

Dieter Mersch: *Pro-Grammata*. Einige Überlegungen zu einer Theorie der Programme.

Brechts Intervention

In seinen Reden zum Rundfunk bezeichnete Bertolt Brecht diesen „kolossalen Triumph der Technik“ zugleich als eine „vorsintflutliche“ Angelegenheit, weil die gesendeten Programme lediglich „Wiener Walzer und ein Küchenrezept endlich der ganzen Welt zugänglich“ machten: „Ich wünsche sehr, dass diese Bourgeoisie zu ihrer Erfindung des Radios noch eine weitere Erfindung machte: eine, die es ermöglicht, das durchs Radio Mittelbare auch noch für alle Zeiten zu fixieren. Nachkommende Geschlechter hätten dann die Gelegenheit, staunend zu sehen, wie hier eine Kaste dadurch, dass sie es ermöglichte, das, was sie zu sagen hatte, dem ganzen Erdball zu sagen, es zugleich dem Erdball ermöglichte, zu sehen, dass sie nichts zu sagen hatte.“¹ Ist inzwischen *diese* Erfindung auf verschiedenste Weise durch Magnetaufzeichnung, *Tape-recording* und digitaler Speicherung realisiert worden und wird von ihr massenhaft Gebrauch gemacht, bestätigt sie nur noch umso mehr Brechts Diagnose, weil die Archive der Nichtigkeiten überquellen und eine beeindruckende Monstrosität der Leere demonstrieren, die vielleicht nur dadurch abgemildert wird, das auf ihre Wiederholung weitgehend verzichtet wird.

Allerdings erkannte Brecht im Rundfunk noch ein *zweites*, systematischeres Problem, das, nach seiner optimistischen Analyse, dem ersten insoweit korrespondiert, als seine Lösung, so die Hoffnung, dem nutzlosen Überfluss ein jähes Ende bereiten würde: Denn das Radio, als Verbreitungsmedium, besitze „seiner Natur nach“ nur „eine Seite“, wo es „zwei haben müsste“: Es sei „ein reiner Distributionsapparat, (es) teilt lediglich zu.“² Brecht setzte dem einen „demokratischen Zugang“ entgegen, den er als eine „Demokratisierung von unten“ verstand, die den Rundfunk für eine Öffentlichkeit, die ihren Namen verdiente, allererst funktionabel machen sollte – eine Utopie, die Hans-Magnus Enzensberger in seinem *Baukasten zu einer Theorie der Medien* von 1970 unter kybernetischen Prämissen wieder aufnahm und deren Phantasmagorien bis heute nicht mehr verstummt sind.³ Danach wäre der Distributionsapparat erst in einen „wirklichen“ Kommunikationsapparat zu verwandeln; er „wäre der denkbar großartigste Kommunikationsapparat des öffentlichen Lebens, ein ungeheures Kanalsystem, das heißt, er wäre es, wenn er es verstünde, nicht nur auszusenden, sondern auch zu empfangen, also den Zuhörer nicht nur hören, sondern auch sprechen zu machen und ihn nicht zu isolieren, sondern ihn in Beziehung zu setzen.“⁴

Auch dieser technikutopische ‚Fortschritt‘ scheint heute realisiert: Was unter den Stichworten der ‚Interaktivität‘, des *Web 2.0* oder der ‚social media‘ diskutiert wird und den Stachel der

¹ Bertolt Brecht: Radiotheorie, in: Günter Helmes, Werner Köster (Hg.): Texte zur Medientheorie, Stuttgart 2002, S. 148-154, hier S. 149f.

² Ebenda, S. 152.

³ Vgl. Hans-Magnus Enzensberger: Baukasten zu einer Theorie der Medien, München 1997 sowie die Replik von Jean Baudrillard, Requiem für die Medien, in: Koolhaas, Berlin 1978, S. 83-118 sowie deren Diskussion in Dieter Mersch: Medientheorie zur Einführung, Hamburg 2006, S. 71-79.

⁴ Brecht, Radiotheorie, a.a.O., S. 150, 152 passim.

politischen Debatten um mediale Emanzipation ausmacht, war auf diese Weise ebenso vorweggenommen wie die Auflösung der statischen Programmsysteme in dynamische Netze, die ihre individuelle Kollektion wie wechselseitige Partizipation zu erlauben schienen – und deren Schwundstufen im Internet nicht weniger zu besichtigen sind wie die narkotisierenden Unterhaltungen der Radiosendungen zu Brechts Zeiten zu hören waren. Vorerst schien jedoch die erträumte Umgestaltung der Programme an der Struktur des Medienmechanismus selbst zu scheitern: Rundfunk- und Fernsehsender gossen ihre Produkte lediglich unterschiedslos vor einer indifferenten Masse von Konsumenten aus, deren Gewohnheiten sie dressierten und deren vermeintliche Geschmäcker sie durch Befragung beharrlich auszuforschen trachteten; doch blieb, was immer man dabei herauszufinden glaubte, das Publikum grundsätzlich von der Apparatur entkoppelt. Die prinzipielle Beschränkung schlug auf die Programme ebenso zurück wie sie sich bis in die digitalen Medien, denen man seither die endgültige Realisation des Partizipatorischen bescheinigt, fortsetzen: Diese lassen wie jene nur *Wahlalternativen* zu und gestatten damit weder *Responsivität* noch *Reziprozität*. Sie sperren sich, genauso wie direkte Intervention, aufgrund der Nichtantizipierbarkeit und Unbestimmtheit von Alterität, gegen jede mögliche Programmierbarkeit.

Gleichwohl steigert die Digitalisierung die Komplexitäten und damit die Ambivalenzen, die ihrerseits die Rolle der Programme verwandeln. Denn die zunehmend unsichtbaren Netze erlauben nicht nur die Zirkulation eines überbordenden Rauschens von Desinformationen, vielmehr ermöglicht die Ubiquität mathematischer Programmierungen, auf denen sie fußen, auch *Gegenprogrammierungen*, die sich den Systemen der Macht prinzipiell zu widersetzen vermögen. Bezeugt die Organisation spontaner politischer Aktionen, Flashmops oder unkontrollierbarer Aufklärungskampagnen die Kraft der Umkehrung, gerade weil ihre Grundlage in der Performanz mathematischer Funktionen besteht, die sich politisch neutral verhalten, kann dadurch, im Grundsatz, jeder zum Programmierer und folglich Strategen einer anderen Gewalt werden, die sich zentralistischer Kontrollen entzieht. Dennoch darf nicht vergessen werden, dass sich, anders als die ‚Sendung‘ und ihr Prinzip egalitärer Streuung,⁵ deren Struktur an *formale Wahlen* bindet, wie sie technisch in Schaltungen zum Ausdruck kommen. Die These, die im folgenden auszuführen sein wird, lautet daher, dass die ubiquitären Programmierungen, wie sie durch den Triumph digitaler Medien virulent geworden sind, gleichzeitig eine *Ordnung von Kommunikationen* erkaufen, die an sich weder progressiv noch repressiv ist, *sondern eine Praxis oder Präferenz erzwingen*, die ausnahmslos einer *Logik von Entscheidungen* gehorcht. Innerhalb des digitalen Registers gibt es zu ihr keine Alternative. Die Notwendigkeit der Wahl lässt sozusagen *keine Wahl. Programme und ihre Programmatik kommen darin gleichsam zu sich*.

Begriffliche Annäherungen

Offenbar erweist sich als zentraler Punkt für die Rekonstruktion einer ‚Theorie‘ der Programme die *Logik von Entscheidung* oder *Schaltung*, wie sie den technischen Agenten implementiert sind. Sie macht aus ihnen *Modulare von Auswahlen*. Wir haben es dabei mit binären Anordnungen zu tun, die nicht nur die Programmierung der Geräte, sondern auch das betrifft, was sie an Inhalten produzieren und zur Verfügung stellen wie ebenfalls das, was übertragen, gespeichert und konstruiert werden kann. Gilt der Befund in erster Linie für

⁵ Vgl. John Durham Peters: *Speaking into the Air. A History of the Idea of Communication*, The University of Chicago Press 1999, bes. S. 33ff.

technische Medien, seien es Auszeichnungs-, Distributions- oder Kommunikationsmedien, wie sie sich seit etwa 1900 durchgesetzt haben, wäre ihnen gegenüber zunächst jedoch ein allgemeiner Programmbegriff zu erarbeiten, der ihn gleichzeitig vom Präjudiz der Wahl erlöst und diesen bestenfalls als Spezialfall ausweist. Doch bedeutet dies, in die Geschichte zurückzugehen das Programm derart in Beziehung zum Medialen zu rücken, dass zunächst einmal von einer *Differenz* zwischen deren logischen Entscheidungsprogrammen und Schaltern einerseits und den Vorschriften oder Ankündigungen andererseits auszugehen ist, als die sie, unabhängig von technischen Medien, in Erscheinung traten. Beide wären zu unterscheiden und der Programm-Begriff einem *allgemeineren* Verständnis zuzuführen, ohne ihn bereits medial oder technologisch zu überformen.⁶

Tatsächlich taucht die Rede von Programmen und Programmatiken unabhängig von ihrer medienspezifischen Manifestation Ende des 18. Jahrhunderts mit Blick auf öffentliche Annotationen und Bekanntmachungen auf. ‚Pro-grammata‘, das besagt auch die griechische Etymologie mit Bezug auf verwandte Ausdrücke wie *pro-graphein* oder *prographo*, bezeichnen im eigentlichen Sinne ‚Vor-Schreibungen‘ oder ‚Auf-Schriften‘ bzw. ‚Auf-Zeichnungen‘ mit der doppelten Konnotation einer Festsetzung und Anordnung, die, indem sie präskribieren zugleich auch antizipieren. Historisch bezeugbar werden solche Programme oder Programmatiken seit der Aufklärung in Gestalt mannigfacher Aufrufe oder schriftlicher Fixierungen, etwa von Grundsätzen, Tagesordnungen oder Zielen, um standardisierbare Abläufe ebenso zu regeln wie verfügbar zu machen, wobei das Präfix ‚pro-‘, das gleichermaßen im Lateinischen existiert, sowohl eine räumliche als auch zeitliche Ausrichtung impliziert, die die Programme an Zukünfte anschließt. Im wörtlichen Sinne bezeichnen sie folglich öffentliche Ankündigungen, die in Form tabellarischer Listungen, die sie gleichzeitig verräumlichen, künftige Ereignisse schematisieren und damit wählbar machen. Man versteht, warum Programmatiken mit Plänen oder der Formulierung prospektiver Leitvorstellungen assoziiert sind: Sie strukturieren vor und wirken dadurch normierend.

Joachim Paech hat zudem die wachsenden Regime der Programme und Programmatiken seit dem 19. Jahrhundert mit der aufkommenden Moderne in Verbindung gebracht.⁷ Ihre Vorherrschaft spiegle deren Willen zur Rationalisierung. Dazu gehört gleichfalls die bekannte Selbstbeschreibung der Moderne als ‚Vorhut‘, ‚Spitze‘ oder *Avantgarde*. Geradezu ubiquitär wird ihr Gebrauch darum im 20. Jahrhundert – von den Revolutionen der Künste mit ihrer Flut von Manifesten und Proklamationen bis zu den zahllosen politischen Programmen und Parteistatuten samt ihrer nicht enden wollenden ‚Zukunftspakte‘. Hinzuzurechnen wären außerdem die technischen Programmierungen von Charles Babbage *Difference Engine* über die Kybernetik bis zur Turingmaschine und den genetischen Programmen und Selbstprogrammierungen des Menschen. Augenscheinlich haben wir es nunmehr mit einer *universalen Kategorie* zu tun, einem Passepartout, das mal politisch, mal ästhetisch oder technologisch ausgelegt wird, worin sich vor allem jedoch jene *hegemoniale Macht des*

⁶ Vgl. dazu Knut Hickethier: Das Programm – Schlüsselbegriff der Medienwissenschaft. Zur Programmgeschichtsforschung der Rundfunkmedien, in: Medien und Zeit, Heft 2 (2008), S. 13-21, wonach Medien auf Strukturen der Übertragung und Distribution verweisen, Programme jedoch „auf die Angebotsseite“. Vgl. auch seinen Beitrag in diesem Band.

⁷ Joachim Paech: Das Programm der Moderne und dessen postmoderne Auflösungen: Vom Werk zum Text zu Multimedia, in: ders., Andreas Schreitmüller, Albrecht Ziemer (Hg), Strukturwandel medialer Programme. Vom Fernsehen zu Multimedia, Konstanz 1999, S. 13-30. Ebenso Vorwort zu diesem Band.

Programmierbaren, die Vorstellung genereller Machbarkeiten und Verfügbarkeiten ausdrückt, wie sie seit 1800 zur Ära des Ingenieurs gehört. *Ihre Totalisierung ist Signum einer Hegemonie technologischer Kultur*. Wenn es daher scheint, dass die klassischen Programme und ihre Ordnungsfunktion im Zuge der Postmoderne ihre Bedeutung eingebüßt hätten, weil sie sich individualisieren und vervielfältigen, dann trifft dies nur eingeschränkt zu, nämlich dort, wo sie im traditionellen Sinne Verhalten präformieren oder autoritativ steuern. Das gilt besonders für solche Medienangebote, die nicht länger einseitig ihren Konsum determinieren, sondern sich mehr und mehr in komplette Portfolios verwandeln, die sich jederzeit den einzelnen Wünschen fügen – wir sind dann offenbar mit einem Exzess von Programmierungen konfrontiert, die ihren Charakter als Programmierungen maskieren, weil sie im Hintergrund laufen und die technische Möglichkeiten verwirklichen, d.h. ebenso wohl die Ubiquität der Wahl regeln als auch längst in die Strukturen des Denkens eingegangen sind, um zum technologischen Schlüssel kultureller Praktiken selbst zu avancieren. Eher müsste man deshalb sagen, dass eine bestimmte Form von ‚Programmkultur‘ untergegangen ist, um einer anderen, totalisierten und vor allem mathematischen Platz zu machen, die, wie noch zu zeigen sein wird, einen ganz eigenen Charakter besitzt, aber die Struktur des Programmatischen allererst entschält. Seither treten die Programme in Gestalt von Metaprogrammen oder sich selbst generierender ‚Programm-Programmen‘ auf, deren latente ‚Voraus-Schreibungen‘ sich unauffällig, wie unsichtbare Implantate in den Körper des Sozialen inskribiert haben und ihre Form von Kommunikation determinieren.

Pro-Gramme als Dispositive

Ist es sinnvoll, in allen diesen verschiedenen Hinsichten noch dasselbe Wort zu benutzen und gleichermaßen von ‚Programmen‘, ‚Programmatiken‘ und ‚Programmierungen‘ zu sprechen? Die verschiedenen Untersuchungen der vorliegenden *Programm(e) der Medien* geben darauf keine eindeutige Antwort. Vielmehr arbeiten sie das Thema im engeren Sinne von den Programmstrukturen in Medien und ihren Nutzungsstrategien über die ‚Grenzen‘ des Programmierbaren bis zu den Programmatiken einer künftigen Medienwissenschaft aus, um in unterschiedlichen Lektüren die mannigfachen Faltungen und Fassetten des Begriffs sichtbar zu machen und damit eine Diagnose der Gegenwart abzugeben. Gibt es dennoch zwischen ihnen ein Gemeinsames, das rechtfertigt, zumindest in Grundzügen die Frage nach einer ‚Theorie‘ des Programms aufzuwerfen? Vielleicht ist die Frage ebenso sinnlos oder falsch gestellt, genau wie die Frage nach der „Familienähnlichkeit“ zwischen den Sprachspielen, die Wittgenstein sarkastisch mit dem Hinweis quittierte, „(e)benso könnte man sagen: es läuft ein Etwas durch den ganzen Faden, – nämlich das lückenlose Übergreifen dieser Fasern.“⁸ Dementsprechend erweist sich der Begriff des ‚Programms‘ – analog dem des Spiels oder der Sprache – als in sich ungeschlossen; es handelt sich, wie sich ebenfalls mit Wittgenstein sagen lässt, um einen Begriff „mit verschwommenen Rändern“.⁹ Dabei legen sämtliche Beiträge des vorliegenden Bandes nahe, von Programmen weniger im Sinne von analysierbaren *Objekten*, denn in der Bedeutung von *Prozessen* zu reden, die wiederum spezifische *Dispositive* erfordern, die sie als solche terminieren. Zudem behandeln sie die Programme als *medienübergreifende Kategorien*, nicht medienspezifisch. Dabei lässt sich – um einen hypothetischen Anfang zu nehmen – ihre allgemeinste Grundlage bereits vom

⁸ Ludwig Wittgenstein: *Philosophische Untersuchungen*, Frankfurt/M. 1971, § 67; ferner §§ 66ff.

⁹ Ebenda, § 71.

Wort her in der *Schrift* – oder genauer: einer gewissen Art von ‚Schriftbildlichkeit‘ – ausmachen.¹⁰ Als Textur eines Registers, als Ablaufschema oder Plan in Gestalt formaler Matrizen, die Einzelschritte gleichermaßen aufzählbar machen wie festlegen oder Anordnungen wiedergeben, die *operative Entscheidungen* (Wahlen) ermöglichen. Anordnungen oder Aufstellungen setzen ihrerseits noch ein Tableau als ‚Medium‘ voraus, das ihre Übersicht, ihre Lesbarkeit und Auswahl besorgt. Dann folgt, dass Programme, obzwar verschieden von Medien, ihrerseits auf medialen Strukturen fußen. Ihr Verbindendes ist – medienphilosophisch betrachtet – das *Diagramm* als Weise ihrer Visualisierung, sodass beide, *Programme* und *Diagramme*, auf eine implizite Weise aneinander gekoppelt scheinen: Letztere – zurückgehend auf das griechische Präfix ‚dia‘, lateinisch ‚per‘, im Deutschen ‚durch‘ – nennt im literalen Sinne eine ‚Durchschreibung‘, d.h. jene Tätigkeit, die *durch* eine Aufschreibung oder Zeichnung etwas sichtbar oder handhabbar macht und damit als eigentliche *mediale Praxis* fungiert; während das *Programm* im oben ausgeführten Sinne als ‚Vor-Geschriebenes‘ oder eine ‚Voraus-Schreibung‘ gilt, welche nicht selbst schon als Medium funktioniert, sondern als dessen Inhalt. Es kennzeichnet damit dasjenige, was, in der Bedeutung eines Präskripts, ein anderes, das sich in der Schrift oder als Textur manifestiert, so konditioniert bzw. seine Performanz *aus-,richtet‘* oder reglementiert, dass *durch* seine Strukturalität Selektionen angeleitet werden oder Praktiken eine Richtung erhalten.¹¹ Programme basieren also auf Medien wie sie gleichermaßen ‚Mediationen‘ vorausgehen, sie anleiten oder ‚be-dingen‘. Sie können folglich als ‚meta-mediale‘ Konditionen aufgefasst werden: Sie initiieren oder steuern mediale Prozeduren, strukturieren und lenken sie als Möglichkeit, weshalb es kein Zufall ist, dass sie im Rahmen von Digitalisierung und Kybernetik eine universale Note gewinnen. Programme, selbst medial grundiert, gehen ihrerseits anderen Medien voraus, deren Praktiken, Potenziale und Grenzen sie bestimmen: Dennoch fußen nicht alle Medien auf Programmen; sie bilden keine *notwendigen* Voraussetzungen, keine Präsuppositionen, sondern gleichsam ‚Hilfslinien‘, insbesondere dort, wo, ab einer bestimmten Komplexitätsstufe, im technischen Sinne *Entscheidbarkeiten* verlangt werden. Die ‚meta-mediale‘ Funktion der Programme kommt insbesondere dort zum Tragen: Sie definieren die Entscheidungslogiken oder Algorithmen, durch die digitale Medien wiederum determiniert sind. Gleichwohl geht das Verhältnis von Medien und Programmen darin nicht vollständig auf: Vielmehr beruhen diese selbst noch auf medialen Formationen, wie sie sich z.B. in den Texturen der Tableaus oder Diagrammen niederschlagen, sodass davon auszugehen ist, dass sie ebenfalls mediale Eigenschaften besitzen, die ihrerseits Effekte auslösen, und zwar umso mehr, wie sie wiederum Entscheidungsräume bereitstellen, über die andere mediale Formen operieren. Programme – im Gegensatz zu Medien – operieren darum eher wie Regeln, die Angeboten gleichen, welche auch *nicht* oder *anders* befolgt werden können, wie gegenfinale Anwendungen, Subversionen oder Umprogrammierungen bekunden. Dennoch schränken sie

¹⁰ Vgl. Sybille Krämer: ‚Schriftbildlichkeit‘, oder: Über eine (fast) vergessene Dimension der Schrift, in: Horst Bredekamp, Sybille Krämer (Hg): *Bild, Schrift, Zahl*, München 2003, S. 157-176; Dies.: *Operationsraum Schrift. Ein Perspektivenwechsel im Schriftverständnis*, in: Gernot Grube, Werner Kogge, Sybille Krämer (Hg): *Schrift. Kulturtechnik zwischen Auge, Hand und Maschine*, München 2005, S. 13-32; Dies.: *Karten – Kartenlesen – Kartografie. Kulturtechnische Überlegungen*, in: Philine Helas et al. (Hg): *Bild / Geschichte. Festschrift für Horst Bredekamp*, Berlin 2007, S. 73-82; sowie ihr Beitrag in diesem Band.

¹¹ Zur Bedeutung des ‚Dia‘ – ‚Durch‘, ‚Per‘ – im Medialen vgl. meine Überlegungen in: Dieter Mersch: *Meta / Dia. Zwei unterschiedliche Zugänge zum Medialen*. Zeitschrift für Medien- und Kulturforschung, Bd. 2 (2010) Hamburg, S. 185-208.

die Alternativen ein und behaupten damit eine *bestimmte* Art der Befolgung, die ihrerseits keine Alternative lässt. Programme erweisen sich darin als latent *diktatorisch*: Wenn sie auch ontologisch der Sphäre der *potentia* angehören, die erst dadurch („dia’/per’) ihre Realität oder *actualis* gewinnen, dass ihnen Handlungen oder Materialitäten erst entsprechen, deren Ordnung sie ‚vorgeben’ oder imprägnieren, gestatten sie selbst doch stets nur wieder ein ‚Alles’ oder ‚Nichts’, ein ‚An’ oder ‚Aus’, d.h. die Rigorosität eines Entweder-oder, das sich mit der Wahl des Programms bereits entschieden hat. Keineswegs sind damit andere Umgangsformen schon ausgeschlossen: Der Zusammenhang lässt sich, statt an expliziten Programmvorgaben, paradigmatisch anhand von Architekturen oder Raumordnungen ablesen. Bauen bedeutet, wie Martin Heidegger gesagt hat, zuvor und zunächst ein „Räumen“:¹² Es „bildet“ Räume und Stätten des „Aufenthalts“ – *Topoi* oder Orte, die ihrerseits Lebensformen determinieren. In diesem Sinne handelt es sich um ‚Dispositive’, die ebenso eröffnen wie verschießen, ebenso ermöglichen wie engführen. Erst ihre dispositive Struktur gestattet eine besondere Weise des „Wohnens“, wie sie diese gleichermaßen abrichtet und erzwingt. Blickt man auf ihre diagrammatische Struktur, ihren Grundriss oder Bauplan, sieht man zugleich, auf welche Weise die so gestalteten Räume Wege und Bewegungsabläufe bereits konditionieren: In ihnen zeichnet sich eine Programmierung ab, die Verhalten gleichermaßen festlegt wie erzwingt. Dennoch lassen sie auch eigene Freiheitsgrade zu: Trotz aller intrinsischen Determination bleibt stets die Möglichkeit des konträren Gebrauchs, der Fehlbesetzung oder Umnutzung – das Internet, trotz der bizarren Banalität seines gewöhnlichen Gebrauchs, produziert immer wieder diese überraschenden, nicht zu antizipierenden Widerständigkeiten. Ähnliches gilt für Baupläne: Grundrisse ‚programmieren’; doch ist nicht *alles* an ihnen programmiert: An Blaupausen ist nicht schon die Medialität ganzer Bauwerke oder Plätze ‚vorgezeichnet’, vielmehr müssen ihre Praktiken erst zum Leben erweckt und verwirklicht werden, auch wenn dies nur innerhalb ihres Rasters vollzogen werden kann. Ihr architektonisches ‚Programm’ ist, gemessen an der Diagrammatik des Plans, ihre Präfiguration als Potenz; es legt den *Rahmen* fest, bestimmt die Möglichkeiten, doch bedeutet es noch lange keine ‚Bewohnung’, wohingegen das Mediale, das selbst ohne Ort und Bestimmung bleibt, allererst den Praktiken entstammt, *durch* („dia’/per’) die diese sich ‚mediiert’.

Anordnung von Raum und Zeit: Einige Strukturmerkmale

In diesem Sinne verweisen Programme vor allem auf Verfahrensordnungen: Sie sind in erster Linie prozedural terminiert; sie geben den medialen Prozessen eine gleichermaßen temporale wie räumliche Struktur vor. Das lässt sich bereits anhand ihrer einfachsten Form, einer Serie von Programmpunkten, deutlich machen: Ihre Aufstellung in Gestalt einer Liste oder einer Tafel organisiert eine Sukzession, die auf diese Weise Raum und Zeit als Möglichkeitsbedingung von Praxis und Entscheidung präfiguriert. Tatsächlich ist die elementare Gestalt des Programms die Aufzählung als Folge von Ordinalzahlen. Sie gehorcht der Logik der Syntax. Deshalb gründen Programme und Programmatiken in arithmetischen Reihen: Ihre Ordnung setzt eine Diskretierung voraus, die ihrerseits Handlungen in Verfahrensschritte einteilt und damit klassifizierbar macht. In diesem Sinne formatieren Programmatiken Prozesse, einzig zu dem Zweck, ihnen ein Schema oder eine Grammatik

¹² Vgl. Martin Heidegger: Bauen Wohnen, Denken, in: ders.: Vorträge und Aufsätze, Pfullingen 4. Aufl. 1978, S. 139-156, insb. S. 148ff.; ders.: Die Kunst und der Raum, St. Gallen, 2. Aufl. 1983, S. 8ff.

aufzuerlegen. Gleichzeitig erlauben sie ebenso sehr eine *Synopsis* wie ihre ‚formale‘ Prozeduralisierung, ihre schrittweise Abarbeitung – wie sie etwas bei der quälenden Abhandlung von ‚Geschäftsordnungen‘ auftreten. Zwar schaffen, analog diagrammatischer Inskriptionen, auch Karten, Listen, Register oder Tabellen ähnliche „Operationsräume“ (Krämer), die Verknüpfungen, Zuordnungen, Zählungen, Montagen und dergleichen gestatten, indem sie ‚Über-Sichten‘ generieren, in denen sich ‚denkend‘ hin- und herbewegen lässt, doch verräumlichen ‚programmatische‘ Anordnungen auf eine Weise die Zeit, dass sie Abläufe wie Wahlen sequenzialisieren und folglich handhabbar und manipulierbar machen. Diagramme schließen neue ‚Kon-Texte‘ auf, geben Muster zu erkennen, ermöglichen ein produktives Spiel von Relationen, während Programme linear operieren. Wir haben es folglich weniger mit einer spezifischen Form von Kreativität zu tun, als vielmehr mit einer *Rationalisierung oder Ritualisierung von Prozessen* durch ihre ‚Präskription‘, die die Ereignisse vorstrukturieren, um sie *dadurch* (‚dia‘/‚per‘) beherrschbar zu machen. Programme bilden also Instrumente einer Herrschaftlichkeit, die in Bezug auf die Ordnung der Zeit einen disziplinierenden Charakter besitzen. Sie zerlegen komplexe Vorgänge in Einzelschritte und normieren sie dadurch. Bildet, wie Sybille Krämer mit Bezug auf das Diagrammatische immer wieder betont hat, die *Notation* eine maßgebende Grundbedingung, deren formaler Graphismus gleichermaßen *sichtbar* macht wie sie *kognitive Prozesse* anleitet und allererst ermöglicht – denn nicht vergessen werden darf, dass Diagramme vor allem Medien des Denkens sind –, manifestieren sich demgegenüber im ‚Programmatischen‘ die Regime eines *Normativismus*. Sie sind im Wesentlichen dezisionistischer Natur. Zwar unterliegen Programme, wie bereits ausgeführt, selbst noch einer bestimmten Form von ‚Aufzeichnung‘ oder Diagrammatizität – auch die technischen oder mathematischen bedürfen noch der Turingtafel oder des Fließschemas –, doch handelt es sich gerade *nicht* um die *Eröffnung* von Operabilitäten, sondern im Gegenteil um deren *Einschränkung* oder *Zurichtung* durch Imprägnierung von Schrittfolgen, Teleologien oder Wahlalternativen, die ihnen ihre eigene Orientierung verleihen. In einem *positiven* Sinne – in der Bedeutung eines Produktionsprinzips – gehören deshalb Programme der *causa finalis* an: Sie tendieren nicht unweigerlich zu einer logischen Strukturierung, vielmehr zu einer solchen, die Kontingenz vermeidet oder einschränkt, indem sie den Tatsachen Ziele oder Richtungen vorgeben und damit eine Ordnung auferlegen. Kurz: Programme und Programmierungen sind der Ausdruck einer antizipatorischen Vernunft. Sie flüchten vor der Unregierbarkeit der Zukunft und kompensieren sie, indem sie mögliche Entscheidungsräume kontrollierbar machen. Das lässt sich erneut anhand von Blaupausen als architektonische Äquivalente demonstrieren. Es ist klar, dass diese, wie Diagramme, etwas rastern, das zwar nicht notwendig realisiert werden muss, wohl aber *nicht ohne sie nicht* realisierbar wäre. Wie auch immer das Verhältnis zwischen Grundriss und Realisation ausfällt, dieser gibt den Raum vor, worin sich die Praktiken des Gebrauchs immer schon abgezeichnet und dressiert haben werden, *bevor* die Frage nach alternativen Nutzungen gestellt werden kann. Und wie diese auch immer sich im einzelnen zu artikulieren vermögen, sie können nicht umhin, sich *innerhalb* ihres präformierten Gehäuses zu erfinden und vorgegebenen Strukturen und Ressourcen zu folgen, um ihnen in dem Maße einen Widerstand entgegenzusetzen wie dieser ihnen selbst widersteht. Wir hatten ähnliches bereits gleich zu Anfang mit Bezug auf das Verhältnis zwischen Digitalisierung und Kommunikation festgestellt: Diese wird durch jene programmiert, so jedoch, dass wir es mit einer bestimmten, nämlich entscheidungslogischen

Form von Kommunikation zu tun bekommen – Programmierungen formatieren; sie erzwingen Ordnungen und damit das, was Roland Barthes ebenfalls in Ansehung der Struktur der Sprache bemerkt hat: Sie gibt uns ihre Weise des Sagens auf und entfremdet damit von möglichen Alternativen, weil sie sich immer schon entschieden hat, lediglich *eine einzige* Wahl – und nichts anderes – zu haben.¹³ Dasselbe findet sich ebenfalls bei Bruno Latour hinsichtlich der Eigengesetzlichkeit der Dinge: Unsere Interventionen enden dort, wie ebenso unsere Strategien der Dekontextuierung und Subversion, wo ihre Materialität sich dem Willen unseren Konstruktionen widersetzt¹⁴ – im gleichen Maße endet auch die Freiheit unserer Entscheidungen, wo die antizipatorische Vernunft ihnen bereits einen Rahmen ‚vorsetzt‘ und sie ‚programmiert‘. Alle Programmatiken basieren auf solchen Einschlüssen und Ausschlüssen, die den Zufall, das Chaos oder die Nichtordnung zu vermeiden suchen, um die Unberechenbarkeit der Zukunft einzuschränken; mit allen Programmen formuliert sich damit zugleich ein Machtanspruch, der, indem diese die Alternativen reduzieren, gleichzeitig jede Wahl zwischen Programm und Nicht-Programm vereitelt. Programme induzieren Entscheidungen – aber sie erscheinen *als* Programme nicht selbst wählbar. Vor allem künstlerische Programmatiken haben diesen Rigorismus vorgeführt: In ihnen manifestiert sich die souveräne Geste eines Neuanfangs, einer anderen Form von Kunst, die, noch bevor sie existiert, bereits ihre Ankunft feiert. Sie verfahren darum in einem provokanten Sinne *prospektiv* und schaffen so, was sie noch nicht sein können. Es ist kein Zufall, dass sich die Avantgarden, die sich ohnehin an die ‚Speerspitze‘ der Zeiten zu stellen wagten, ‚voranmarschierten‘, um im gleichen Maße ihre künftige Vormachtstellung zu sichern wie auszubauen. Wir müssen schneller rennen als die Kunst, heißt es sinngemäß bei Jean Cocteau, dann hat es den Anschein, als liefe sie uns hinterher.

Zur Differenz von Programmatik und Grammatik

Beruhend Programme derart auf ‚Präskriptionen‘, deren Strukturen jene Entscheidungsräume erst formatieren, woran sich ihre Zukünfte ausrichten sollen, wäre allerdings noch ihre spezifische ‚Grammatizität‘ zu klären. Wie zuvor schon zwischen ‚Programm‘ und ‚Diagramm‘ unterschieden wurde, gilt eine analoge Differenz genauso für die Beziehung zwischen ‚Programmatik‘ und ‚Grammatik‘. In der Tat ‚syntaktisieren‘ Programme Wahlalternativen, wie der Elementarfall, die Liste oder Aufzählung, demonstriert, die ja nichts anderes als eine Sequenzialisierung vornimmt. Dann liegt ihnen ebenfalls eine rudimentäre Grammatik zugrunde, die freilich nicht Aussagen oder Bedeutungen generiert, sondern lediglich *Selektionen* regelt. Doch gilt das Umgekehrte nicht: Grammatiken lassen sich nur in Ausnahmefällen als Programme verstehen; sie programmieren nicht die Sprache, sondern stellen flexible Regelwerke dar, die sich bestenfalls *rekonstruktiv* und mit einer Reihe von *Exklusionen* systematisieren lassen.¹⁵ Die Extension des Grammatik-Begriffs reicht daher weiter als die des Programm-Begriffs, dennoch ist für unsere Überlegungen nicht uninteressant, dass wiederum formale Grammatiken, wie sie der Theorie der Semi-Thue-

¹³ Roland Barthes: Lektion/Leçon: Antrittsvorlesung im Collège des France. Französisch und Deutsch, Frankfurt/M. 1980, S. 17-21.

¹⁴ Vgl. insb. Bruno Latour: Der Berliner Schlüssel: Erkundungen eines Liebhabers der Wissenschaften, Berlin 1996; ders.: Das Parlament der Dinge. Für eine politische Ökologie, Frankfurt/M. 2006.

¹⁵ Dies ist die Crux formaler Grammatik-Theorien in der Linguistik, insbesondere der rekonstruktiv verfahrenen Syntax-Theorie von Chomsky. Sie verdankt ihre Herkunft explizit mathematischen Modellen formaler Sprache; vgl. bes. Noam Chomsky: Syntactic Structur, The Hague 1957.

Systeme oder der ‚formalen Sprachen‘ der Mathematik zugrunde liegen, nichts anderes als ‚Programm-Sprachen‘ bilden, in denen Programm und „Erzeugendensystem“ zusammenfallen. Ihre Minimalbedingungen sind ein „Alphabet“ – man könnte sie mit den Objekten der Programm-Tableaus parallelisieren – und ein Set von „Produktionsregeln“, die aus Verknüpfungs- und Transformationsregeln bestehen und deren Reihenfolge in der Anwendung zunächst nicht festliegen.¹⁶ Ihre Grundlage bilden Funktionen, die sich ihrerseits einer Matrix zuordnen lassen, welche ihr ‚Produkt‘ vollständig definiert. Dabei dienen Funktionen in der Mathematik gewöhnlich der Darstellung von raum-zeitlichen Relationen, ebenso wie das Koordinatensystem, das ihre Verläufe ‚abbildet‘, ihre Zuordnung verräumlicht. Durch fortgesetzte Rekursion entstehen daraus Schemata, die alle Eigenschaften von Programmen erfüllen, die aber auf diese Weise eine Formalisierung erfahren und damit noch von einer anderen Seite einen Hinweis darauf zu geben vermögen, *was diese sind* bzw. *sein können*.

Man muss jedoch den gewöhnlichen Begriff des Programms vom mathematischen trennen, der gleichsam dessen radikalisierte Form darbietet – die äußerste Spitze seiner logischen Repräsentation, die zugleich ein spezifisches Cluster auszeichnet. Man kann also keine generelle Theorie des ‚Programms‘ anhand mathematischer Sätze aufstellen – sowenig wie es eine Mathematik der Grammatik gibt –, wohl aber lassen sich bestimmte Bedingungen präzisieren und ihre Eigenschaften diskutieren. Denn im allgemeinen sind Programme *nicht* logisch geordnet; sie gehorchen keiner formalen Struktur, keiner Gesetzmäßigkeit, allerdings bilden die logischen mit ihrem Rigorismus eine Spezial- oder Steigerungsform, die zunehmend beginnt, für diese als ‚Ideal‘ oder Modell zu fungieren. Deshalb erscheint ein Vergleich zwischen Programmatik und Grammatik fruchtbar: als formalisierbares Schema, dessen Fundament der Funktionsbegriff ist, avanciert im Mathematischen das Programm zu einer abstrakten Grammatik, die – gemäß der Prinzipien von Eingabe und Ausgabe – die Programmierung wie die Spielregeln eines Spiels determiniert: Wenn auch, sowenig wie ein allgemeiner Begriff des Programms, ein allgemeiner Begriff des Spiels existiert – erinnert sei an den Wittgensteinschen Ausdruck der „Familienähnlichkeit“ – gleicht die Regel der ‚Vollstreckung‘ einer Norm, die Spielzüge ‚programmiert‘: Devianz, wie Wittgenstein ebenfalls zu verstehen gegeben hat, ebenso wie Regelbruch oder Variation auch nur eines marginalen Teils der Definition heißt hier bereits, ein anderes Spiel zu spielen.¹⁷ Im selben Maße erweisen sich mathematische Programme, wie man sagen könnte, als ‚strikt‘: Sie präntendieren keine Möglichkeiten, sondern operieren *apodiktisch*. Wo gewöhnliche Programme oder Programmatiken in Gestalt von losen Annotationen oder Schrittfolgen ihre Performanzen lediglich vorstrukturieren, ohne zu ihnen zu nötigen, erzeugen die mathematischen ein logisches Gerüst, gleichsam eine ‚Algebra‘, deren Ordnung einem Verhängnis gleicht, dessen Analyse zuletzt der gleichen formalen Methoden bedarf wie diejenige, die sie selbst in Anschlag bringt.¹⁸ Nicht das Antizipatorische spielt hier eine Rolle,

¹⁶ Vgl. Theorien Formaler Sprachen, exemplarisch: Günter Holz, Hermann Walter: Automatentheorie und formale Sprachen, Mannheim 1968, Heinrich Becker, Hermann Walter, Formale Sprachen, Braunschweig 1977.

¹⁷ Vgl. Ludwig Wittgenstein: Philosophische Grammatik, Frankfurt/M 1973, §§ 45ff.; ders.: Philosophische Untersuchungen, Frankfurt/M. 1971, §§ 82ff., 199ff.; ferner Saul A. Kripke: Wittgenstein über Regeln und Privatsprache, Frankfurt/M. 1987.

¹⁸ Vgl. dazu auch meine Überlegungen in Dieter Mersch: Die Geburt der Mathematik aus der Struktur der Schrift, in: Gernot Grube, Werner Kogge, Sybille Krämer (Hg): Schrift. Kulturtechnik zwischen Auge, Hand und Maschine, München 2005, S. 211-233.

der Ausschluss von Kontingenz, sondern die Generierung einer Struktur, deren Zeitlichkeit freilich ohne jede Relevanz bleibt.

Anders gewendet: Der Begriff des Programms fällt hier mit dem des *Algorithmus* zusammen, und während wir dort mit *Prospekten* konfrontiert sind, bekommen wir es hier mit *formalisierten Verfahren* zu tun. Dennoch kann deren Untersuchung hilfreich sein, um, so die Vermutung, an ihnen einige Strukturmerkmale zu erhellen, die beide teilen. Die Mathematik der Programme bildet dann den Prüfstein, um gemeinsame Schnittlinien aufzusuchen, die schließlich zwischen der ‚Programmatur‘ komplexer Prozesse und algebraischen oder arithmetischen Ordnungen tiefe Verwandtschaften aufdecken.

Arithmetik der Programme: Einige mathematische Definitionen

Allgemein gilt damit, dass Programme in der Regel auf der Selektion und Zerlegung von Prozessen basieren, um sie operabel zu machen. Sie ‚diskretieren‘ Abläufe. Die trifft ebenso für mathematische Programme zu wie im besonderen Maße für organisatorische Komplexe, die – via *Komplexreduktion* – schrittweise auf kleinere Einheiten zurückgeführt werden. Man kann in diesem Sinne von einer *impliziten Arithmetisierung* sprechen, einer Separation oder ‚Aufteilung‘, deren bevorzugte mediale Aufzeichnungsform das Tableau oder die Matrix darstellt, die ihre zumindest partielle Mathematisierung nahelegt. Programmatur und Mathematik konvergieren genau in diesem Punkt: Jeder Programmierung geht – wie der Modellierung eines mathematischen Problems – die Konzeptualisierung des ‚Materials‘ voraus, die Räume, Zeiten, Ideen oder Relationen und Handlungen allererst regierbar macht. Sie legt sie zugleich in ihren Möglichkeiten und Grenzen fest. Je nachdem, *wie* das Material, die Vorstellungen, die Organisationsformen, die praktischen Vollzüge oder Prozesse und was auch immer eingeteilt und angeordnet wird, ergibt sich anderes, zum Teil auch unbedachte Wirkungen, Widerstände oder Synergien, weil sie nie ganz zu überblicken sind und ihre Emergenzen ins Exponentielle wachsen, je komplexer die Strukturen werden. Man kennt dies vor allem bei Computerprogrammen, die modular aufgebaut sind, zuweilen sogar durch Subprogramme geniert werden und jede angemessene Transparenz verweigern. Sie erzeugen ihre eigene Opazität mit Konsequenzen, die niemand vorausszusehen, geschweige denn zu berechnen vermag. Computerprogramme haben damit heute einen Komplexitätsgrad erreicht, der sie undurchdringlich macht: Sie sperren sich jeder Intelligibilität selbst für ihre Programmierer. Gleich Systemen, die sich von ihren Urhebern entkoppelt haben, führen Fehlläufe zur Revision kompletter Bauteile, weil die Fehlersuche ineffektiv und im einzelnen nicht nachvollziehbar wäre. Ihre spezifische Magie rührt daher: Ein Automat im wörtlichen Sinne des *automaton*, dem allein mit Glauben und Unterwerfung, nicht mit logischer Analyse zu begegnen ist.

In der Tat scheint es, als bewohnten die Programme inzwischen ihre eigene, vom Menschen getrennte Welten; doch wird es im Mathematischen im Gegensatz zu ästhetischen, sozialen oder politischen Programmen immer um *universelle Programmierbarkeiten* gehen, deren Effekte zuletzt vollständig zu beschreiben und damit zu kontrollieren sind, auch wenn die tatsächliche Kontrolle versagt. Entsprechend finden sie ihr Vorbild in der *Theorie der Turingmaschine*, die im besonderen die Grundlage der Computerisierung und vor allem der Informatik bildet, auch wenn es sich zunächst um nicht mehr als ein abstraktes

mathematisches Modell handelt.¹⁹ Programme, die einst noch aus Ankündigungen, ‚Vor-Schriften‘ oder Strukturen bestanden, beginnen sich hier gleichsam technisch zu überformen, zu verallgemeinern und sich ubiquitär auszubreiten, um rückhaltlos von dem Besitz zu ergreifen, was sich mathematisch erfassen lässt. Dabei lassen sich Computerprogramme als syntaktische Regelwerke verstehen, die homolog zu ‚formalen Sprachen‘ und Semi-Thue-Systeme funktionieren indem sie wie diese auf eine Reihe von Anfangsbedingungen und Instruktionen basieren, deren Zweck allein in der Lösung spezifischer Aufgaben oder Funktionen besteht. Anders als der klassische Programmbegriff handelt es sich also um Serien von Befehlsketten, die gemäß der Churchschen These dem intuitiven Begriff der *Rechnung* bzw. des *Rechenschemas* entsprechen und in nichts anderem als Formeln bestehen, die ‚Algorithmen‘ definieren.²⁰ Wechselseitig erscheinen dann die Ausdrücke ‚Algorithmus‘ und ‚Programm‘ austauschbar – doch bleibt der Gemeinplatz, die Grundlage der Computerisierung bilde überhaupt der Algorithmus, solange trivial, wie dessen Begriff nicht präzisiert wird. Sucht man indessen nach einer exakten Definition, können in einer ersten Annäherung Rechenschemata als Verknüpfung von ‚Vor-schriften‘ (*programmata*) aufgefasst werden, deren Kern rekursive Funktionen f_i darstellen, die über *eine* Grundmenge bzw. *Alphabet* A operieren, sodass jeder Algorithmus durch eine – arithmetisch bezifferbare – Folge f_1, \dots, f_n vollständig erklärt werden kann.²¹ Sie stiften Übergänge $G \rightarrow H$ zwischen zwei Maschinenzuständen, mit denen wiederum die mathematischen Begriffe der ‚Operation‘ bzw. der ‚Produktionsregel‘ (‚Erzeugungsregel‘ / ‚Erzeugendenfunktion‘) korrespondieren, die aus Variationen von Anordnungen $a_1 \dots a_k \rightarrow b_1 \dots b_m \in A$ bestehen.²² Alle diese Begriffe, die im übrigen unabhängig voneinander in unterschiedlichen mathematischen Theorien wie der Beweistheorie, der formalen Algebra, der Spieltheorie etc. definiert worden sind, um nachträglich ihre Gleichheit zu erweisen, sind folglich ineinander überführbar: Programm, Algorithmus, Semi-Thue-System, rekursive Funktion sowie Produktionsregel, Operation und formale Syntax (Sprache). Gleichzeitig offenbart ihre Identität maßgebliche Strukturgemeinsamkeiten, die zwischen ihnen bestehen und die sich gegenseitig erhellen, sodass die Mathematik der Programme wiederum einen wichtigen Fundus für eine Theorie des ‚Programmatischen‘ überhaupt bietet – auch wenn sich nicht erwarten lässt, eine für alle Programme gleichermaßen gültige Theorie zu entwickeln.

Programme als Turingmaschinen

Kann so im speziellen der *mathematische* Begriff des Programms mit dem der Turingmaschine identifiziert werden, dürfen diese allerdings nicht mit *technischen* Computerprogrammen verwechselt werden, die zwar auf ihnen beruhen, aber grundsätzlich im Endlichen operieren, während Turingmaschinen Abstraktionen – „Papiermaschinen“, wie Turing sagte – bilden, die an den entscheidenden Stelle auf eine hypothetische Unendlichkeit rekurriert.²³ Weit davon entfernt, ‚wirklichen‘ Automaten darzustellen, erweisen sie sich

¹⁹ Hans Hermes: *Aufzählbarkeit Entscheidbarkeit Berechenbarkeit*, Berlin, Heidelberg, New York 2. Auflage 1971, S. 33ff.; Josef Weizenbaum: *Die Macht der Computer und die Ohnmacht der Vernunft*, Frankfurt/M. 1977, S. 107ff., 180ff.

²⁰ Hermes, *Aufzählbarkeit Entscheidbarkeit Berechenbarkeit*, a.a.O., S. 9ff; Herbert Merthens: *Moderne Sprache Mathematik*, Frankfurt/M. 1990, S. 300ff.

²¹ Sie auch: Mersch, *Die Geburt der Mathematik aus der Struktur der Schrift*, a.a.O.

²² Vgl. Walter Becker, *Formale Sprachen*, a.a.O., S. 73ff.

²³ Alan Turing: *Rechenmaschinen und Intelligenz*, in: ders., *Intelligence Service*, hg. von Bernhard Dotzler und Friedrich Kittler, Berlin 1987, S. 147-182.

vielmehr als imaginäre Konstruktionen, die die Berechnung einer mathematischen Aufgabe in endlich vielen Schritten auf einem unendlich langen Rechenband besorgen. Dabei basieren die Turingmaschinen – genauso wie Programme und formale Sprachen – auf einer Serie von *Instruktionen*,²⁴ deren Ausführung allerdings ausschließlich von ihrem syntaktischen Gebrauch abhängt, der zwar *Zeit* verbraucht, aber selbst ohne *Zeit* *ist*, denn kein Beweis eines mathematischen Theorems oder eines Rechenresultats hängt davon ab, in welcher *Zeit* sie erstellt wurden: Die Evidenz der Mathematik geschieht *atemporal*. Zwar bedarf es zur Erfüllung eines Programms der *Schrittfolge seiner Ausführung*, die eine *Zeitfolge* impliziert, wie sie wiederum als Matrix oder auf einem idealen (räumlichen) *Tableau* formuliert werden kann, doch ist die Richtigkeit eines Programms, seine Fehlerlosigkeit, *keine* Funktion der *Zeit*, sondern seiner Widerspruchsfreiheit. Wir erhaschen hier den Vorschein einer Differenz zwischen Syntax und Semantik oder Form und Sinn: Erstere rechnet *nicht* mit der *Zeit*, während letztere notwendig *zeitrelativ* verläuft. Ebenso sehr berühren sich in diesem Punkt die Aufzeichnungsarten mathematischer Programme mit denen anderer Programmformen: Entscheidend ist hier wie dort der Übergang von der *Zeit* zum *Raum*, von der Hintereinanderausführung zur Tabellierung, von der Skandierung oder Sequenzialisierung zur Synopsis. Er zeigt an, dass die Programme über *Raum* *und* *Zeit* herrschen und sich beide miteinander verschränken, dass aber ihre maßgebliche Organisationsform gerade *nicht* in der Verzeitlichung, sondern in der Verräumlichung besteht. Wir hatten diesen Aspekt bereits mehrfach gestreift: Das Medium des Programms bildet das *Diagramm*, die *Tafel* oder *Matrix*, die auch im Mathematischen ihre Stellung behauptet. Beide inkludieren die *stillgestellte* oder *angeeignete Zeit* – nicht ihre Performanz, vielmehr sistieren Diagramme, Tafeln und Matrizen die Prozesse in Gestalt von relationalen Konfigurationen oder ‚Anordnungen‘, die wiederum im Doppelsinn von ‚Anordnung‘ zwischen Struktur und Imperativ zu lesen ist. Immer wieder kehren wir auf diese Weise zum gleichen Topos zurück: Das mathematische Programm als universalisierte Form bezeichnet nichts anderes als eine *Syntax* im Sinne einer ‚Voraus-Schreibung‘ oder Vor-Schrift‘: Sie funktioniert analog einer Regel ohne eine zu sein. Gleichzeitig kombiniert sie eine *diagrammatische Ordnung* (Raum/Algebra) mit einer *Sukzession* (Zeit/Arithmetik). Ein allgemeiner Begriff des Programms findet daran seine wesentliche Bestimmung.

In der Tat bedeutet eine Turingmaschine in diesem Sinne „nichts anderes als eine *Turingtafel*“.²⁵ Sie verzeichnet allein ein Repertoire von Transformationen im Sinne *Vorwärts-, Rückwärts-, Wiederholungs- und Haltebewegungen* (Zeitlichkeit) einerseits und *Zeichensetzungs- bzw. Zeichenlöschungsfunktionen* (Räumlichkeit) andererseits. Dann genügt es, die *Turingtafel*, die *auf diese Weise ein formales ‚Programm-Modell‘ definiert*, allein auf vier Grundoperationen mit zusätzlichen *Zeitmarkern* t_i zurückzuführen, die ihre Nacheinanderausführung festlegen. Setzt man zusätzlich für mögliche Maschinenzustände das Symbol a_i sowie für die vier Arten von Instruktionen μ_k ein, erhält man als Zeilen:

$$t_j \quad a_i \quad b_i \quad \mu_k \quad t_n, \text{ mit } i, j, k, n = 0, 1, 2, \dots$$

und damit als Gesamtmatrix das Schema:

²⁴ Vgl. Wilfried Brauer, Klaus Indermark: Algorithmen, rekursive Funktionen und formale Sprachen, Mannheim 1968, S. 16f.

²⁵ Hermes, Aufzählbarkeit Entscheidbarkeit Berechenbarkeit, a.a.O., S. 26.

t_0	a_0	μ_1	b_0	t_1
t_0	a_0	μ_2	b_0	t_2
t_0	a_1	μ_1	b_1	t_3
...
t_j	a_i	μ_k	b_j	t_n
...
t_x	a_y	μ_0		

Alle *mathematischen* und folglich auch alle technischen Computerprogramme sehen so aus;²⁷ gleichzeitig machen sie jedoch nicht nur etwas *berechenbar*, sondern sie lassen sich auch *als Entscheidungsprozesse* verstehen, die mit den Zeilen $t_j \ a_i \ b_i \ \mu_k \ t_n$ aus einem binär geordneten Datenraum Alternativen aus 0 oder 1 bzw. ‚Ja‘ oder ‚Nein‘ selektieren. Der ‚Algorithmus‘, als jederzeit wohlfeile Antwort auf die Frage nach den ‚mathematischen Fundamenten des Computers‘, erfährt so eine andere Wendung, weil auf diese Weise die Grundoperationen der Turingmaschinen Entscheidungsfunktionen entsprechen, die ihrerseits nichts anderes als eine iterative Wahl zwischen 0 / 1, *rechts / links* oder *drucken / löschen* etc. gewährt. Algorithmen lassen sich demnach auf Entscheidungsprozesse zurückführen, die immer schon digital terminiert sind, wie überhaupt für digitale Medien charakteristisch ist, notorisch ihr *tertium* gleichermaßen auszuschließen wie ebenfalls das Verweilen ‚in der Mitte‘, im Unentschiedenen oder auf der Schwelle, in Zwischenräumen und Hybriden jenseits klassischer Dichotomien. Umgekehrt lassen sich nichtentscheidbare Prozesse oder abinäre Strukturen nicht programmieren: Alle Unbestimmtheit schwindet – Programme erfordern Binarisierungen, wie alle Opposition eindeutig ausfallen muss. Darum unterliegen Verfahren der Digitalisierung durchweg einem *Zwang zur Entscheidbarkeit*, wie sie an den Geräten selber, ihren Eingabefeldern oder Steuerungsmodulen, den Menüs der DVDs oder auch den Interfaces ablesbar ist, soweit diese aus einem Set von Icons bestehen, die allein eine Auswahl via Mouseclick zulassen. Selbst virtuell erzeugte Bilder, womit Simulationen und Computerspiele arbeiten, deren artifizielle Räume aus verzweigten Topologien bestehen, folgen noch der dyadischen ‚Logik‘ von Labyrinthen aus Gängen, Türen, Brücken und dergleichen, die grundsätzlich ‚hodologisch‘ organisiert sind.²⁸ Wir haben es demnach überall mit einem *zweiwertigen* Entscheidungsbegriff zu tun, der von existenziellen Entscheidungsbegriffen, wie sie etwa Heidegger oder Jean-Paul Sartre entfaltet haben, weit entfernt ist, insofern er lediglich Ja-Nein-Stellungen erlaubt, keine „Entschlossenheiten“.²⁹ Und wenn im Rahmen seiner Überlegungen zum „Vorlauf zum Tode“ Heidegger in *Sein und Zeit* darauf verweist, dass jede Entscheidung in einen Rahmen gestellt ist, wie er von der *Endlichkeit des Daseins* vorgegeben ist, dann bedeutet jede *konkrete* Wahl eine Beschneidung oder Grenze des Möglichen, während binäre Programme – aufgrund des unendlichen Turingbandes – mit infiniten Mengen von Alternativen operieren.

²⁶ μ_1 = Gehe-nach-rechts, μ_2 = Gehe-nach-links; μ_0 = stopp. Vgl. auch ebenda und S. 33ff.

²⁷ Vgl. auch Weizenbaum, *Die Macht der Computer*, a.a.O., S. 80ff.

²⁸ Vgl. dazu meine Überlegungen in: Dieter Mersch: *Logik und Medialität des Computerspiels*. Eine medientheoretische Analyse, in: J. Distelmeyer, Chr. Hanke, Dieter Mersch (Hg): *Game over? Zur Medialität des Computerspiels*, Bielefeld 2008, S. 19-41.

²⁹ Vgl. Martin Heidegger: *Sein und Zeit*, Tübingen, 12. Aufl. 1972, §§ 60, 62.

Entscheidungslogische Fundierung

Turingmaschinen eröffnen dergestalt nichts anderes als logische Entscheidungsräume: Sie kennen keine „Dramen“, keine Entscheidungsprobleme, keine Wahlen, von denen etwas abhängt, vielmehr nur solche Alternativen, die sich entlang diskreter Ordnungen durchdeklinieren lassen. Man muss allerdings sehen, dass die Begriffe der Berechenbarkeit und Entscheidbarkeit wechselseitig ersetzbar sind, denn bezeichnet M die Menge aller *berechenbaren* Werte einer Funktion f , dann kann x genau dann als *entscheidbar* in Bezug auf M gelten, wenn es ein eindeutiges Verfahren, d.h. einen endlichen Algorithmus gibt, der entweder $x \in M$ oder $\sim (x \in M)$ (\sim für ‚nicht‘) herleitet. Dabei gehören die Termini der Berechenbarkeit und des Algorithmus dem Bereich der Metamathematik an, während der Terminus der Entscheidbarkeit in das Gebiet der zweiwertigen (digitalen) Logik fällt. Entsprechend lassen sich mathematische Programme als Spielanleitungen auffassen, die Auswahlen (Entscheidungen) gemäß einer Regel (Algorithmus) treffen. Anders wiederum als Spiele, die ‚in der Regel‘ nicht programmiert sind – andernfalls wären sie keine –, sind ihre Regeln so gesetzt, dass sie einen Spielraum von Möglichkeiten aufschließen, der einer Dialektik von Bestimmung und Bestimmtwerden, von Offenheit und Geschlossenheit oder Kontingenz und Notwendigkeit gehorchen. Am Spiel ist genauso viel Zufall wie Determination oder Kausalität, ebensoviel Emergenz wie Ordnung, weil Spiele sich allererst durch eine Praxis der – im Wortsinne – Spielver/wendung (im Sinne von Gebrauch *und* Bruch mit den Regeln) verwirklichen.³⁰ Dagegen realisieren sich mathematische – wie auch mathematisierbare – Spiele als Programme: Der Erfolg von *Deep Blue* bezeugt den Triumph des Programms über die Strategie – doch ist damit noch kein Präjudiz für digitales ‚Denken‘ gegeben.³¹

Setzt man, darüber hinaus, hinzu, dass im Feld mathematisch-syntaktischer Konstruktionen Existenz und Widerspruchsfreiheit in dem Sinne zusammenfallen,³² dass das, was konsistent herleitbar ist, auch existiert, bedeutet das Treffen einer widerspruchsfreien Wahl nicht nur eine Möglichkeit, eine Virtualität, sondern bereits eine Wirklichkeit. Der mathematische Begriff des Virtuellen fällt mithin – solange er formal konstruierbar ist – mit dem Reellen zusammen. Dabei schließt Geltung innerhalb eines Kalküls weniger eine ‚Wahrheit‘ ein, als vielmehr eine ‚Richtigkeit‘ im Sinne eines *Sichrichtens nach Regeln* bzw. *eines Algorithmus*. Oder anders ausgedrückt: Existenz oder Gültigkeit meinen zugleich eine Konstruierbarkeit, wobei sämtliche Prinzipien der klassischen Logik unter Einschluss des *principium contradictionis* $\sim (a \text{ und } \sim a)$ (\sim für ‚nicht‘) vorauszusetzen sind. Folglich überschneiden sich auch die Kategorien der Richtigkeit und der Entscheidbarkeit: Eine Entscheidung oder ein ‚Spielzug‘ erweisen sich dann als ‚richtig‘, wenn sie durch ein zugrunde liegendes Programm oder einen Algorithmus verbürgt, d.h. in endlich vielen Schritten ‚entschieden‘ werden kann. Dann, in der Tat, gleichen Programme ‚primitiven‘ Spielen, die auf einem Set von Transformations- und Verknüpfungsregeln fußen, doch generiert nicht schon jedes Programm ein Spiel: Viele Spiele lassen sich zwar – wie das Beispiel des Schachcomputers lehrt, mathematisieren, aber bei weitem nicht alle – bestenfalls solche, die nicht auf Kreativität

³⁰ Vgl. meine Überlegungen in: Dieter Mersch: Spiele des Zufalls und der Emergenz, in: Maske und Kothurn. Internationale Beiträge zur Theater- Film- und Medienwissenschaft, (54. Jg.) 2008, Heft 4, S. 19-34.

³¹ Vgl. meine Überlegungen in: Dieter Mersch: Kunstmaschinen. Zur Mechanisierung von Kreativität, Körpermaschinen – Maschinenkörper. Mediale Transformationen, hg. v. Klaus-Peter Köpping, Bettina Papenburg, Christoph Wulf, Parerga. Internationale Zeitschrift für historische Anthropologie, Band 14 (2005) (2), S. 183-202.

³² Merthens, Moderne Sprache Mathematik, a.a.O., S. 300.

setzen, sondern allein aus binären Entscheidungsstrukturen folgen, die die strikte Herleitbarkeit bzw. Nichtherleitbarkeit von Spielkonstellationen durch Regeln garantieren. Sie werden freilich im Rahmen einer ‚Kultur des Digitalen‘ allgegenwärtig. Die Ubiquität der Programme in der Postmoderne bildet das Korrelat ihrer Durchdringung. Gleichzeitig avanciert der Entscheidungsbegriff zu einem Grundlagenbegriff sowohl digitaler Programme als auch der *Programmatik des Digitalen überhaupt*. Sie ruft nicht nur mit der Verkettung von ‚Entscheidbarkeit‘, ‚Berechenbarkeit‘, ‚Beweis‘ oder ‚Algorithmus‘ eine ganze Klasse verwandter Maxime auf, vielmehr modellieren diese, indem ‚ $x \in M$ ‘ bzw. ‚ $\sim(x \in M)$ ‘ durch die Ziffern 1 und 0 substituierbar erscheinen und damit einer unbeschränkten Manipulation Vorschub leisten, lauter *Entscheidungsprogramme*, die gleichzeitig einer generellen Tendenz zur Ludifizierung entgegenkommen. Weil Algorithmen ohne Bezug auf einen Inhalt, sondern nur über eine Menge von ‚digits‘, von mathematischen Objekten oder Daten operieren, d.h. ohne Semantik auskommen, neigt ihre Anwendung zum Ludischen. Sie eröffnen dadurch einen Raum beliebiger Kombinatorik innerhalb eines zuvor definierten Programmrahmens. Alle Programme gewinnen demnach die generelle Form einer Spielanleitung. In dem Maße, wie die Digitalisierung unterschiedslos technische, ästhetische und soziale Prozesse durchdringt, schlägt die Ludifikation als allgemeine kulturelle Gestalt durch. Das Digitale – wie das Ludische als seine Entsprechung – ist eine *Denkform*.³³

Dennoch erweist sich das *allgemeine* Problem der Entscheidbarkeit mathematisch als unlösbar, wie es auch keinen generellen Algorithmus gibt – kein ‚Meta-Programm‘ –, der alle anderen inkludiert. „In einem (...) formalen System lässt sich die Frage stellen, ob jeder syntaktisch regelrecht gebildete Ausdruck entweder wahr oder falsch ist. Dies ist das Entscheidungsproblem.“³⁴ Seine Lösung widersetzt sich einer universalen Methodik, denn jedes hinreichend komplexe und konsistente System, das mindestens der Prädikatenlogik zweiter Stufe entspricht, wie etwa die Mengenlehre, die Arithmetik oder andere komplexe mathematische Theorien, bleibt *unvollständig*, sofern sie notwendig *Unentscheidbarkeiten* gebiert – Aussagen, die zwar syntaktisch korrekt gebildet werden können, deren ‚Richtigkeit‘ oder ‚Falschheit‘ aber nicht entschieden werden können.³⁵ Daraus folgt, dass der *Begriff der Entscheidbarkeit* – wie auch der *Berechenbarkeit* und des *Algorithmus* oder die *Frage nach einem ‚Programm aller Programme‘* – sich seiner eigenen *Universalisierbarkeit* verweigert: Es handelt sich um offene, nicht ihrerseits totalisierbare Begriffe, die sich gegen jede allgemeine Schematisierung sperren. Wir hatten bereits auf die Vermutung der Unmöglichkeit einer nichtreduktiven Theorie aller Programme hingewiesen: Das Resultat korrespondiert mit dem „Turingschen Halteproblem“, das nach der Konstruierbarkeit einer Maschine fragt, die in endlich vielen Schritten *sich selbst*, d.h. alle ihre möglichen ‚Outpute‘ errechnet.³⁶ Einmal angesetzt, findet sie keinerlei Halt: Wie ein *circulus vitiosus* stürzt sie in den Abgrund einer unendlichen Selbstberechnung, wie sie zuletzt der *Grenze der Entscheidbarkeit* entspricht. Dasselbe gilt für die Programmierung: Kein Programm vermag sich selbst zu programmieren, sowenig wie es über alle möglichen Folgen seiner Programmierung verfügt: Stets haben wir

³³ Es gehört zu den fatalen medienwissenschaftlichen Verwechslungen, diese Denkform – im Sinne rationalitätskritischer Interventionen – mit ‚Nichtdiskursivität‘ bzw. einem *Differenzdenken* zu identifizieren. Vgl. dazu kritisch mein früher Aufsatz Dieter Mersch: Digitalität und nichtdiskursives Denken, in: ders, Kristof Nyíri (Hg): Computer, Kultur, Geschichte, Wien 1991, S. 109-126.

³⁴ Ebenda, S. 302.

³⁵ Hermes, Aufzählbarkeit Entscheidbarkeit Berechenbarkeit, a.a.O., vor allem S. 176ff.

³⁶ Vgl. Brauer, Indermark, Algorithmen, rekursive Funktionen und formale Sprachen, a.a.O., §§ 10 und bes. 11, S. 49ff.

es mit Einbrüchen von Kontingenzen oder Unwahrscheinlichkeiten zu tun, wie gleichfalls die Menge der Konsequenzen nie vollständig antizipierbar erscheint.

Grenzen des Programmierbaren

Allerdings hatte Turing das Modell seiner Maschine noch auf die Logik des Denkens selbst appliziert, ähnlich wie die Kybernetik Norbert Wieners, Heinz von Foesters oder Warren McCulloughs, die im Regelkreis, der „zirkulären Kausalität“ (Foerster) den Schlüssel zur Rekonstruktion kognitiver Prozesse erblickten.³⁷ Der verhängnisvolle Fehlschluss, auf dessen Konturen wir noch zu sprechen kommen werden, verkennt dabei insbesondere die Differenz zwischen *Rechnen* (Syntax) und *Bedeutung* (Semantik), wie sie der stets interpretierenden Vernunft zukommt. Während jedoch in der Kybernetik das Prinzip der Rekursion und damit einer formalisierten Reflexivität dominierte, suchte Turing das Problem ebenso spielerisch wie entscheidungstheoretisch zu lösen. Bekanntlich erfand er ein ‚künstliches Imitationsspiel‘, das – im Sinne eines Blindversuchs – vor die Wahl zwischen einem Menschen und einem Computer stellte, ohne dass der Spieler wissen konnte, wessen Antwort er vernahm, um so erneut vor die Frage nach der ‚*Programmierbarkeit*‘ von *Intelligenz* gestellt zu sein.³⁸ Von Anfang an gerinnt damit das Denkproblem zu einem Entscheidungs- und nicht mehr zu einem *Unterscheidungsproblem*, weil fortan bestenfalls die *Wette* zwischen Mensch und Maschine im Zentrum stand, nicht ihre jeweilige Bestimmung, sodass sich das Modell je schon *für* eine mögliche Programmierbarkeit entschieden hatte.³⁹ Die Frage ist dann bereits das Resultat, denn wenn für eine gegebene Antwort ‚a‘ unklar bleibt, ob sie von einem Menschen oder einer Maschine getroffen wurde, folgt aus ihrer Nichtentscheidbarkeit noch lange nicht, dass beiden – Mensch wie Maschine – gleichermaßen ein *Geist* bzw. – wenn schon – der *gleiche Geist* zugeschrieben werden muss, sondern lediglich, dass *auf diese Weise* dem Problem nicht beizukommen ist: Die Indifferenz von Mensch (M) und Maschine (M)⁴⁰ verweist gerade auf die *black box* (M/M), die sie umhüllt und den Ort ihrer Distinktion dadurch verwischt, dass dieser von Beginn an zu einem *entscheidbaren Resultat* depraviert. Gleichzeitig kann jedoch aus dem Scheitern des Turingtests auch der umgekehrte Schluss gezogen werden. Denn legt man den formalen Begriff des Programms als Entscheidungsprogramm zugrunde, erweist sich alles als programmierbar, was sich durch Wahlalternativen konstruieren und in Form von Operationsketten sequenzialisieren lässt. Schmiegt sich der Terminus des Programms so ans Technologische und besonders an die Schaltung und Digitalisierung an, brechen aus ihm gleichzeitig die Momente einer Dysfunktionalität, d.h. einer Widerständigkeit oder Unerfülltheit hervor. Sie bilden die andere Seite der Programme, ihr notwendiges Korrelat als *Dissonanz* oder *Zusammenbruch*. Von hier aus wäre dann das Verhältnis zwischen dem Programmierbaren und Nichtprogrammierbaren neu auszuloten, wie es besonders im dritten Teil der vorliegenden Beiträge thematisiert worden ist. Dabei kann die Grenze als Resultat unserer Überlegungen – zumindest für

³⁷ Vgl. insb. Heinz v. Foerster: Zirkuläre Kausalitäten. Die Anfänge einer Epistemologie der Verantwortung, in: Claus Pias (Hg): *Cybernetics – Kybernetik. The Marc-Conferences 1946-1953*, Zürich, Berlin 2003, S. 19-26.

³⁸ Vgl. Turing: *Rechenmaschinen und Intelligenz*, a.a.O., sowie ders., *Intelligente Maschinen. Eine häretische Theorie*, in: ders., *Intelligence Service*, a.a.O., S. 9-15.

³⁹ Mersch, *Kunstmaschinen*, a.a.O. Vgl. ferner Hubert L. Dreyfus: *Was Computer nicht können. Die Grenzen künstlicher Intelligenz*, Frankfurt/M. 1989. Zu den Grundlagen der Computerisierung vgl. auch Christel Schachtner: *Geistmaschine*, Frankfurt/M. 1993; Sybille Krämer: *Symbolische Maschinen*, Darmstadt 1988.

⁴⁰ Mit Bedacht wählen wir hier das gleiche Symbol, um die Absurdität der Operation deutlich zu machen: Die vermeintliche Indifferenz impliziert deren formale Identität.

mathematische Programmierungen – entlang der Bruchlinie zwischen Entscheidbarkeit und Unentscheidbarkeit festgemacht werden. Als klassische Unentscheidbarkeiten können dabei das Turingsche Halteproblem und die Gödelsche Unvollständigkeit angeführt werden: Sie treten bereits im Register formaler und formalisierbarer Prozesse als Demarkationen von Ipsoflexivitäten auf. Dann *gibt es* – wiederum formal – Nichtprogrammierbares, weil es Unberechenbares gibt. Insbesondere offenbaren sich die Ränder des Programmierbaren anhand der in die Programme selbst eingehenden Bedingungen ihrer Möglichkeit sowie ihrer impliziten, aber nie ganz beherrschbaren Effekte. Das Programmierende der Programme, genauso wie der Zwang, den ihre Anwendungen erfordern, oder der Ausschluss, den sie erzeugen, indem sie selbst noch zwischen dem Programmierten und dem unterscheiden, was keiner Programmierung unterliegt, bleiben ihrerseits schon aus logischen Gründen alternativlos und von der Entscheidung ausgenommen. Programme ‚sehen‘ nur, was sie selbst aufteilen oder anordnen, nicht das, was auch anders sein kann – vielmehr bleibt das Anders am Platz einer intrinsischen Blindheit. Was daher die Programme jeweils normieren, klassifizieren oder kontrollieren, verdankt sich ihrer ‚vor-gezeichneten‘ oder ‚vor-geschriebenen‘ Strukturierung, *durch* (,dia‘) dass das Heterogene, Unwahrscheinliche oder Nichtereignis gerade ausgenommen wird, wie umgekehrt deren ‚Ausnahme‘ die Bedingung ihres ‚Anmachens‘ und ‚Ausmachens‘ (On/Off), wie es bei Reinhold Görling und Stephan Trinkaus heißt, darstellt.

Zu trennen wären in diesen Überlegungen jedoch noch zwischen zwei ‚Anderesheiten‘, gleichsam einem ‚äußeren Anderen‘ und einem ‚Anderen‘ als immanente Heterogenität: Eine zweifache Alterität, entlang deren Grenzen die Programme ihre Rigorosität und Macht erst entfalten. Denn das ‚äußere Andere‘ folgt noch der Differenz zwischen Programm und Programmierung einerseits sowie andererseits dem, was durch diese *überhaupt nicht in Frage kommt oder in Betracht gezogen* werden, was ihr gleichsam auf immer fremd bleibt und nicht Gegenstand einer Einteilung oder Zurichtung sein kann – das Offene, worin die Teilung zwischen Programmiertem und Nichtprogrammierten noch vorgenommen und hineingeschnitten wird, während das ‚innere Heterogene‘ dessen intrinsische Barriere markiert, gewissermaßen jene paradoxen Stellen, an denen die Programmierung sich auf sich selbst zurückbeugt und ins Unentscheidbare oder Unberechenbare ausläuft. Solche Stellen lassen sich – im Digitalen – nicht nur logisch aufweisen, sondern wären gleichfalls mit Blick auf die Materialität und Performativität der Programme auszubuchstabieren: Schon das Tableau, die Tafel als Medium gehört zur Kategorie des ‚äußeren Anderen‘, ferner die Nichtprogrammierbarkeit der Konsequenzen, denn die Programme ‚schreiben vor‘, ohne das, was sie präskribieren, schon zu überblicken, geschweige denn das ‚Wie‘ ihrer ‚Vor-Schriften‘ mitzeterminieren. Was sie bewirken, wohin sie ausgreifen, welche weiteren Felder sie noch zu erobern vermögen, kann kein Teil der Programme im Sinne eines von ihnen Programmierbaren sein; es bleibt ihnen vielmehr äußerlich, ein Effekt ihrer Anwendung, Übertragung oder Umwidmung sowie jenes Unvorhersehbaren, das jeder Ordnung in der Zeit eignet. Zudem zeigt sich überall dort, wo sich Paradoxa einfinden, die Fragilität der Programmierung: Sie entzieht sich den Kalkülen der Entscheidbarkeit, um an ihnen das grundsätzlich Unbestimmte kultureller Prozesse aufscheinen zu lassen. Entsprechend erweisen sich das Neue oder die Kreativität als Kategorien ‚immanenter Heterogenität‘ im eigentlichen Sinne als unprogrammierbar; gäbe es eine solche Programmierung, hätten wir es mit einer *contradictio in adjecto* zu tun, weil auf diese Weise die Eingebung oder Erfindung

vorhersehbar würde – trotz der verschiedenen Versuche, besonders in den technoiden Künsten der 1960er und 70er Jahre, etwa bei Abraham Moles, Max Bense, Herbert W. Francke oder Lejaren A. Hiller und Leonard M. Isaacson, Emergenzen und schöpferische Unwahrscheinlichkeiten durch stochastische Simulationen oder Generierung von Zufallsreihen nachzuahmen.⁴¹ Stattdessen bezeugen sie ihre Vergeblichkeit dadurch, dass sie den Glanz der *creatio* zur Kontingenz abstumpfen.

Aspekte des Nichtprogrammierbaren

Begriffe wie Denken, Kreativität, Wissen, Zeitlichkeit, Referenz, Leiblichkeit oder Bewusstsein und ähnliches zeigen sich so als doppelte Aspekte eines Nichtprogrammierbaren, auch wenn zunehmend die technische Programmierung beginnt, in sie einzubrechen, um neu zu modellieren. Sie bilden nicht nur illusionäre Zuschreibungen des Menschen, wie Friedrich Kittler insinuiert hat, durch die dieser sich von der Maschine zu unterscheiden trachtet,⁴² vielmehr erscheinen sie für die Beschreibung sämtlicher menschlicher Hervorbringungen unverzichtbar. Gleichzeitig bekunden diese dort ihre unveräußerbaren Grenzen, wo sie ihr Hervorbringen nicht selbst wieder hervorbringen können, sondern *aus Anderem* schöpfen. Dem entspricht in Ansehung mathematische Programme erneut das bereits genannte Halteproblem und in Bezug auf die logische Begriffssprache das Russellsche Paradox. Dass jedoch umgekehrt das Turingsche Halteproblem noch nie eine konkrete Realisierung von Programmen abgeschnitten oder behindert hat, spricht nicht gegen diesen Befund: Vielmehr steuern Programme Abläufe, machen operabel, während das Halteproblem allein im Moment der Unentscheidbarkeit von Selbstreferenz aufscheint. Jederzeit lassen sich alle möglichen Programme entwickeln, solange sie einer bestimmten Arithmetik und Konsistenz folgen; doch vermögen Programme ihre eigene ‚Programmizität‘ weder zu berechnen noch selbst zu programmieren – vielmehr sind wir mit einer grundlegenden *Unschärfe* oder *Unmöglichkeit programmatischer Konstruktionen* konfrontiert: Überall bedarf es – gemäß der Russellschen Antinomie und ihrer Zerlegung von Terminologien in Objekt- und Metasprachen – der Selbstexklusion. Sie garantiert allererst ihre Funktionalität.

Der Umstand verweist auf eine prinzipielle Nichtdarstellbarkeit, die mit dem korrespondiert, was Heidegger als Reduktionen des Denkens auf Rechnen kritisiert hatte.⁴³ Bedeutet Rechnen in erster Linie eine syntaktische, regelgeleitete Operation, in der Semantiken nicht nur keine Rolle spielen, sondern auch ihrer nicht bedürfen, gehört zum Denken die Fähigkeit, Unterscheidungen treffen, wie auch die semantische Strategie des Folgerns oder die unerschöpfliche Erfindung von Reflexionen und Selbstreflexionen. Sie fügen sich keiner Programmierung.⁴⁴ Ebenso wenig wie Sinn in Syntax aufgeht, entzieht sich auch die Offenheit der Bezugnahme der Formalisierung:⁴⁵ Wir berühren hier von neuem den Punkt

⁴¹ Vgl. dazu Mersch, *Spiele des Zufalls und der Emergenz*, a.a.O., sowie ferner auch: Manfred Eigen, Ruthild Winkler: *Das Spiel. Naturgesetze steuern den Zufall*, München: Piper 1975, S. 344ff.

⁴² „Was Mensch heißt, bestimmen keine Attribute (...), sondern technische Standards.“ Vgl. Friedrich Kittler: *Draculas Vermächtnis. Technische Schriften*, Leipzig 1993, S. 63; ähnlich ders.: *Grammophon, Film, Typewriter*, Berlin 1986, S. 332.

⁴³ Vgl. Martin Heidegger: *Identität und Differenz*, Pfullingen 1957, S. 30, sowie ders.: *Gelassenheit*, Pfullingen 1959, S. 12f., ders.: *Was heißt denken?* in: ders.: *Vorträge und Aufsätze*, Pfullingen 1954, bes. S. 127ff.

⁴⁴ Vgl. Bertrand Russell: *Einführung in die mathematische Philosophie*, Wiesbaden 1960, Kap. 17, 18, S. 198ff; ders.: *Die Entwicklung meines Denkens*, München 1973ff., 6. Kap. ff, S. 66ff., ferner Gregory Bateson: *Double bind* 1969, in: ders.: *Ökologie des Geistes*, Frankfurt/M. 2. Aufl. 1983, S. 353ff.

⁴⁵ Nach Tarski gehört zu den Bedingungen einer syntaktisch definierten Semantik eine „wesentlich reichhaltigere“ Metasprache über eine Objektsprache; vgl. Alfred Tarski: *Die semantische Konzeption der Wahrheit und die*

einer Unmöglichkeit nicht-widerspruchsfreier Programmierung, der auf den Begriff des Menschlichen selbst zurückführt. Anders formuliert: Jede Programmierung bildet eine Form; aber es gibt keine Form der Form, keine Vorhersage eines Bezugs, sowenig wie die Möglichkeit einer programmatischen Ereignung von Andersheit.

Ein weiteres kommt hinzu: Programme entfalten ihre Performativität allein in Gestalt positiver Operationalisierungen. Weder vermögen sie im strengen Sinne Verneinungen zu programmieren, noch das ‚Nichts‘ im Sinne eines ebenso unkonturierten wie unbestimmten und offenen Raumes; vielmehr affirmieren sie sich selbst. Es gibt folglich im strengen Sinne keine Programmatik der Negativität – sondern stets nur die positive Entfaltung von Regularien, Schrittfolgen oder ‚Anordnungen‘. Sind wir so erneut mit dem konfrontiert, was wir ihr ‚äußeres Anderes‘ bezeichneten, wird an ihm deutlich, dass zwar die Programme Wahlen ermöglichen, nicht aber ihrerseits wieder wählbar erschienen. Sie stiften Alternativen, fügen sich jedoch selbst keiner Entscheidbarkeit. Wir haben diesen Punkt bereits mehrfach gestreift: Ein Programm erstellen, es anwenden heißt, seine Wahlen schon gewählt zu haben, bedeutet die spezifischen Strukturen seiner Entscheidungen bereits affirmiert zu haben. Es gibt hier nur alles oder nichts: Entweder wird ein Programm ausgeführt oder nicht: Als Ganzes steht es *nicht* zur Disposition: *Es gibt hier kein Drittes*, keinen Zwischenraum; es gilt nur das *Ja* oder das *Nein*, weshalb jedes Programm seine eigene Anwendung alternativlos diktiert. Davon unbenommen bleibt, dass sich Programme auch *gegenwenden*, *unterlaufen* oder *unterbrechen* lassen, doch ist die Negation kein Teil ihrer selbst, vielmehr bedarf es dazu gerade jener Kreativität, die der Programmierung *entwendet* bleibt. Entsprechend gibt es auch kein *Dazwischen*, keine halbe, uneigentliche oder metaphorische Verwendung – auch wenn ein Programm nur teilweise verwendet werden kann, wie nahezu alle Computerprogramme verdeutlichen: Doch ist hier Partialität eine Funktion von Selektion, nicht des Abbruchs. Daraus folgt auch: Mit der Wahl eines Programms haben wir seine Geltung bereits bestätigt und uns *für es* schon entschieden; die Frage nach seiner Anwendbarkeit oder Nichtanwendbarkeit ist an dieser Stelle gleichsam ausgesetzt. Demgegenüber folgt der Entschluss zur Ausführung oder Unterlassung eines Programms nicht länger den Direktiven einer Entscheidbarkeit, sondern einzig der praktischen Vernunft, die sie unter ihre Verantwortung stellt, sodass ihre Verwendung beginnt, wie Derrida sich ausdrückt, „unverantwortlich zu sein“.⁴⁶ Es gibt also, mit anderen Worten, einen technologischen Imperativ zur Programmierung, der umgekehrt ebenfalls einen Imperativ zum Technologischen einschließt. Und indem die mathematische Programmierung als Entscheidungsprogramm im Technischen gleichsam zu sich selbst kommt, affirmiert sie sich nicht nur selbst, sondern tendenziell auch die Regime einer Technologie, die sich der ‚Digitalität‘⁴⁷ und ihrer Programmatiken bedient, um in dieser Hinsicht ebenso sehr ihre Unverantwortlichkeit zu evozieren.

Das bedeutet keineswegs, wie man vermuten könnte, das Programmierbare und seine Grenze aus einer bereits ‚vorgeschobenen‘ Differenz zwischen Mensch und Technik und damit einem

Grundlagen der Semantik (1944), in: Gunnar Skirbekk (Hg): Wahrheitstheorien, Frankfurt/M. 1977, S. 140ff. Zur Kritik der KI-Forschung John R. Searle: Geist, Hirn und Wissenschaft, Frankfurt/M. 1986, S. 31f.

⁴⁶ Vgl. Jacques Derrida: Das andere Kap, Frankfurt/M. 1992, S. 36.

⁴⁷ Diesen fruchtbaren Ausdruck entnehme ich Jan Distelmeyer: Game-Glanz. Spielen im Licht der Digitalität, in: Jürgen Sorg, Jochen Venus (Hg.): Erzählformen im Computerspiel. Zur Medienmorphologie digitaler Spiele, Bielefeld 2011 sowie ausführlicher ders.: Das flexible Kino. Ästhetik und Dispositiv der DVD, Berlin 2012 (im Erscheinen).

bestimmten Begriff des ‚Humanums‘ abzuleiten, um den Programmen gewissermaßen das entgegenzuhalten, was sie im Unterschied zu diesem nicht können. Tatsächlich gilt im Zeichen technologischer Rationalitäten eher das Gegenteil: Die Einverleibung dessen, was der Mensch *ist*, durch die hegemoniale Macht der Programme und einer ebenso syntaktisch wie mathematisch operierenden Technik, um ihm, dem Menschen, seinen Begriff und seine Bestimmung, insbesondere seinen „Geist“ „auszutreiben“. ⁴⁸ So werden seit Wiener und McCullough nicht nur die Strukturen programmierbarer Computer den Strukturen und vermeintlichen ‚Programmen‘ des Denkens entlehnt, die sich für die ‚Logik‘ und ihre ‚Algebra‘ immer schon vorentschieden haben, sondern diese dienen umgekehrt als Metapher für die Beschreibung des Funktionskreises des Gehirns als Denkorgan, indem angeblich die Synapsen als Schalter fungieren, die immer dort, wo das Gehirn denkt, Rechenprozesse ausführen. ⁴⁹ Wir bekommen es dann mit einer exakten Rückprojektion einer diskursiv-technischen Reduktion auf das zu tun, was Denken bedeutet – eine Bedeutung, die im Denken niemals zu erschließen ist –, um aus ihm umgekehrt ein Bild des Menschen und seines ‚Humanums‘ zu gewinnen, das diesem entspricht, wie es auf der anderen Seite bestenfalls dazu taugt, die Grenzen der Computerisierung und ihrer Programmierung kenntlich werden zu lassen. Das doppelte Spiel – in beiden Fällen durch eine Verkennung bedingt – funktioniert dabei wie ein Spiegel oder die zwei Seiten eines Blattes, das dieselbe Medaille enthüllt: doch steht dabei letztlich das Schicksal des Menschen überall auf einem anderen Blatt.

⁴⁸ Vgl. Friedrich Kittler (Hg): *Austreibung des Geistes aus den Geisteswissenschaften*, Stuttgart 1992.

⁴⁹ Tatsächlich gründet dieser ‚Kurzschluss‘ in einer doppelten Operation: *zum einen* der ‚digitalen‘ bzw. ‚binären‘ Definition des Denkens, die entscheidungslogisch in einem strikten ‚Entweder-Oder‘ gründet und keine Graustufen oder Zwischentöne duldet – die biologische Entsprechung bildet eine Vorstellung vom Gehirn als ‚Nervenapparat‘, in dem die Synapsen entweder ‚feuern‘ oder nicht und so die Rolle binärer Schaltungen übernehmen –, *zum anderen* der Unterstellung, dass die Nervenzellen ‚nach Programm‘ laufen. Basiert die erste Annahme auf einem Reduktionismus, wie er gleichermaßen im Turing-Test seinen Ausdruck findet, ist die zweite Annahme irrig: Es gibt kein ‚übergeordnetes‘ Programm der Zellen, vielmehr ist es eher das Interaktionsgeschehen der Zellen untereinander, welche alle ihre eigene Struktur besitzen, das ‚Denken‘ erzeugt. Die Gleichsetzung von Gehirn und Computer scheidet daran: dieser arbeitet nach implementierten Algorithmen, die sich zu den einzelnen Chips und Prozessoren indifferent verhalten, jedes gleicht einem offenen Prozess ohne Steuerung und Kontrolle, aus dem sich ‚Denken‘, was immer dieser Ausdruck bedeutet, jeden Augenblick neu erfindet. Mindestens wären dazu sowohl die Möglichkeit einer selbständigen Vernetzung der Elemente nach den Erfordernissen eines Außen als auch die Minimalbedingungen einer Reziprozität und Responsivität noch vorzusetzen.

Literatur:

Barthes, Roland: Lektion/Leçon: Antrittsvorlesung im Collège des France. Französisch und Deutsch, Frankfurt/M 1980.

Bateson, Gregory: Double bind 1969. In: ders., Ökologie des Geistes. Frankfurt/M 2. Aufl. 1983, S. 353 ff.

Baudrillard, Jean: Requiem für die Medien. In: Kool Killer, Berlin 1978, S. 83-118.

Becker, Heinrich; Hermann Walter: Formale Sprachen. Braunschweig 1977.

Brauer, Wilfried; Klaus Indermark: Algorithmen, rekursive Funktionen und formale Sprachen. Mannheim 1968.

Brecht, Bertolt: Radiotheorie. In: Günter Helmes, Werner Köster (Hg.): Texte zur Medientheorie. Stuttgart 2002, S. 148-154.

Chomsky, Noam: Syntactic Structur. The Hague 1957.

Derrida, Jacques: Das andere Kap. Frankfurt/M 1992.

Distelmeyer, Jan: Game-Glanz. Spielen im Licht der Digitalizität. In: Jürgen Sorg, Jochen Venus (Hg.): Erzählformen im Computerspiel. Zur Medienmorphologie digitaler Spiele. Bielefeld: 2011 (im Erscheinen)

Distelmeyer, Jan: Das flexible Kino. Ästhetik und Dispositiv der DVD. Berlin 2012 (im Erscheinen).

Dreyfus, Hubert L.: Was Computer nicht können. Die Grenzen künstlicher Intelligenz. Frankfurt/M 1989.

Eigen, Manfred; Ruthild Winkler: Das Spiel. Naturgesetze steuern den Zufall. München: Piper 1975.

Enzensberger, Hans-Magnus: Baukasten zu einer Theorie der Medien. München 1997.

Foerster, Heinz v.: Zirkuläre Kausalitäten. Die Anfänge einer Epistemologie der Verantwortung. In: Claus Pias (Hg): Cybernetics – Kybernetik. The Marc-Conferences 1946-1953. Zürich, Berlin 2003, S. 19-26.

Heidegger, Martin: Was heißt denken? In: ders., Vorträge und Aufsätze. Pfullingen 1954.

Heidegger, Martin: Identität und Differenz. Pfullingen 1957.

Heidegger, Martin: Gelassenheit. Pfullingen 1959.

Heidegger, Martin: Sein und Zeit, Tübingen, 12. Aufl., 1972.

Heidegger, Martin: Bauen Wohnen, Denken. In: ders., Vorträge und Aufsätze, Pfullingen 4. Aufl. 1978, S. 139-156.

Heidegger, Martin: Die Kunst und der Raum. St. Gallen, 2. Aufl., 1983.

Hermes, Hans: Aufzählbarkeit Entscheidbarkeit Berechenbarkeit, Berlin, Heidelberg, New York 2. Aufl., 1971.

- Hickethier, Knut: Das Programm – Schlüsselbegriff der Medienwissenschaft. Zur Programmgeschichtsforschung der Rundfunkmedien. In: Medien und Zeit, Heft 2 (2008), S. 13-21.
- Holz, Günter; Hermann Walter: Automatentheorie und formale Sprachen. Mannheim 1968.
- Kittler, Friedrich: Grammophon, Film, Typewriter. Berlin 1986. Kittler, Friedrich (Hg): Austreibung des Geistes aus den Geisteswissenschaften. Stuttgart 1992.
- Kittler, Friedrich: Draculas Vermächtnis. Technische Schriften. Leipzig 1993.
- Krämer, Sybille: Symbolische Maschinen. Darmstadt 1988.
- Krämer, Sybille: ‚Schriftbildlichkeit‘, oder: Über eine (fast) vergessene Dimension der Schrift. In: Horst Bredekamp, Sybille Krämer (Hg): Bild, Schrift, Zahl, München 2003, S. 157-176.
- Krämer, Sybille: Operationsraum Schrift. Ein Perspektivenwechsel im Schriftverständnis. In: Gernot Grube, Werner Kogge, Sybille Krämer (Hg): Schrift. Kulturtechnik zwischen Auge, Hand und Maschine. München 2005, S. 13-32.
- Krämer, Sybille: Karten – Kartenlesen – Kartografie. Kulturtechnische Überlegungen. In: Philine Helas et al. (Hg): Bild / Geschichte. Festschrift für Horst Bredekamp. Berlin 2007, S. 73-82.
- Kripke, Saul A.: Wittgenstein über Regeln und Privatsprache. Frankfurt/M 1987.
- Latour, Bruno: Der Berliner Schlüssel: Erkundungen eines Liebhabers der Wissenschaften. Berlin 1996.
- Latour, Bruno: Das Parlament der Dinge. Für eine politische Ökologie. Frankfurt/M 2006.
- Mersch, Dieter: Digitalität und nichtdiskursives Denken. In: ders, Kristof Nyíri (Hg): Computer, Kultur, Geschichte. Wien 1991, S. 109-126.
- Mersch, Dieter: Die Geburt der Mathematik aus der Struktur der Schrift. In: Gernot Grube, Werner Kogge, Sybille Krämer (Hg): Schrift. Kulturtechnik zwischen Auge, Hand und Maschine. München 2005, S. 211-233.
- Mersch, Dieter: Kunstmaschinen. Zur Mechanisierung von Kreativität, Körpermaschinen – Maschinenkörper. Mediale Transformationen, hg. v. Klaus-Peter Köpping, Bettina Papenburg, Christoph Wulf, Parerga. Internationale Zeitschrift für historische Anthropologie, Band 14 (2005) (2), S. 183-202.
- Mersch, Dieter: Medientheorie zur Einführung. Hamburg 2006.
- Mersch, Dieter: Logik und Medialität des Computerspiels. Eine medientheoretische Analyse. In: J. Distelmeyer, Chr. Hanke, Dieter Mersch (Hg): Game over? Zur Medialität des Computerspiels. Bielefeld 2008, S. 19-41.
- Mersch, Dieter: Spiele des Zufalls und der Emergenz. In: Maske und Kothurn. Internationale Beiträge zur Theater- Film- und Medienwissenschaft, (54. Jg.) 2008, Heft 4, S. 19-34.
- Mersch, Dieter: Meta / Dia. Zwei unterschiedliche Zugänge zum Medialen. Zeitschrift für Medien- und Kulturforschung, Bd. 2 (2010) Hamburg, S. 185-208.

Merthens, Herbert: *Moderne Sprache Mathematik*, Frankfurt/M 1990.

Paech, Joachim: *Das Programm der Moderne und dessen postmoderne Auflösungen: Vom Werk zum Text zu Multimedia*. In: ders., Andreas Schreitmüller, Albrecht Ziemer (Hg), *Strukturwandel medialer Programme. Vom Fernsehen zu Multimedia*. Konstanz 1999, S. 13-30.

Peters, John Durham: *Speaking into the Air. A History of the Idea of Communication*. The University of Chicago Press 1999.

Russell, Bertrand: *Einführung in die mathematische Philosophie*. Wiesbaden 1960.

Russell, Bertrand: *Die Entwicklung meines Denkens*. München 1973.

Schachtner, Christel: *Geistmaschine*. Frankfurt/M 1993.

Searle, John R.: *Geist, Hirn und Wissenschaft*. Frankfurt/M 1986.

Tarski, Alfred: *Die semantische Konzeption der Wahrheit und die Grundlagen der Semantik* (1944). In: Gunnar Skirbekk (Hg): *Wahrheitstheorien*, Frankfurt/M 1977, S. 140ff.

Turing, Alan: *Intelligence Service*. Hg. von Bernhard Dotzler und Friedrich Kittler, Berlin 1987.

Weizenbaum, Josef: *Die Macht der Computer und die Ohnmacht der Vernunft*. Frankfurt/M 1977.

Wittgenstein, Ludwig: *Philosophische Untersuchungen* Frankfurt/M 1971.

Wittgenstein, Ludwig: *Philosophische Grammatik*, Frankfurt/M 1973.

