und gesteuerten Phänomens bewusst. Dem Konstrukteur bleibt jedoch immer eine Überlegenheit in Form einer Metasprache. Dadurch ist es prinzipiell nicht möglich, die schöpferische Tätigkeit des menschlichen Bewusstseins zu reproduzieren. Es muss nach Günter, wenn wir eine dem menschlichen Gehirn ebenbürtige Logikmaschine konstruieren, die eine arithmetische Maschine bedienen kann, eine 'transzendentale' Steuerung für dieses System geben. Und dieser Steuerungsmechanismus müsste einen höheren logischen Typus repräsentieren, als unsere 'normale klassisch aristotelische' Logik. "Bruchstücke solcher Logiken, die unsere 'normale' Logik als Spezialfall enthalten, existieren schon jetzt. Man nennt sie mehrwertige Logiken, und die theoretischen Mittel, sie weiter zu entwickeln, sind längst vorhanden."[191]

2.4.5 Eine zweite Logik

Erich Hörl skizziert, dass die Lebensfrage Gotthard Günthers – das neue, trans-klassische Bild des Denkens – bereits in seiner Dissertation grundgelegt ist. Es basiert auf Hegels Herangehensweise an die Logik. Hegel hatte dieser Sichtweise gemäß 'stillschweigend' die Geltung der traditionellen aristotelischen Logik, als erschöpfende Definition des Denkens bestritten. "Die überlieferte Logik deckte nach Hegel nur einen besonderen Bereich des Denkens ab, nämlich das Denken des Seins, das objektive Denken also, das Denken von dem, was ist, oder das, was seit Kant Erkennen oder Erkenntnis hieß. Das Denken des Denkens aber, das nicht mehr den Inhalt des Denkens dachte, sondern den Denkprozess selbst,

[191] Vgl. Günther: *Die 'zweite' Maschine* S.196

das Denken des Denkens des Seins also, das eben war eine andere Art des Denkens, die sich nicht mehr mit den traditionellen logischen Mitteln des Seinsdenkens darstellen ließ. Das Denken des Denkens lag in einem für das klassische Denken unfaßbaren, undarstellbaren, undenkbaren Bereich. Zwischen der 'ersten Logik' des Seinsdenkens und der 'zweiten Logik' der Reflexion klaffte ein 'grundlegender Struktur und Geltungsunterschied'."[192] Analog zur 'zweiten Maschine' entstand bei Günter die Idee einer 'zweiten Logik'. Diese stellt eine neue Reflexionssituation für das logische Bewußtsein her, indem das Denken sich nicht mehr nur zweiwertig als unmittelbarer Gegensatz zu seinem Objekt versteht. In einem dritten Schritt wird nun auf die zweiwertige Struktur des klassisch-ontologischen Denkens reflektiert. Von der 'alten' Position aus betrachtet, für die Zweiwertigkeit und Formalisierbarkeit zusammenfallen, muss sich die 'zweite Logik' im "Jenseits alles Formalen und Formalisierbaren befinden und damit als Reich des Unberechenbaren gelten."[193] Dass um 1800 noch keine ausreichenden formalistischen Kalküle zur Verfügung standen, soll die Ursache für Hegels "Verwerfung jeglicher Form als logisch-kritischer Grundlage philosophischer Systeme überhaupt"[194] gewesen sein. Die Folge war die 'antilogische' und 'antimathematische' Träumerei der romantischen Philosophie dieser Zeit.

Es ist freilich nicht so ganz klar, worin nun die 'zweite Logik' – die neben Geist und Materie als drittes auch Information einschließt – bestehen soll. Entgegen der ersten Intuition, es möge sich um eine dreiwertige (oder mehrwertige) Logik handeln, stellt sich heraus, dass es sich, so wie in 'Die

[192] Hörl S.182
[193] Hörl S.190
[194] Hörl S.191

klassische Metaphysik und das Problem der Kybernetik'[195] beschrieben, eher um eine Kombination von drei zweiwertigen Logiken handelt. Herbert Hrachovec kritisiert die Position Günthers als Verwegenheit: "[Er] erhebt den Anspruch, die zweiwertige Logik von außen, mit Hilfe des Instrumentariums des deutschen Idealismus, zu revolutionieren."[196] Während auf der anderen Seite Erich Hörl, Günthers kybernetische Reinterpretation der Transzendentalphilosophie, im Speziellen den Vergleich des menschlichen Denkapparates mit trans-klassischen Maschinen, als logische Gegenbewegung zur Romantisierung und Poetisierung der Frage des Denkens sieht, und ihnen eine gewisse Desillusionierungskraft bezüglich der eingesessenen Interpretationen zugesteht.

[195] Günther, Gotthard: Die klassische Metaphysik und das Problem der Kybernetik. In Das Bewußtsein der Maschinen. Baden-Baden und Krefeld: Agis-Verlag, 1963 S.38ff

[196] Hrachovec, Herbert: Gotthard Günthers Geltung, oder: die Grenzen der Geduld in Cybernetics. In Pias, Klaus (Hrsg.): Kybernetik. Band 2, Zürich-Berlin: Diaphanes, 2004 S. 263-275

2.4.6 John Searle, Gehirne und Computer

Diese beiden Abschnitte zu Searle passen nicht originär in dieses Kapitel, da sie aber – von der Perspektive meiner Fragestellung her gesehen – ein ähnliches Thema behandeln, sind sie hier gelandet.

John Searle betrachtet das Bewusstsein als ein rein biologisches Phänomen: “It [the brain] is a specific biological organ and its specific neurobiological processes cause specific forms of intentionality. In the brain, intrinsically, there are neurobiological processes and sometimes they cause consciousness. But thats the end of the story.”[197] Zur Ehrenrettung von Oron Shagrir (Vgl. Beispiel in der Einleitung) muss ich anfügen, dass er, wie auch ich, mit Searles Schlussfolgerungen nicht ganz zufrieden sein kann. Insbesondere damit, dass anscheinend überhaupt nichts über das Verhältnis von Gehirnen und Computern, sowie über Computer überhaupt aussagbar ist. Searles Dilemma liegt aber nicht etwa darin, dass diese Schlussfolgerungen falsch wären – die Schwierigkeit erwächst aus seinen Grundannahmen. Oron Shagrirs Text ’Why we view the brain as a Computer?’ aus dem Jahr 2006 ist in weiten Teilen eine Bearbeitung von Searles Text ’Is the Brain a Digital Computer?’ von 1990, bei Shagrir in der Ausgabe von 1992 zitiert. Darin versucht Searle quasi in einem Rundumschlag mit den Kognitivismus-Märchen aufzuräumen. Shagrir beherzigt zwar den größten Teil von Searles Vorschlägen und goutiert deren Grundannahmen, will aber die radikale – und notwendige – Folgerung nicht mitmachen. Aber nun zum Text: Searle formuliert eine Art Kern- oder Urgeschichte der künstlichen Intelligenz, die auf den Texten von Alan Turing basiert und deren erster Teil besagt:

[197] Searle S.13

> We begin with two results in mathematical logic, the Church-Turing thesis (or equivalently, the Churchs's thesis) and Turing's theorem. For our purposes, the Church-Turing thesis states that for any algorithm there is some Turing machine that can implement that algorithm. Turing's thesis says that there is a Universal Turing Machine which can simulate any Turing Machine. Now if we put these two together we have the result that a Universal Turing Machine can implement any algorithm whatever.[198]

Der zweite Teil:

> It is clear that at least some human mental abilities are algorithmic. For example, I can conciously do long division by going through the steps of an algorithm for solving long division problems. It is furthermore a consequence of the Church-Turing thesis and Turing's theorem that anything a human can do algorithmically can be done on a Universal Turing Machine. I can implement, for example, the very same algorithm that I use for long division on a digital computer. In such a case, as described by Turing ('Computing, Machinery and Intelligence' 1950), both I, the human computer and the mechanical computer are implementing the same algorithm, I am doing it consciously, the mechanical computer nonconsciously.[199]

Gibt es gute Gründe, anzunehmen, dass unser Gehirn (nur) eine Universale Turing Maschine ist? Searle ortet drei Arten von Fehlern in kognitivistischen Texten: Der erste Fehler bestehe in der Ansicht, die einzige

[198] Searle S.2
[199] Searle S.2

Alternative zu einem Dualismus von unsterblicher cartesianischer Seele und vergänglichem Körper bestünde darin, dass das Gehirn ein Computer ist. Der zweite Fehler besteht laut Searle in der Auslegung der Frage, ob Gehirnprozesse den Prozessen in einem Computer ähnlich ('computational') sind, als rein empirische Frage. So wie zum Beispiel das Herz als Pumpe oder ein Gelenk als Scharnier gesehen werden kann. Der dritte Fehler besteht im Fehlen einer – auch nur teilweise haltbaren – Definition für das, was wir 'Computer' nennen. Folgen wir weiter dem Text: Searle zieht für seine 'Definition of Computation' noch einmal Alan Turing heran:

> According to Turing, a Turing machine can carry out certain elementary operations: It can rewrite a 0 on its tape as a 1, it can rewrite a 1 on its tape as a 0, it can shift the tape 1 square to the left, or it can shift the tape 1 square to the right. It is controlled by a program of instructions and each instruction specifies a condition and an action to be carried out if the condition is satisfied.[200]

Dabei merkt er noch an, dass es irrelevant sei, wie in einem digitalen System eine '0' oder eine '1' realisiert iat. Daraus folgt: "[...] the irrelevance of hardware realization to computational description."[201] Eine solche Realisierung könnte optisch, mechanisch, hydraulisch sowie auch elektrisch bzw. elektronisch – wie ja in den meisten Fällen – sein. In der Formulierung von Pylyshn könnte es sogar eine Gruppe von Tauben sein, die trainiert ist, eine Turingmaschine emulierend Körner zu picken. Wichtig ist nur, dass es etwas gibt, das wir als '0' und '1' ansehen können (Allgemeiner könnte man formulieren: Wichtig ist nur, dass es diskret unterschiedene

[200] Searle S.4
[201] Searle S.5

Zustände gibt, denn es ist irrelevant, auf Basis welchen Zahlensystems ein Computer realisiert wird. Dazu später).

2.4.7 Syntax und Homunkulus

Da die relevanten Eigenschaften ganz unabhängig vom physischen Material sind, ergibt sich für Searle die erste Schwierigkeit: "Computational states are not discovered within the physics, they are assigned to the physics."[202] Somit ist nach Searle das, was einen Computer ausmacht immer von einem Beobachter abhängig: "The Chinese Room Argument[203] showed that semantics is not intrinsic to syntax. I am now making the separate and different point that syntax is not intrinsic to physics. For the purposes of the original argument I was simply assuming that the syntactical characterization of the computer was unproblematic. But this was a mistake."[204] Und daraus folgt: "There is no way you could discover that something is intrinsicaly a digital computer because the characterization of it as a di-

[202] Searle S.6

[203] Das Chinesische Zimmer (The Chinese Room Argument): Man soll sich einen einsprachigen Eglischsprecher vorstellen, der in einen Raum gesperrt ist. Er bekommt einen großen Stapel Papier mit chinesischen Schriftzeichen, einen weiteren kleinen Stapel mit chinesischen Schriftzeichen und einen Satz Regeln auf Englisch. Er bekommt einen dritten Stapel mit chinesischen Zeichen und einer englischen Anweisung, die es ihm ermöglicht, die Symbole der ersten beiden Stapel mit dem dritten in Verbindung zu bringen. Die Anweisung besagt weiters, dass er die mittels der Regeln von den erhaltenen Schriften abgeleiteten chinesische Symbole ausgeben soll. Diejenigen, die ihm die Stapel überreichen, nennen den ersten Stapel 'script' den zweiten 'story' und den dritten 'questions', die englischen Regeln 'program' und die Symbole, die er wieder ausgibt, werden 'answers to the questions' genannt. Der Proband versteht nichts von all den Chinesischen Symbolen, obwohl ein aussenstehender Chinese die Antworten als 'Richtig' identifiziert. Der Proband verhält sich wie ein Computer. Umgekehrt versteht ein Computer, so gut er auch programmiert ist um sich wie ein Mensch zu verhalten von alledem nichts – er ist nicht wirklich intelligent, er ist nur am Abarbeiten von Regeln. "It's not actually thinking. Its internal states and processes, being purely syntactic, lack semantics (meaning); so, it doesn't really have intentional (i.e., meaningful) mental states." Vgl. Borchert, Donald M.: *Encyclopedia of Philosophy*. Band 10, Detroit: Thomson Gale, 2006

[204] Searle S.6

gital computer is always relative to an observer who assigns a syntactical interpretation to the purely physical features of that system."[205] Es ist keine Frage, dass es möglich ist, in der Natur Objekte zu finden, die so aussehen wie 'Sessel' und daher auch so benützt werden können. Jedoch können wir keine Objekte finden, die 'als Sessel funktionieren' außer relativ zu jemandem ('to some agents'), der sie als solche ansieht oder benützt. Wodurch wir zum zentralen Problem seiner Ausführung kommen: 'The Homunculus Fallacy'. Da Syntax kein Teil des Physischen ist, muss immer, wenn eine solche im Spiel ist, auch ein Beobachter anwesend sein, der sie zuweist. Searle führt Versuche an, die einen jeweiligen Homunculus oder Beobachter sukzessive durch eine syntaktische Struktur plus einen kleineren Homunculus ersetzen wollen, so lange, bis er ganz verschwunden wäre. Aber: "Without a homunculus that stands outside the recursive decomposition, we do not even have a syntax to operate with. The attempt to eliminate the homunculus fallacy through recursive decomposition fails, because the only way to get the syntax intrinsic to the physics is to put a homunculus in the physics."[206] Gibt es einen Ausweg aus dem Dilemma?
Searles Analyse ist nicht falsch. Vielmehr sind die Schwierigkeiten eine logische Folge seiner Grundvoraussetzungen. Das betrifft den Stellenwert, den man der Syntax zuspricht, sowie deren Verortung. Searle ist strenger Realist und behauptet 'syntax is not intrinsic to physics'. Syntax ist in seinem Sinne aber nicht mit Logik gleichzusetzen, sondern eher mit diskreter Information – zum Beispiel im Sinne einer Kombination von Programmcode und Daten bei einem Desktoprechner[207] – die zwar irgendwo in der

[205] Searle S.6
[206] Searle S.7
[207] vgl. von Neumann Architektur eines Digitalrechners. Eine Prozessoreinheit mit einem Speicher der sowohl die zu bearbeitenden Daten als auch die Programmanweisungen enthält. Diesem Modell entspricht im Prinzip jeder heutige Desktoprechner.

Physis realisiert sein kann (zum Beispiel in einem Speichermedium), aber als solche nur Bedeutung erlangt, wenn sie interpretiert wird, ihr eine Bedeutung als diskrete Information zugewiesen wird. Als Beispiel dient ihm das Wordstar-Programm, welches genausogut wie im Speicher des Computers auch irgendwo in der Feinstruktur (in der Ordnung der Moleküle, Quarks oder was auch immer) der Wand hinter ihm realisiert sein könnte: “Thus for example the wall behind my back is right now implementing the Wordstar program, because there is some pattern of molecule movements which is isomorphic with the formal structure of Wordstar.”[208] Und das betrifft, neben Programmen und Algorithmem als rein syntaktische Elemente, auch die Ausführung der Berechnung, denn “computational states are not discovered within the physics, they are assigned to the physics.“[209] Und das passt auch, denn im Sinne einer 'von-Neumann-Architektur' wären ja auch diese 'Berechnungsschritte' wieder – wenn auch nur kurz – im Speicher abgelegte Informationen. Einerseits ist für Searle Syntax nicht in der Physis zu finden ('syntax is not intrinsic to physics'), andererseits besteht aber die Möglichkeit die selbe syntaktische Struktur isomorph in einem materiellen Zusammenhang vorzufinden.

Eine weitere Behauptung lautet: 'Syntax has no causal powers'. Strukturierte Information allein, ohne jemanden, der sie interpretiert, verursacht gar nichts, und steht somit nicht auf einer Stufe mit kausalen Erklärungen aus den Naturwissenschaften, die immer einen Beobachter voraussetzen. Zum Beispiel die Evolutionstheorie oder die Theorie der Photosynthese: “In each case a causal mechanism is specified, and in each case

[208] Searle S.5
[209] Searle S.6

the specification gives an explanation of the output of the mechanism."[210] Diese Ansage wendet sich in erster Linie gegen kausale Erklärungstheorien aus den kognitivistischen Wissenschaften. Also gegen die Ansicht, die Funktion der Gehirnprozesse ließe sich mittels der Spezifikation eines 'Computer-Programmes' beschreiben: "So the puzzle is, how do we reconcile the fact that syntax, as such, has no causal powers with the fact that we do give causal explanations that appeal to programs?"[211] Wenn wir uns noch einmal an die Urgeschichte zu Turings Theorem oben erinnern: "In Turing's human computer there really is a program level intrinsic to the system and it is functioning causally at that level to convert input to output. This is because the human is consciously following the rules for doing a certain computation, and this causally explains his performance. But when we program the mechanical computer to perform the same computation, the assignment of a computational interpretation is now relative to us, the outside homunculi."[212] Nun entsteht aber ein Widerspruch, wenn der Kognitivismus versucht uns zu lehren, dass das Gehirn genauso funktioniere wie ein kommerzieller Computer und gerade dies 'cognition' (im umfassenden Sinne von: wahrnehmen, erkennen, urteilen und schließen) verursache. Wenn – nach Searle – ein Computer einen Menschen voraussetzt, der ihn interpretiert, und dieser Beobachter wiederum so funktioniert wie ein Computer, argumentieren wir im Kreis. "Without the homunculus there is no computation, just an electronic circuit."[213] So bleibt nach einer 'computational-explanantion' des Gehirns immer noch ein zu erklärender Homunculus. Dieses Argument besagt also, dass die Funktionsweise eines Gehirnes nicht erschöpfend durch die Beschreibung eines Program-

[210] Searle S.8
[211] Searle S.9
[212] Searle S.8f
[213] Searle S.11

mablaufes darstellbar ist. Diese Schlussfolgerung wäre durchaus auch aus anderen Perspektiven argumentierbar und Searles Begründung bleibt problematisch. So mag etwa für Rechner, die programmiert werden um einen einzelnen Algorithmus auszuführen und dann wieder zu halten, durchaus plausibel sein, dass sie auf einen Beobachter – jemanden der sie initiiert und die Ergebnisse auswertet – angewiesen sind. Aber man würde kaum annehmen, dass der moderne Desktoprechner oder die Steuerung, die die Klimaanlage steuert, nur arbeitet wenn jemand zusieht.
Das Homunkulusargument verläuft umgekehrt parallel zu Günthers Argument. Während bei Günther immer eine reichere Metasprache existieren muss, innerhalb der die Bedeutung – etwa eines 'Programmcodes' – bestimmt werden kann, reicht bei Searle ein ganz kleiner, dummer Homunkulus der einer Syntax Bedeutung verleiht. Da es aufgrund des Gödelbeweises nicht möglich ist, mit einem Beweis in einem Axiomensystem X die Abgeschlossenheit desselben Axiomensystems zu zeigen, bedeutet das für Günther, dass immer noch eine Metasprache mit einem reicherem Axiomensystem vorhanden sein muss. Für den Konstrukteur einer 'zweiten Maschine' muss immer noch eine Metasprache zur Bedienung und zur Konstruktion zur Verfügung stehen. Die Bedeutung wird stets in die allgemeinere Sprache verlegt. Searle, der die Intentionalität, und damit die Bedeutung in ein punktförmiges Subjekt verlegt, muss diesen Punkt, den Homunkulus immer weitervererben, wenn die Bedeutung nicht verlorengehen soll.
David L. Thompson schreibt über Searles Intentionalität, dass: "[...] the relationship of the mental structures of Intentionality to the physical structures of the brain, a relationship that Searle understands as causal."[214]

[214] Thompson, David L.: Intentionality and Causality in John Searle. Canadian Journal of Philosophy, 16 1986, Nr. 1

Wenn Intentionalität und damit 'Bewusstsein' als kausales Phänomen von biologischen Prozessen gesetzt ist, aber "in case of the Brain none of the relevant neurobiological processes are observer relative"[215], dann bleiben kausale Phänomene zurück, die beobachterunabhängig sind. Anders gesagt, wie kann ein rein kausales Phänomen nicht-kausale Intentionen hervorbringen? Noch einmal anders: Für eine gesetzte, nicht-kausale Intentionalität können wir klarerweise keine kausale Erklärung finden. Der Homunculus wird in dem Moment in die Theorie eingebaut, wo der physikalischen Welt eine intrinsische Kausalität attestiert wird. Und so erkauft sich Searle die Freiheit vom cartesianischen Dualismus durch eine Schar von Homunculi – oder Erklärungsverweigerung: "there are neurobiological processes and sometimes they cause consciousness. But thats the end of the story."[216]

[215] Searle S.12
[216] Searle S.13

3 Schluß

3.1 Zusammenfassung der Argumentation

3.1.1 Die Bestimmung der Rechenwesen

Wenn ich noch einmal den Vergleich mit dem Buch über Vögel strapaziere, so kann ich feststellen, dass sich Vögel von anderen Tieren unterscheiden da sie gefiedert sind, sich von Steinen unterscheiden da sie aus organischem Material sind, sich von Maschinen unterscheiden da sie Lebewesen sind, und so fort (Vgl. Kapitel 2.1.1 über die Kategorien des Aritstoteles). Die natürliche Ordnung drängt mir diese Unterscheidungen beinahe auf und ich würde niemals auf die Idee kommen, dass nur die Zuschreibung eines Beobachters den Vogel ausmacht (Nur ein sehr sehr radikaler Konstruktivist würde das annehmen). Schon bei der Waschmaschine gestaltet sich dieser Zusammenhang jedoch etwas anders. Sie ist durch ihr Maschine-Sein und ihre Funktion charakterisiert. Ich kann zwar auch einen Steinbottich als Waschmaschine bezeichnen, wenn ich ihn dazu verwende Wäsche zu waschen, der Beliebigkeit sind aber Grenzen gesetzt. Mit dem Computer verhält es sich ähnlich wie mit der Waschmaschine, nur dass auf den ersten Blick der Funktionsaspekt nicht in der selben Art und Weise eingeschränkt ist. Ein Computer hat meist keine einzige, spezielle Funktion, ihm wird vielmehr eine gewisse Universalität bezüglich seines Funktionierens zugesprochen. Allerdings besteht dieses Funktionieren nicht im Waschen von Wäsche oder Sägen von Holz, sondern im Manipulieren von Symbolen. Es besteht kein Zweifel, ein Computer ist eine Symbolische Maschine. –

Wie sich in den Kapiteln 2.1.1 und 2.1.2 gezeigt hat, widersetzen sich

Werkzeuge und Maschinen notorisch der Einordnung in ein hierarchisches, zur Naturbetrachtung entworfenes System. Dadurch, dass sie durch ihre Funktion bestimmt sind und – je nach Herstellung – auf verschiedene Materialien und Formen zurückgegriffen werden kann, lassen sich ihre Unterschiede leichter in einem relationalen Modell abbilden, das keine fixierte Einteilung des 'Seienden' bietet, sondern auf (empirische) Evaluierung angewiesen ist. Trotzdem, oder gerade deswegen, sind Computer auf ein Substrat angewiesen das sich sehr gut mit dem Aristotelischen Begriff der Substanz beschreiben lässt. Abgesehen vom praktischen Aspekt, dass wir Computer als elektronische Bauteile recht platzsparend und sehr leistungsfähig konzipieren können, ist es ganz egal, ob wir sie zum Beispiel als hydraulisches System, als mechanische Anordnung von Hebeln und Zahnrädern[217], optisch mit Hilfe von Spiegeln verwirklichen, oder sogar auf biologischen Substraten aufbauen. Es ist aber immer ein Token[218] notwendig das manipuliert werden kann. Das Kapitel 2.1.3 zeigt, dass die Aristotelische Substanz diese Bedingungen erfüllt, insofern ihr als Zugrundeliegendem etwas Konträres zukommen kann. Konträres bietet sich für ein binäres System an. Die Substanz eines modernen Computersystems sind etwa Tausende oder eher Millionen Schalter, die in Form von Transistoren auf einem Halbleitermaterial verwirklicht wurden.
John Searle – und mit ihm auch Oron Shagrir – sieht das, was ich 'Substanz sein' eines solchen Schalters nenne, als rein von der Zuschreibung durch einen Beobachter abhängig. Insofern ein Schalter zum Beispiel die Werte '0' und '1' repräsentieren kann, kommt ihm zwar Bedeutung zu, aber eben nur dadurch, dass sie jemand zuschreibt. Shagrir schreibt: "[...] it is

[217]Vgl. Charles Babbages 'Difference Engine'
[218]Engl. Zeichen oder Marke; kleinste Sinngebende Einheit in einer Programmiersprache.

we who ascribe to them this representational force."[219] Und Searle geht sogar noch weiter, indem er sowohl Syntax als auch die inneren Zustände eines Computers rein vom Beobachter abhängig macht. Seiner Meinung nach besteht keine Möglichkeit, dass ich etwas, das sich später als Digitalcomputer herausstellt, einfach entdecke.[220] Im Kapitel 2.4.7 wurde näher darauf eingegangen weshalb Searles Konzept der Intentionalität keine andere Zugangsweise zulässt. Der Sprachwissenschaftler Searle, der stets mit dem Artefakt der (geschriebenen) Sprache, die – wenn sie keiner versteht – wohl genauso gut nichts als ein beliebiges Muster auf einem physischen Objekt sein könnte, konfrontiert ist, vergisst auf den Prozess der Herstellung und Materialisierung. Dass die symbolverarbeitende Maschine die Bedeutung dessen, was sie prozessiert nicht versteht, das ist zu ihrem Tun nicht notwendig. Umgekehrt ist nicht klar wie ich (der beobachtende Homunkulus) den Prozessen in meinem Heimcomputer Bedeutung spende, wenn ich mir nicht anmaße, alles zu verstehen was dieser tut.
Die Syntax besteht nun nur noch in den von der Bedeutung abgekoppelte formalen Strukturen, der formale Ordnung einer Zeichenreihe. Die in Kapitel 2.2.1 besprochene Geschichte der Formalisierung zeigt die Entwicklung vom Stäbchenrechnen bis zu formalisierten Kalkülen bei Frege. "Die Grundidee der Formalisierung besteht darin, das Manipulieren von Symbolreihen von ihrer Interpretation abzutrennen."[221] Das Manipulieren der Symbole einer Zeichenreihe, das sich jeweils als Äquivalent zur Anwendung von Rechenoperationen erweist, kann nur bis zu einem gewissen Grad festgelegt werden. Der Gödelsche Unvollständigkeitsbeweis und ihm folgend das Entscheidungs- und Halteproblem torpedieren die Be-

[219] Shagrir S.5
[220] Vgl. Searle S.6
[221] Krämer: *Symbolische Maschinen* S.176

strebungen ein abgeschlossenes, formales System zu erstellen. Der Berechenbarkeitsbegriff kann allen Bestrebungen zum Trotz nicht gesichert festlegen, was eine Berechnung ist und was nicht, aber alle bisher erstellten allgemeinen Formulierungen einens Berechenbarkeitsbegriffes haben sich als äquivalent erwiesen. Somit kann Church seine These formulieren: 'Der intuitiv gegebene, allgemeingebräuchliche Begriff der berechenbaren arithmetischen Funktion ist identisch mit dem exakt definierten Begriff der allgemein-rekursiven Funktion.' Mit dieser nicht beweisbaren Behauptung stellt er in Aussicht, dass mit dem Begriff der allgemein-rekursiven Funktion und äquivalenten Berechenbarkeitsbegriffen alle tatsächlich 'berechenbaren Funktionen' erfasst sind.
Mit Alan Turings Maschinenmodell, das sich ebenfalls als zu den obigen Berechenbarkeitsbegriffen äquivalent erweist, lässt sich die These näher spezifizieren, und besagt nunmehr, dass die Klasse der von einer Turingmaschine berechenbaren Funktionen, genau jener, der intuitiv berechenbaren Funktionen entspricht. So wird die Tätigkeit eines Mathematikers beim Rechnen endgültig auch zur Maschinensache. Im Kapitel 2.2.5 wird, durch Zuweisung einer natürlichen Zahl zu jeder Funktion, der Übergang von der speziellen Turingmaschine zur universellen Turingmaschine (UTM) beschrieben. Die Mächtigkeit der Menge der berechenbaren Zahlen ist so auf eine abzählbar unendliche Menge festgelegt, und somit nur ein Bruchteil der überabzählbar vielen reellen Zahlen. Nur diejenigen reellen Zahlen sind berechenbar, deren Nachkommastellen sich von einer Turingmaschine (oder auch von einem Menschen) in endlicher Zeit aufschreiben lassen. Jeder aktuelle Desktoprechner in Verbindung mit einer höheren Programmiersprache ist im Funktionsumfang einer UTM äquivalent. Er kann, abgesehen von Limitierungen der Zeit oder des Speicher-

platzes jede mögliche Turingmaschine emulieren. Heutzutage erleben wir einen ständigen Zuwachs an Rechengeschwindigkeit und Speicherplatz – und allein durch diesen graduellen Zuwachs erweitert sich der Bereich der effektiv berechenbaren Funktionen. Aber eben nur graduell, zum Beispiel bezüglich der Zeitspanne die es dauert um eine Berechnung durchzuführen.

Es gibt jedoch Spekulationen über einen möglichen 'absoluten' Zuwachs in der Klasse der berechenbaren Funktionen. Berechnungen über die Möglichkeiten der UTM hinaus werden 'Hyperturing' oder 'Hypercomputation' genannt. Wie in Kapitel 2.3.2 dargelegt, sind derartige Maschinen nur mittels theoretischer Tricks denkbar. Es wird aufgezeigt, dass eine Orakelmaschine die etwa, das Halteproblem berechnet, wiederum mit einem neuen Halteproblem konfrontiert ist. Die Spekulation in diese Richtung wird über die Formulierung einer 'physikalischen Church-Turing These' (PCT) noch erweitert (siehe Kapitel 2.3.4). Es besteht die Hoffnung, dass der unsichere Berechenbarkeitsbegriff über die Physikalität der Turingmaschine mit Hilfe der Naturgesetze festgelegt werden kann. Dieses Unternehmen macht jedoch nur Sinn, wenn man der Meinung ist, dass die Naturgesetze tatsächlich das Verhalten der physischen Dinge festlegen (Deutsch, Bennett, Searle, Günther, McCulloch und andere). Wenn so die 'Berechnung' in der Physis entdeckt werden kann, wer bestimmt dann, was eine Berechnung ist und was nicht? (Vgl. 'Searle versucht das Problem durch ein intentionales Subjekt zu lösen'.) Ein Beobachter, der in einem solchen System auch durch die Naturgesetze festgelegt ist, kann diese Bestimmung nicht vornehmen, da ansonsten ein Zirkelschluß entsteht (Vgl. Kapitel 2.3.4 S.73). Ein weiteres Problem taucht auf, da die PCT mit wissenschaftlichen Methoden weder falsifizierbar noch verifizierbar ist. Die Entdeckung

eines ’Hypercomputation’-Prozesses würde die PCT falsifizieren, eine solche Entdeckung ist aber von erkenntnistheoretischer Seite aufgrund der Beschränkung eines etwaigen Experimentes auf endlich viele Versuche auszuschließen. Und auch ’Nicht-Berechnung’ entzieht sich einer Festlegung durch mathematische Beweismethoden (Vgl. Kapitel 2.3.4).
Den Angelpunkt dieser Problemstellung bildet der Analogrechner. So wie ich im Kapitel 2.1.3 erwähnt habe dass der Aristotelischen Substanz ein ’mehr oder minder’ zukommen kann, könnte man meinen, dass ein Analogrechner ebenso als mit reellen Werten rechnend gedacht werden kann, wie ein Digitalrechner mit diskreten Werten rechnet. Ein Analogrechner mit absoluter Präzision[222] könnte über das Limit der UTM hinaus Berechnungen anstellen und somit die PCT falsifizieren. Da dies aber auszuschließen ist, legt sich der Schluss nahe, einem Analogrechner abzusprechen, dass er tatsächlich eine Berechnung durchführt. Angenommen, eine gewisse Spannung wird in einem elektrischen Analogrechner an eine vierfach- Multipliziereinheit angelegt – instantan haben wir am Ausgang das Ergebnis: einen vierfach hohen Spannungspegel. Wir haben es mit einem physischen Phänomen in einer Laborsituation zu tun, aber es ist nicht ersichtlich, wo in diesem Zusammenhang eine Berechnung stattfindet. Die vierfache Multiplikation ist immer schon durch den verwendeten Baustein vorweggenommen und nur beschränkt genau. Für das Integral einer komplexen Funktion gilt dasselbe. Auch wenn der Analogrechner um vieles schneller ein Ergebnis liefert, ist die Berechnung doch immer schon vorweggenommen. Damit bleibt er bloß eine analoge Rechenhilfe zur Bestimmung von Funktionswerten (Vgl. Kapitel 2.3.1). Hartmut Fitz stellt eine ’Turing-Approximation-Hypothesis’ (TAH) auf, die besagt, dass

[222]Sogar nur mit linearer Präzision. Siehe Kapitel 2.3.4

sich die Schwankungen in den Messwerten eines Analogrechners mit den sukzessive besseren Annäherungen des Wertes durch eine Turingmaschine die Waage halten. Um noch einmal einen Vergleich zu strapazieren: einem Menschen sprechen wir zu, dass er rechnet, wenn er tatsächlich eine gewisse Regelfolge ausführt oder wir annehmen, dass er dies tut. Wenn jemand das Ergebnis einer Rechnung schätzt, fühlt, auswendig weiß oder errät, würde ich sagen er hat es geschätzt, gefühlt, auswendig gewusst oder erraten. Ich würde aber nicht behaupten, dass er 'gerechnet' hat.

Hartmut Fitz schlägt vor, zu zeigen, dass es physische Systeme gibt, die nicht konsistent als 'eine Berechnung durchführend' klassifiziert werden können, oder aber, objektive physische Kriterien zur Unterscheidung von berechnenden und nicht berechnenden Systemen zu suchen. Ich behaupte, dass eine Mischung aus beidem möglich ist. Dem ersten Vorschlag kann ich nachkommen, indem ich Analogrechner – gemäß der Argumentation im obigen Absatz – als 'nicht eine Berechnung durchführend' klassifiziere. Damit schließe ich viele Spekulationen (vgl. Shagrir) aber keine erfahrbaren Tatsachen aus. Der Schlüssel zur Identifikation einer Berechnung liegt in der Idetifikation der zugrundeliegenden Substanz. Ein Schalter kann einfach identifiziert werden. Er ist konstituiert durch eine Schalt-Hysterese, im binären Fall zwei Schwellwerte, durch die er auf Ein oder Aus geschaltet wird. Tertium non datur. Selbst wenn ich annehme, dass die Realität nicht auf völlig examinierbare kausale Tatsachen zurückzuführen ist, kann ein solcher Schalter in der Hinsicht, dass er mit ausreichender Wahrscheinlichkeit exakt diese beiden Zustände verkörpert, entdeckt und genutzt werden.

Es gibt keinen Grund anzunehmen, dass der menschliche Verstand rein durch seine Fähigkeit Berechnungen anzustellen bestimmt ist.

> Die Grenzen der Formalisierbarkeit sind die Grenzen eines mechanisch verfahrenden, phantasielosen Verstandes. Die Auszeichnung unserer Vernunft liegt nicht nur darin, einer Regel folgen, sondern auch darin, eine Regel gegebenenfalls außer Kraft setzen zu können. Ohne diese Fähigkeit zur Außerkraftsetzung könnten die Bedingungen, die erfüllt sein müssen, damit wir formale Systeme aufbauen können, überhaupt nicht erfüllt werden.[223]

Dieses 'Mehr' näher zu bestimmen, wird die Aufgabe anderer Arbeiten sein.

> Die Bedeutung der Überlegungen von Gödel, Church und Kleene besteht vielmehr darin, daß sie uns zeigen, daß das Tun des Mathematikers niemals vollständig auf das Berechnen zurückzuführen ist.[224]

[223] Krämer: *Symbolische Maschinen* S.181
[224] Krämer: *Symbolische Maschinen* S.157

Literatur

Aristoteles; Rolfes, Eugen (Hrsg.): Kategorien / Lehre vom Satz. Band 8/9, Philosophische Bibliothek. Hamburg: Verlag von Felix Meiner, 1962, Unveränderter Abdruck der Ausgabe von 1925

Berger, Burkhard: Was ist ein Vogel? Hannover: Landbuch Verlag, 1991, LB-Naturbücherei

Blass, Andreas; Gurevich, Yuri: Algorithms: A Quest For Absolute Definitions. In **Adam Olszwski, Jan Woleǹski, Robert Janusz (Hrsg.):** Church's Thesis After 70 Years. Band 1, Heusenstamm: Ontos Verlag, 2006, 24 – 57

Borchert, Donald M.: Encyclopedia of Philosophy. Band 10, Detroit: Thomson Gale, 2006, 2nd Edition

Cassirer, Ernst; Orth, Ernst W. und Krois, John M. (Hrsg.): Symbol, Technik, Sprache. Band 372, Philosophische Bibliothek. Hamburg: Felix Meiner Verlag, 1985, 2.Auflage

Codd, Edgar F.: The relational model for database management. Reading (Mass.): Addison-Wesley, 1990, korrigierte Auflage

Craig, Edward (Hrsg.): Routledge Encyclopedia of Philosophy. Band 9, New York: Routledge, 1998

Craig, Edward (Hrsg.): Routledge Encyclopedia of Philosophy. Band 6, New York: Routledge, 1998

Fitz, Hartmut: Church's Thesis And Physical Computation. In **Adam Olszwski, Jan Woleǹski, Robert Janusz (Hrsg.):** Church's Thesis After 70 Years. Band 1, Heusenstamm: Ontos Verlag, 2006, 175 – 219

Floridi, Luciano: Two Approaches to the Philosophy of Information. Minds and Machines, 13 2003, Nr. 4, 459 – 469

Frege, Gottlob; Patzig, Günther (Hrsg.): Funktion, Begriff, Bedeutung. Band 1144, Kleine Vandenhoeck-Reihe. Göttingen: Vandenhoeck & Ruprecht, 1975, 4., ergänzte Auflage

Görnitz, Thomas: Quanten sind anders. München: Spektrum Akademischer Verlag, 2006, 1. Auflage

Günther, Gotthard: Die klassische Metaphysik und das Problem der Kybernetik. In Das Bewußtsein der Maschinen. Baden-Baden und Krefeld: Agis-Verlag, 1963, 2. Edition

Günther, Gotthard: Die 'zweite' Maschine. In Das Bewußtsein der Maschinen. Baden-Baden und Krefeld: Agis-Verlag, 1963, 2. Edition

Hrachovec, Herbert: Gotthard Günthers Geltung, oder: die Grenzen der Geduld in Cybernetics. In **Pias, Klaus (Hrsg.):** Kybernetik. Band 2, Zürich-Berlin: Diaphanes, 2004, 163 – 195, Essays and Documents / Essays und Dokumente

Hörl, Erich: Das kybernetische Bild des Denkens. In **Hager, Michael. und Hörl, Erich (Hrsg.):** Die Transformation des Humanen. Band 1848, Frankfurt am Main: Surkamp Verlag, 2008, 163 – 195

Jammer, Max: Der Begriff der Masse in der Physik. Darmstadt: Wissenschaftliche Buchgesellschaft, 1974, 2. Auflage

Karagiannis, Dimitris; Endres, Albert. Krallmann, Herbert. und Schnupp Peter (Hrsg.): Wissensbasierte Datenbanken. Band 8.2, Handbuch der Informatik. München: R.Oldenburg Verlag, 1994

Kilian, Ulrich. und Weber, Christine (Hrsg.): Lexikon der Physik. Band 3, Heidelberg: Spektrum Akademischer Verlag, 1999, in sechs Bänden

Knuth, Donald Ervin: Fundamental Algorithms. Band 1, The Art of Computer Programming. Boston: Addison-Wesley, 1997, Third Edition

Krämer, Sybille: Symbolische Maschinen. Darmstadt: Wissenschaftliche Buchgesellschaft, 1988

Krämer, Sybille; Patzig, Günther. Scheibe, Erhard. Wieland Wolfgang (Hrsg.): Berechenbare Vernunft. Band 28, Quellen und Studien zur Philosophie. Berlin, New York: Walter de Gruyter, 1991

Krämer, Sybille: Was haben die Medien, der Computer und die Realität miteinander zu tun? In **Krämer, Sybille (Hrsg.):** Medien Computer Realität. Band 1379, Frankfurt am Main: Surkamp Verlag, 1998

McCulloch, Warren S.: What Is a Number, that a Man May Know It, and a Man, that He May Know a Number? In Embodiments of Mind. Cambridge: The MIT Press, 1988

Mocigemba, Dennis: Die Ideengeschichte derComputernutzung. Dissertation Technischen Universität Berlin, Berlin, Deutschland, 2003

Mycka, Jerzy: Analog Computation and Church's Thesis. In **Adam Olszwski, Jan Woleǹski, Robert Janusz (Hrsg.):** Church's Thesis After 70 Years. Band 1, Heusenstamm: Ontos Verlag, 2006, 331 – 352

Neumann, John von (Hrsg.): Die Rechenmaschine und das Gehirn. München, Wien: R.Oldenburg Verlag, 1980, 4.Auflage

Ritter, Joachim. und Gründer, Karlfried (Hrsg.): Historisches Wörterbuch der Philosophie. Band 5, Basel: Schwabe & CO AG Verlag, 1980

Russell, Bertrand: Philosophie des Abendlandes. Wien: Europa Verlag, 2001, 10. Auflage

Schneider, Hans-Jochen (Hrsg.): Lexikon Informatik und Datenverarbeitung. München: R.Oldenburg Verlag, 1997, 4., aktualisierte und erweiterte Auflage

Searle, John: Is the Brain a Digital Computer? Cambridge: MIT Press, 1992

Shagrir, Oron: Why We View the Brain as A Computer. Synthese, 153 2006, Nr. 3, 393–416

Thompson, David L.: Intentionality and Causality in John Searle. Canadian Journal of Philosophy, 16 1986, Nr. 1, 83–97

Turing, A. M.: Computing Machinery and Intelligence. Mind, 59 1950, Nr. 236, 433–460

Turing, Alan M.: Systems of logic based on ordinals. Proceedings of the London Mathematical Society. Second Series, 45 1939, 161–228

Turing, Alan M.; Dotzler, Bernhard. Kittler, Friedrich (Hrsg.): Intelligence Service. Berlin: Birkmann & Bose, 1987, Deutsche Erstausgabe

Zusammenfassung

Diese Diplomarbeit ist eine Untersuchung über den ontologischen Status von Computersystemen. Es ist nicht so klar, wie ein Computer von anderen, nicht berechnenden Dingen zu unterschieden ist oder welche Eigenschaften etwas haben muss, damit wir es 'Computer' nennen. Ich gestehe mir zu, nicht katalogisch alles, was mit Computern in Verbindung gebracht werden kann, zu erfassen, sondern greife einige Aspekte der gestellten Was-Frage heraus, um sie genauer zu untersuchen. Das betrifft vor allem die Frage nach Materialität und physischer Gebundenheit von Computern, die Frage nach Berechenbarkeitsbegriffen und Formalisierung, weiters Fragen zur Analogie von Gehirn und Computer und daraus entstehenden Schwierigkeiten und Missinterpretationen und nicht zuletzt, Fragen zum Unterscheidung von Analog- und Digitalcomputern.

Lebenslauf

Von 1993 bis 1998 besuchte ich die Höhere Technische Bundeslehranstalt für Automatisierungstechnik in Neufelden (i. M.). Nach der Matura am 26.Juni 1998 wurde ich im Oktober 1998 für zwölf Monate zum Zivildienst in einer Wohneinrichtung der Lebenshilfe Oberösterreich in Linz einberufen. Anschließend absolvierte ich im Wintersemester 1999 das erste Semester der Pädagogische Akademie in Linz und fasste danach den Entschluss nach Wien zu gehen. Seit dem Sommersemester 2000 studiere Philosophie und Germanistik an der Universität Wien. Ab dem Sommersemester 2001 war ich für zwei Semester als Mitbeleger für den Studienzweig Informatik an der TU-Wien eingeschrieben. Vom Wintersemester 2000 bis zum Wintersemsester 2001 war ich als Tutor und vom Sommersemester 2002 bis zum Sommersemester 2004 als Studienassistent am Institut für Philsosophie beschäftigt. Seit dem 03.Juni 2004 bin ich im Rahmen des allgemeinen Universitätspersonals als EDV-Techniker – unter anderem ebenfalls am Institut für Philosophie – tätig.