

# “Laden eines Computerprogramms als Vervielfältigung?” – Eine wesentliche Frage falsch gestellt. (Teil 2)

Felix Gantner

<b>Teil 1 (im letzten Heft – 8/94*)</b> 1. Einleitung 2. Argumentation mit technischen Details 2.1 Abstraktionsebenen der Informatik 2.2 Abstraktionsebenen im rechtlichen Diskurs 2.3 Das Verhältnis Programm – Hardware (*Heft 8/94 kann kostenlos beim Verlag angefordert werden.)	<b>Teil 2</b> 3. Die Universalmaschine Computer 3.1 Die Turingmaschine 3.1.1 Der Aufbau der Turingmaschine 3.1.2 Die universelle Turingmaschine 3.2 Maschinen und Datenverarbeitung 3.2.1 Algorithmus und Maschine 3.2.2 Daten und Information 3.2.3 Hardware und Software	<b>Teil 3 (im nächsten Heft – 10/94)</b> 4. Rechtliche Beurteilung 4.1 Ist das Laden eines Programms eine Vervielfältigung? 4.2 Netzwerke 4.2.1 Datenaustausch im Netz 4.2.2 Programme im Netz 4.3 Weiterführende Anmerkungen 5. Zusammenfassung
---	--	---

## 3. Die Universalmaschine Computer

Der grundlegende Unterschied zwischen Waschmaschine, Webstuhl, Klavier und Computer ist die Tatsache, daß ein Computer die technische Realisierung einer Universalmaschine ist. Um zu verstehen, was eine Universalmaschine ist, wie sie funktioniert und wodurch sie sich von anderen Maschinen unterscheidet, muß man das grundlegende Modell dafür betrachten: Es ist dies die Turingmaschine.

### 3.1 Die Turingmaschine

#### 3.1.1 Der Aufbau der Turingmaschine

Rund ein Jahrzehnt bevor die ersten Computer gebaut wurden, veröffentlichte der britische Mathematiker *Alan M. Turing*<sup>20</sup> die theoretischen Grundlagen dafür und stellte das mathematische Modell einer sehr einfach aufgebauten Maschine, der nach ihm benannten Turingmaschine vor. Auch wenn er dieses Modell nicht entwickelte, um Computer zu bauen, sondern um ein mathematisches Problem, das von *Hilbert* formulierte Entscheidungsproblem<sup>21</sup> zu lösen, so darf die Bedeutung dieses Modells für die Entwicklung von Universalrechnern nicht unterschätzt werden: Es wurde die grundsätzliche Möglichkeit des Baus eines Universalautomaten gezeigt und der Umfang dessen und was durch ihn berechenbar<sup>22</sup> ist, beschrieben. Außerdem diente dieser Archetyp<sup>23</sup> einer Maschine als Vorbild für die Konstruktion der heute verwendeten Computer.

Eine Turingmaschine besteht nur aus einem sehr einfach aufgebauten Apparat, der eine endliche Anzahl von internen Zuständen einnehmen kann, und einer Schreib/Lese-Einrichtung, die mit einem unendlich langen Band versorgt wird, das in einzelne Felder unterteilt ist. Die Felder des Bandes können entweder leer oder mit einem Symbol beschrieben sein. Die Maschine tastet pro Arbeitsschritt den Inhalt eines einzigen Feldes mit der Schreib/Lese-Einrichtung ab.

Bei jedem Arbeitsschritt kann die Turingmaschine folgende Handlungen ausführen: Sie kann anhalten, das Band um ein Feld nach links oder rechts bewegen und das Feld, auf dem die Schreib/Lese-Einrichtung positioniert ist, löschen, bzw. mit einem Symbol beschreiben. Das Band enthält den gesamten Input, den die Turingmaschine bearbeiten soll. Das Ergebnis der ausgeführten Berechnung, der Output, wird auch auf das Band geschrieben, wobei bei jedem Arbeitsschritt nur ein Feld des Bandes mit einem Symbol beschrieben werden kann.

### Ein Urahn unseres Computers ...

Turing, *Alan Mathison* (1912-54), British mathematician, whose pioneering work on theories of computer logic came to be fully appreciated only after his death. He was born in London and educated at Cambridge and Princeton universities. In his 1936 paper, "On Computable Numbers," published while he was still a graduate student, Turing introduced the concept of a theoretical computing device now known as a Turing machine. The mode of logical operation indicated the path of development that would later be followed in the creation of the digital computer. During World War II, Turing worked as a cryptographer for the British Foreign Office, and he also extended his mathematical work to the study of artificial intelligence and biological forms. Turing died, perhaps accidentally, of self-administered poison at the age of 41. "Turing, Alan Mathison," Microsoft (R) Encarta. Copyright (c) 1993 Microsoft Corporation. Copyright (c) 1993 Funk & Wagnall's Corporation

... wie er funktioniert ...

... und was er kann.

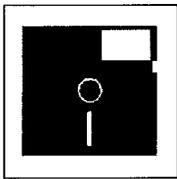
Dipl.-Ing. Mag. Felix Gantner, Röhrenbach (Österreich).

<sup>20</sup> Turing, On Computable Numbers, with an Application to the Entscheidungsproblem, Proceedings of the London Mathematical Society (2) 42, 1937 (Eine Übersetzung ins Deutsche liegt vor in Dotzler – Kittler (Hrsg.), Intelligence Service (1987), 17).

<sup>21</sup> Vgl. dazu Hopcroft, Turingmaschinen, Spektrum der Wissenschaft 7/84, 34.

<sup>22</sup> Vgl. dazu Vollmer, Denkzeuge, in Fischer (Hrsg.), Mannheimer Forum 90/91 (1991), 59 ff.

<sup>23</sup> Beer, Kybernetik und Management (1962), 114.



Das Verhalten der Turingmaschine hängt nur von ihrem augenblicklichen internen Zustand ab. Dieser wiederum wird ausschließlich durch den vorhergehenden internen Zustand und den Inhalt des eingelesenen Zeichens bestimmt. Diese internen Zustände und Zustandsübergänge sind daher der „Befehlssatz“ der Maschine und bestimmen den Algorithmus, den sie ausführen soll.

Turing zeigte mit diesem Modell, daß zu jedem Algorithmus bei gleichbleibendem Aufbau der Maschine nur durch die Festlegung des „Befehlssatzes“ eine Turingmaschine konstruiert werden kann, welche die durch den Algorithmus festgelegte Berechnung ausführt.<sup>24</sup> Es gibt daher für jede Berechnung, die ein moderner Digitalcomputer durchführen kann, auch eine spezielle Turingmaschine, die genau diese Rechnung ausführt.<sup>25</sup> Sie kann allerdings nur diese eine Berechnung durchführen.

Zu jeder Berechnung, die ein Computer ausführen kann, läßt sich daher auch eine Turingmaschine konstruieren, die eben diese Berechnung auch in einer endlich langen Zeit ausführt und umgekehrt. Was eine Turingmaschine jedoch nicht berechnen kann, kann auch niemals durch einen Computer berechnet werden.

Vom Einfachen zum  
Universellen

### 3.1.2 Die universelle Turingmaschine

Neben dieser für die Mathematik und auch Informatik fundamentalen Beschreibung von Berechenbarkeit leistet das Modell der Turingmaschine auch noch etwas weiteres grundlegendes: es zeigt die Möglichkeit der Entwicklung einer Universalmaschine.

Die bisher beschriebene Turingmaschine ist keine Universalmaschine. Ihr Verhalten wird durch den Befehlssatz bestimmt. Jede unterschiedliche Berechnung erfordert die Definition und Implementation eines neuen, speziellen Befehlssatzes und damit die Konstruktion einer neuen speziellen Turingmaschine.

Turing zeigte aber auch die Möglichkeit der Konstruktion einer universellen Turingmaschine. Diese Maschine ist ein universeller Imitator. Wird ihr über das Band eingegeben, welche beliebige, spezielle Turingmaschine imitiert werden soll, so verhält sie sich genau wie diese, dh. zu jedem Input liefert sie denselben Output wie die imitierte spezielle Maschine.

Es muß somit nicht mehr für jede Berechnung eine eigene Turingmaschine durch die Festlegung des „Befehlssatzes“ konstruiert werden, sondern nur mehr angegeben werden, welche Maschine imitiert werden soll. Bei identem „Befehlssatz“ können daher unterschiedliche spezielle Turingmaschinen realisiert werden. Die Programmierung der Universalmaschine tritt an die Stelle der Arbeit des Technikers an einer konkreten Maschine, wobei Programmierung letztlich die Beschreibung der Funktionsweise der zu realisierenden speziellen Maschine ist.<sup>26</sup>

Aufhebung des Unterschieds  
zwischen Plan und Realisierung

„Versorgt mit den codierten Instruktionen der speziellen Maschine, liefert die universale Maschine eine Beschreibung dessen, was die spezielle Maschine faktisch tun würde. Dabei gehören die Instruktionen (über die zu imitierende Maschine) und die Beschreibung (der Zustände der imitierten Maschine) ein und derselben Sprache an. Die gemeinhin in Unterschieden der Notationsweisen zum Ausdruck gebrachte Differenz zwischen einer Instruktion und ihrer Realisierung ... wird im Modell der universalen Turingmaschine obsolet.“<sup>27</sup> Der augenfällige Unterschied zwischen der Beschreibung eines technischen Konstrukts und seiner Realisierung – wie zB. bei Plänen im Maschinenbau und den danach gefertigten Werkstücken, bei Zeichnungen des Architekten und dem errichteten Gebäude – scheint daher bei der Universalmaschine zu verschwinden. Dies ist der Grund für all die urheberrechtlichen Probleme, die die rechtliche Einordnung von Computerprogrammen hervorruft.

Zu beachten ist jedoch, daß die Universalmaschine in dem Moment, in dem sie eine spezielle Maschine imitiert, sich nicht wie eine universelle Turingmaschine, sondern wie die nachgeahmte spezielle verhält und daher als diese anzusehen ist.

Dies führt zur Frage, was eigentlich eine Maschine ist und wieso eine Maschine, die eine andere nachahmt, als diese anzusehen ist, obwohl der konstruktive Aufbau – die Hardware – gleich bleibt.

<sup>24</sup> Chaitin, Zahlen und Zufall, in Reichel – *Prat de la Riba* (Hrsg.), Naturwissenschaft und Weltbild (1992) 30 (39).

<sup>25</sup> Chaitin, On the Difficulty of Computations, IEEE Transactions on Information Theory IT-16 (1970), 5; Hopcroft, a.a.O., 34.

<sup>26</sup> Bolz, Computer als Medium – Einleitung, in Bolz – Kittler – Tholen (Hrsg.), Computer als Medium (1994), 12.

<sup>27</sup> Krämer, Denken als Rechenprozedur: Zur Genese eines kognitionswissenschaftlichen Paradigmas, Kognitions-wissenschaft (1991) 2, 1 (5).

## 3.2 Maschinen und Datenverarbeitung

### 3.2.1 Algorithmus und Maschine

Die Entwicklungen in der Informationstechnik bewirkten eine Änderung des Maschinenbegriffs. „Das Aufkommen aller Arten elektronischer Maschinen, insbesondere des elektronischen Computers, hat unsere Vorstellung von einer Maschine als Medium der Umwandlung und Übertragung von Kraft ersetzt durch das Bild eines Umwandlers von Information.“<sup>28</sup>

Maschinen wie die universelle Turingmaschine oder ein Digitalrechner lassen sich keinesfalls mehr unter den klassisch-mechanischen Maschinenbegriff einordnen, der sich an der Zweckbestimmung<sup>29</sup> und der Bau- und Funktionsweise der einzelnen Bauteile bzw. des gesamten Apparats orientiert. Orientiert man sich hingegen an dessen Verhalten, so ist es möglich, „die materiellen und immateriellen Bestandteile, zum Beispiel eines Computers, als Maschine, was er zweifellos ist“<sup>30</sup>, zu erfassen.

Der moderne kybernetische Maschinenbegriff abstrahiert von der technischen Lösung im Einzelfall und deren konkreten körperlichen Aufbau und stellt das Verhalten und die in der Maschine verwirklichte formale Struktur, den Algorithmus, in den Vordergrund.

Was eine bestimmte Maschine definiert, ist nicht ein bestimmter Bauteil oder eine spezielle Konstruktionsweise, sondern die Aufgabe, die sie ausführt.

Der Unterschied in den verschiedenen Betrachtungsweisen wird am Beispiel der Rechenmaschine<sup>31</sup> deutlich: Vor einigen Jahrzehnten wurden noch Rechenmaschinen verwendet, bei denen ein mechanisches Rechenwerk durch eine Handkurbel in Bewegung gesetzt wurde. Der nächste Schritt der technischen Entwicklung waren elektrische Maschinen, deren Mechanik durch einen Motor angetrieben wurde. Die heute verwendeten Maschinen führen dieselben Berechnungen unter Verwendung elektronischer Schaltwerke aus. Mechanische Bauteile finden dabei keine Verwendung mehr.

Eine ähnliche Entwicklung läßt sich bei der Betrachtung verschiedener Uhren, wie zB. Sonnenuhr, Sanduhr, mechanische Uhr und (elektronische) Digitaluhr erkennen.

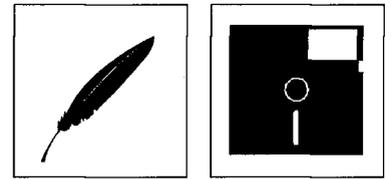
Bei einer klassischen Betrachtungsweise wären mechanische und elektronische Rechenmaschinen bzw. Uhren zwei völlig verschiedene Maschinen, da der Aufbau der beiden Apparate und die Funktion der einzelnen Bauteile keine Gemeinsamkeiten aufweisen.

Der kybernetische Maschinenbegriff hingegen geht davon aus, daß die Maschine nicht an eine bestimmte körperliche Form gebunden ist. Der Entwurf eines Algorithmus ist bereits die Konstruktion einer abstrakten Maschine.<sup>32</sup> Wird er festgelegt, so ist die Maschine definiert und kann auf verschiedene Weise körperlich hergestellt werden. Jeder Apparat, der die durch den Algorithmus beschriebene Funktion ausführt, ist eine Konkretisierung dieser abstrakten Maschine.

Es ist daher möglich, zu sagen: Der Algorithmus ist die Maschine.

Die einzelnen genannten Rechenmaschinen wenden denselben Algorithmus zur Durchführung von Berechnungen an. Sie verkörpern daher ein und dieselbe abstrakte Rechenmaschine und sind nur einzelne von vielen möglichen Realisierungsformen.<sup>33</sup>

Eine Maschine kann somit in verschiedenen materiellen Gestalten ausgebildet werden. Ebenso ist es aber auch möglich – wie beim Digitalrechner – mehrere Maschinen durch ein technisches Gerät zu realisieren. Dies erklärt auch die prinzipielle Gleichwertigkeit von Hardware und Programmen in der Informationstechnik. Es ist bei der Entwicklung von Geräten jederzeit möglich, Schaltwerke durch Programme zu ersetzen und umgekehrt<sup>34</sup>.



*Der neue Maschinenbegriff*

*Im Vordergrund: der Algorithmus ...*

*... der die Maschine definiert.*

*Ersetzung von Schaltwerken durch Programme und umgekehrt*

<sup>28</sup> Weizenbaum, Die Macht der Computer und die Ohnmacht der Vernunft (1978) 67f.

<sup>29</sup> Das Neue DUDEN-Lexikon (1984) definiert in diesem Sinn eine Maschine als „mechanische Vorrichtung, mit der entweder primär eine zur Verfügung stehende Energieform in eine andere, für einen bestimmten Zweck geeignete Energieform umgewandelt wird (Energie-M oder Kraft-M) oder mit der die von einer Kraft-M gelieferte Energie in gewünschte Arbeit umgesetzt wird (Arbeits-M)“ (Band 6, 2436).

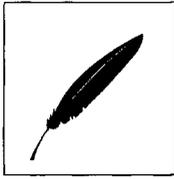
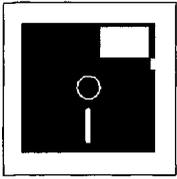
<sup>30</sup> Bammé et al., Maschinen-Menschen, Mensch-Maschinen (1986) 112.

<sup>31</sup> Bammé et al., a.a.O. 113 ff.

<sup>32</sup> Vgl. dazu Wirth: „Ich ziehe es vor, das Programmieren als das Konstruieren einer neuen Maschine aufzufassen, die mit Hilfe einer bestehenden Allzweckmaschine, eben dem Computer, realisiert wird. Die Neukonstruktion wird mittels der Bausteine ausgedrückt, welche die verwendete Notation zur Verfügung stellt, genauso wie ein Schaltkreis durch Symbole für gegebene Bauelemente und deren Verbindungen dargestellt wird.“ (Datenstrukturen und Algorithmen, Spektrum der Wissenschaft 11/1984, 46 ff.)

<sup>33</sup> Ashby, Einführung in die Kybernetik<sup>2</sup> (1985) 53

<sup>34</sup> Zemanek, Information und Ingenieurwissenschaft, in Folberth – Hackl (Hrsg.), Der Informationsbegriff in Technik und Wissenschaft (1986) 17 (43).



Abstrakte Maschine ...

... und Realisierungsformen

Notwendige Unterscheidung

Was sind Daten ...

... und was Informationen?

## „Laden eines Computerprogramms als Vervielfältigung?“

Egal welche Form gewählt wird, es bleibt dieselbe Maschine. „Die Frage des „wie“ hängt nur noch davon ab, was in der individuellen oder industriellen Realisierung als technisch günstiger erachtet wird. Es ist eine organisatorische Entscheidung der dafür Zuständigen, der Hersteller und Verwender technischer Informationssysteme<sup>35</sup>“.

Der Bau eines konkreten Apparates erfordert daher zwei Schritte: Der erste ist die detaillierte Festlegung und Beschreibung der Funktionalität der Maschine, die realisiert werden soll. Es wird die abstrakte Maschine entwickelt.

Nach dieser Bestimmung des Algorithmus folgt der zweite Schritt, die Realisierung der konkreten Maschine in einer der möglichen körperlichen Realisierungsformen. Erst wenn überhaupt bekannt ist, welche Maschine hergestellt werden soll, kann dieser Schritt gesetzt werden.

### 3.2.2 Daten und Information

Bevor aus dem Modell der Turingmaschine Schlüsse über real existierende und eingesetzte Computer gezogen werden, ist noch eine weitere grundlegende Begriffsbestimmung durchzuführen: Es ist dies die Unterscheidung von Daten und Information.

Vielfach werden Daten und Information gleichgesetzt oder wird Information als Oberbegriff für Daten und Information verwendet.

So wird dann auch in diesem Sinne von Informationsträgern<sup>36</sup>, gespeicherter Information und deren Wiedergabe oder ganz allgemein von der Verarbeitung von Information<sup>37</sup> gesprochen.

Es ist jedoch schon alleine um Unklarheiten zu vermeiden sinnvoll, Daten und Information zu unterscheiden.

Eine allgemein anerkannte, umfassende und zufriedenstellende<sup>38</sup> Definition von „Daten“ ist – wie oft bei grundlegenden Begriffen – nicht bekannt. Es ist aber für die Abgrenzung zu Information ausreichend, Daten als Folge von Zeichen oder Symbolen zu definieren. Solche Zeichen können zB. Buchstaben, Zahlen, magnetische Zustände und vieles mehr sein.

Zur Darstellung und Übertragung von Daten notwendig ist ein Datenträger. Ein solcher kann ein mit Zeichen beschriebenes Blatt Papier genauso sein wie modulierte elektromagnetische Wellen oder Disketten, Festplatten bzw. Magnetbänder, bei denen magnetische Zustände zur Informationsdarstellung verwendet werden.

Wesentliche Eigenschaften von Daten sind ihre beliebige Vervielfältigbarkeit und ihre Unkörperlichkeit.

Im Gegensatz dazu muß Information jeweils aus Daten neu erstellt werden<sup>39</sup>, wobei es wesentlich auf die Rahmenbedingungen und Ziele dabei ankommt. Aus ein und demselben Satz von Daten kann bei unterschiedlicher Interessenslage völlig andere Information gewonnen werden bzw. kann die aus denselben Daten gewonnene Information für den Einen wertvoll sein und für den Anderen uninteressant.

Da daher „der Kontext der Handlungssituation für die Spezifität und die Qualität der Information bestimmend ist, hat Information keinen quasi objektiven Charakter, sondern variiert nach den wechselnden Anforderungen und Rahmenbedingungen. ... Informationen können nicht wie Daten *verarbeitet* werden, sondern werden erst unter Berücksichtigung konkreter Bedürfnisse und Handlungssituationen *erarbeitet*. Information ist – wenn man es in eine Formel packen wollte – *Wissen in Aktion*“<sup>40</sup>.

Information läßt sich somit als interpretierte Daten beschreiben und darf nicht vom Interpretierenden losgelöst betrachtet werden. Sie kann daher nicht übertragen oder sonstwie weitergegeben werden. Da sie mit dem Interpretierenden eine Einheit bildet, kann sie auch nicht vervielfältigt werden.

Auch wird die Gesamtheit Interpretierender/Information wohl als körperlich anzusehen sein.

<sup>35</sup> Schober – Lenk, Informationstechnik (1989) 139.

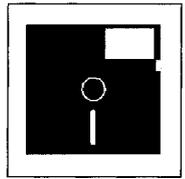
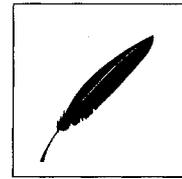
<sup>36</sup> Schober – Lenk, a.a.O., 18.

<sup>37</sup> Vgl. statt vieler Wiebe, a.a.O., 136.

<sup>38</sup> So zB. falsch neue DUDEN-Lexikon, das Daten in der Datenverarbeitung als maschinell verarbeitete Informationen definiert (Das neue DUDEN-Lexikon (1984) Band 2, 725).

<sup>39</sup> Kublen, Hypertext (1991), 62.

<sup>40</sup> Kublen, a.a.O., 63 (Hervorhebungen im Original).



Wird versucht Information weiterzugeben, so kann dies nur über den Umweg der Datenübermittlung geschehen. Aus den Daten kann wieder Information erarbeitet werden, die zwar – da Information ja kontextabhängig ist – nicht dieselbe ist, aber doch für den neuen Interpreten ähnliche oder dieselbe Bedeutung haben kann<sup>41</sup>.

### 3.2.3 Hardware und Software

Nach diesen theoretischen Ausführungen stellt sich die Frage, was sich daraus für die Beurteilung von Hardware und Software und deren Zusammenspiel gewinnen läßt.

Moderne Digitalcomputer sind im Prinzip nichts anderes als universelle Turingmaschinen. Die heute allgemein verwendete „von Neumann“-Computerarchitektur (zB. jeder PC) kann als eine technische Realisierung der universellen Turingmaschine angesehen werden. Die CPU entspricht dabei der Schreib/Lese-Einrichtung und dem Apparat, der eine endliche Anzahl von internen Zuständen einnehmen kann, der Speicher (RAM, ROM, Massenspeicher) dem Band<sup>42</sup>. Es ist daher durchaus zulässig, ja sogar geboten, die aus dem Modell der Turingmaschine gewonnenen Erkenntnisse auch auf Computer anzuwenden<sup>43</sup>.

Die technische Realisierung solcher Universalmaschinen „hat die wichtige Konsequenz, daß es, von Geschwindigkeitserwägungen abgesehen, unnötig ist, immer neue Maschinen für unterschiedliche Rechenprozesse zu entwickeln. Sie können allesamt mit einem Digitalrechner durchgeführt werden, der für jeden Fall geeignet zu programmieren ist“<sup>44</sup>.

Die Unterschiede in der Hardware verschiedener Universalcomputer lassen sich daher vollständig durch Programme kompensieren und verlieren an Bedeutung. Der Programmierer oder theoretisch orientierte Informatiker sieht daher „den Computer nicht als eine nach physikalischen Gesetzmäßigkeiten konstruierte Maschine, mit der er oder sie interagiert, sondern als Abstraktion – als einen Formalismus zur Beschreibung von Verhaltensmustern“<sup>45</sup>.

Dies ist eine der Ursachen für die weit verbreitete Sichtweise, „die Hardware als relativ (vielleicht sogar als ganz) unwichtig und die Software, also das Programm oder den Algorithmus, als den wirklich wesentlichen Bestandteil“<sup>46</sup> zu betrachten.

Diese Sicht ist letztlich auch dafür verantwortlich, daß das Programm und die Hardware während des Programmablaufs nicht als Einheit gesehen werden, sondern weiterhin als zwei getrennte Einheiten, zwischen denen es Wechselwirkungen gibt. Eine (auch rechtliche) Unterscheidung zwischen einem Programm, das auf einem Datenträger gespeichert ist, und dem Programmablauf erscheint daher nach hL. nicht angebracht.

*Der Digitalcomputer als universelle Turingmaschine*

*Realisierung durch Programmierung*

*Die Hardware nichts, die Software alles?*

<sup>41</sup> Dies ist der Grund, warum es nicht möglich ist, Information, der Zugang dazu und ihre Verbreitung rechtlich zu schützen. Alle diesbezüglichen Versuche müssen notwendigerweise an Daten, dem Zugang dazu und ihrer Verbreitung anknüpfen.

<sup>42</sup> Bolter, Turing's Man (1984) 42 ff; Coy, Aus der Vorgeschichte des Mediums Computer, in Bolz – Kittler – Tholen (Hrsg.), Computer als Medium (1994), 19.

Vgl. auch Davis, Mathematical Logic and the Origin of Modern Computers, in Herken (Hrsg.), The Universal Turing Machine A Half-Century Survey (1988), 166: The computers of the postwar period „were conceived, designed, and constructed, not as mere automatic calculators, but as *engines of logic*, incorporating the general notion of what it means to be computable and embodying a physical model of Turing's universal machine.“ (Hervorhebung im Original)

<sup>43</sup> Für die Anwendung der grundlegenden Theorie der Universalmaschinen, wie sie durch Turing dargestellt wurde, spricht auch, daß sie eine von Hardwaredetails unabhängige Betrachtung des Problemkreises Hardware/Software ermöglicht. Dies ist auch für die juristische Problembearbeitung entscheidend, da nur eine von den Zufälligkeiten technisch-konstruktiver Details losgelöste Beurteilung nicht mit jeder Änderung von Speicher- oder Programmgrößen bzw. anderen Hardwaremodifikationen zu neuen, abweichenden rechtlichen Schlußfolgerungen und damit zu angemessener Rechtssicherheit führt.

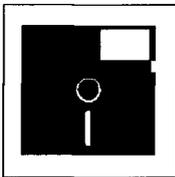
Eine solche Sichtweise ist auch dem Fachgebiet der Informatik angemessener, sind doch „die physischen Teile in einem Computer ... Ausdruck der – vor allem technischen – Bedingungen, unter denen sich ein Entwicklungsprozeß vollzieht. Die meisten Computer bestehen aus elektrischen Schaltern, aber das muß nicht so sein. Im Museum der Digital Equipment Corporation in Maynard, Massachusetts, steht ein Computer, der nur aus Baukastenteilen zusammengesetzt ist. Mit seinen vielen hölzernen Spulen und Dübeln könnte er ein kleines Zimmer füllen. Er wurde am MIT zu dem Zweck gebaut, anschaulich zu demonstrieren, daß es bei Computern auf die Fähigkeit der Maschine ankommt, einem Prozeß Gestalt zu geben, eine Sequenz von Regeln zu spezifizieren.“ (Turkle, Die Wunschmaschine (1986), 338.)

Warum sollte sich der Jurist daher mit RAM, ROM und anderen konstruktiven Details beschäftigen, wenn diese gar nicht das wesentliche sind?

<sup>44</sup> Kublen, a.a.O., 63 (Hervorhebungen im Original).

<sup>45</sup> Winograd – Flores, a.a.O., 147.

<sup>46</sup> Penrose, Computerdenken (1991) 22.



Die zwei Seiten des  
Computerprogramms

Formale Sprache, Compiler,  
Objektprogramm

Neue Qualität durch  
Programmablauf

Das Türingsche Modell legt aber eine andere Sichtweise nahe:

Ein Computerprogramm ist die Beschreibung eines Algorithmus in einer formalen Sprache<sup>47</sup>, der Programmiersprache. Dieser Text wird auch als Quellen- oder Sourceprogramm bezeichnet und ist nichts anderes als eine von vielen möglichen Darstellungsformen einer abstrakten Maschine<sup>48</sup>, die ja – wie oben dargelegt – mit dem Algorithmus gleichzusetzen ist.

Das Verhältnis von Computerprogramm als Beschreibung einer abstrakten Maschine zum realisierten Apparat ähnelt der Beziehung einer Konstruktionszeichnung<sup>49</sup> oder eines Plans<sup>50</sup> zum danach hergestellten Werkstück.

Ein Computerprogramm kann daher sowohl als Ausgangspunkt für die Realisierung eines konkreten, speziellen Apparates dienen, als auch Grundlage für die Imitation eines solchen durch einen Universalrechner sein. Welcher der beiden technischen Lösungen der Vorzug gegeben wird, hängt – wie bereits dargestellt – vom Einzelfall ab. Grundsätzlich sind sie beide gleichwertig.

Und tatsächlich sind auch Maschinenbeschreibungen in formalen Sprachen – also Computerprogramme – Ausgangspunkt für die Fertigung von Hardwarelösungen. So etwa bei der Verwendung von Hardwarebeschreibungssprachen (zB. VHDL Hardware Description Language [VHDL]), mit denen digitale Schaltungen und Systeme von der Systemdefinition bis zum Schaltungsentwurf beschrieben werden können. Die Philosophie hinter solchen Sprachen ist ähnlich der vieler moderner Programmiersprachen<sup>51</sup>. Bei der Beschreibung von Hardware in einer solchen Sprache muß ebenso keine Rücksicht auf Technologien oder Details der Implementierung genommen werden, wie die Darstellung eines Algorithmus in einer höheren Programmiersprache von der verwendeten Hardware unabhängig ist. Aus einer solchen Hochsprachenbeschreibung können auch automatisch Gatternetzlisten erzeugt werden<sup>52</sup>. Die Darstellung digitaler Schaltungen und Systeme in einer Hochsprache steht in einem ähnlichen Verhältnis zur daraus erzeugten Gatternetzliste wie das Quellenprogramm zum Maschinencode.

Ähnliches ist auch bei automatisierter Fertigung (CAM, CIM) der Fall, wenn die herzustellenden Werkstücke und Geräte nur mehr durch Programme für verschiedene rechnergesteuerte Werkzeuge beschrieben werden.

Der große Vorteil der Beschreibung in einer formalen Sprache ist, daß es sich dabei um eine für den Menschen verständliche und übersichtliche Form der Darstellung handelt, die aber gleichzeitig durch eine Übersetzungsmaschine – einen Compiler oder Interpreter, der Programme nur als Ein- bzw. Ausgabedaten behandelt – bearbeitet und in eine von einem Universalrechner verarbeitbare Form umgewandelt werden kann. Das Ergebnis dieses Vorgangs ist wiederum ein Text und wird Objektprogramm oder Maschinencode genannt.

Bei allen diesen Formen des Programmes handelt es sich nur um Daten und nicht um Information im Sinne der oben dargelegten Unterscheidung. Dies zeigt sich auch daran, daß Programme beliebig vervielfältigt werden können und auch zB. für Kompressionsprogramme oder Computerviren als Ein- und Ausgabedaten dienen.

Der Maschinencode erhält aber eine neue Qualität, wenn das Computerprogramm abläuft. Dann wird die durch das Computerprogramm beschriebene abstrakte Maschine unter Verwendung eines Universalrechners dadurch realisiert, daß dieser sich entsprechend der Beschreibung verhält und sie somit imitiert.

<sup>47</sup> Winograd – Flores, a.a.O. 147.

Bei formalen Sprachen handelt es sich nicht um Begriffssprachen (Schober – Lenk, a.a.O.: 36 ff.) mit Semantik, wie die menschliche Sprache, sondern nur um ein „Regelwerk, das in sich hinreichend konsistent und operational vollständig ist, um für alle Informationssysteme eines Typs ... das erforderliche Quantum an Gesetzmäßigkeit und Ordnungsmäßigkeit im Umgehen mit Information zu gewähren“ (Schober – Lenk, a.a.O. 30). Haugeland spricht daher auch von „Sprachen“ bzw. „Programmiersprachen“ als virtuellen Maschinen bzw. Maschinen höherer Ebene (Künstliche Intelligenz – Programmierte Vernunft? (1987) 127).

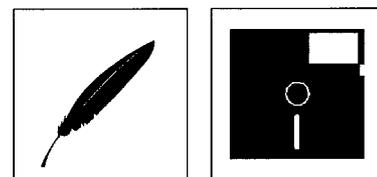
<sup>48</sup> Steinmüller, Programme als abstrakte Problemlösungsmaschinen, in Kühne, Software und Recht (1986) 47 (50). Vgl. auch Rheinwald, die zwischen (nicht semantischen) abstrakten Computerprogrammen und konkreten Computerprogrammen, dh. in einer konkreten Maschine realisierten Programmen, mit abgestufter Semantik, unterscheidet (Können Maschinen eine Sprache sprechen?, Kognitionswissenschaft 1/1991 37 (45 f.)).

<sup>49</sup> Ertl – Wolf, a.a.O., 33.

<sup>50</sup> Holzinger, Produkthaftpflicht und Software, EDV & Recht 4/88, 10 (11).

<sup>51</sup> Lipsitt – Schaeffer – Ussery, VHDL: Hardware Description and Design (1990), 5.

<sup>52</sup> Wagner, Einführung in VHDL, Design & Elektronik 13/92, 58.



Der Maschinencode ist in diesem Fall „das informationelle Baumaterial“<sup>53</sup> des Computers und bestimmt sein Verhalten. Unter Computer ist aber nicht die Hardware alleine, sondern ein technisches Informationssystem bestehend aus Hardware, Betriebssystem und Anwendersoftware zu verstehen<sup>54</sup>, die Einheit Hardware/Software<sup>55</sup>.

Für die Qualifikation als informationelles Baumaterial eines Computers ist es unerheblich, in welchem Speicher des Systems (Massen-, Arbeits- oder Cache-Speicher) das ablaufende Computerprogramm oder Teile davon gespeichert sind, ob es sich um ein technisches oder untechnisches Programm bzw. um ein Standard- oder ein individuell erstelltes Programm handelt. Entscheidend ist, daß die Universalmaschine die in einer formalen Sprache erstellte Beschreibung einer Maschine – das Programm – nicht als Ein- oder Ausgabedaten behandelt, sondern diese so interpretiert, daß sie Eingabedaten genauso bearbeitet wie die im Computerprogramm dargestellte Maschine.

Beim Programmablauf verändern daher sowohl der Rechner als auch das Programm ihren Charakter:

- Der Computer hört auf, eine Universalmaschine zu sein. Die Einheit Hardware/Software ist eine Realisierungsform der im Programm beschriebenen abstrakten Maschine, also eine konkrete Spezialmaschine<sup>56</sup>
- In der Einheit Hardware/Software wird das Programm durch den Rechner interpretiert. Es ist daher als Information zu qualifizieren und nicht mehr nur als Datum. Daher kann auch im Zusammenhang mit dem Programmablauf nicht mehr von einer Vervielfältigung von Daten gesprochen werden, da keine Daten, sondern Information vorliegt und diese wegen ihrer Einheit mit den Interpretierenden nicht vervielfältigt werden kann<sup>57</sup>.

Die Folge des Programmablaufs ist somit, daß dadurch ein Spezialapparat ein informationsverarbeitendes System entsteht. Dieses ist etwas anderes als nur die Summe der einzelnen Teile des Systems, von Hardware und Software, Programm und Universalmaschine. Diese bilden eine funktionelle technische Einheit<sup>58</sup>. Eine Trennung der Systemteile ist nicht möglich, ohne die realisierte Maschine zu zerstören. Es muß daher auch immer das gesamte technische Informationssystem betrachtet werden.

Die Folge einer Beendigung des Programmablaufs ist, daß die Systembestandteile wieder die ursprüngliche Bedeutung der Beschreibung einer Maschine und eines Universalrechners – einer universellen Turingmaschine – erhalten.

Bei der Beurteilung der Eigenschaften von Computerprogrammen sind somit zwei unterschiedliche Dinge, die beide gleich benannt werden, zu unterscheiden: die Beschreibung einer abstrakten Maschine in einer formalen Sprache und das ablaufende Programm als informationelles Baumaterial eines Computers.

(wird fortgesetzt)

*Informationelles Baumaterial*

*Von der Universalmaschine zur konkreten Spezialmaschine ...*

*... und vom Datum zur Information*

*Die Folge des Programmablaufs ...*

*... und die seiner Beendigung*

<sup>53</sup> Schober – Lenk, a.a.O., 136 ff.

<sup>54</sup> Schober – Lenk, a.a.O., 83.

<sup>55</sup> Wiebe, a.a.O., 239.

<sup>56</sup> Vgl. dazu auch Schnupp – Floyd, Software<sup>2</sup> (1979) 16 ff., die zwischen der Basismaschine – der Hardware – und der Benutzermaschine unterscheiden. Erst das Programm transformiert die Basismaschine in die Benutzermaschine. „Es ist verständlich, daß diese „nur“ durch ein Programm realisierte Maschine oft „weniger wirklich“ als die Basismaschine empfunden wird. Es hat sich deshalb eingebürgert, die Hardware als reale Maschine zu bezeichnen – im Gegensatz zu einer virtuellen („scheinbaren“) Maschine, die erst durch Software, durch ein oder mehrere Programme, auf der realen simuliert wird. Für den außenstehenden Benutzer ist die virtuelle Benutzermaschine jedoch ebenso „real“ wie die Hardware – er kann nicht unterscheiden, welche Funktionen unmittelbar von der Hardware oder erst mittelbar über ein Programm ausgeführt werden.“

„Der Vorteil der freien Programmierbarkeit ist, daß durch Auswechseln des Programms die gleiche Basismaschine in viele unterschiedliche Benutzermaschinen transformiert werden kann.“ (Hervorhebungen im Original). Es ist daher der oben angeführten Meinung, daß eine Datenverarbeitungsanlage durch den Programmablauf nicht verändert wird, entgegenzutreten.

<sup>57</sup> Da Programmcode sowohl als Datum als auch als Information im Computer eingesetzt wird, kann es auch sein, daß ein und dasselbe Programm sowohl als Datum als auch als informationelles Baumaterial verwendet wird. Dies ist zB. bei Computerviren der Fall, ja sogar unabdingbare Voraussetzung für ihre Existenz, da ein Virus eine Kopie von seinem eigenen Programmcode erzeugt und somit seinen eigenen Code als Eingabe- und Ausgabedaten benutzt (Neumann, Computerviren – Grundlagen, Entdeckung und Abwehr, in Paul (Hrsg.), GI – 19. Jahrestagung I, Proceedings (1989), 523).

<sup>58</sup> Troller, Der urheberrechtliche Schutz von Inhalt und Form der Computerprogramme, CR 4/1987 213 und 278 (281).