Werner Brecht

Theoretische Informatik

Rechnerarchitektur

von John L. Hennessy and David A. Patterson

Aufbau und Arbeitsweise von Rechenanlagen

von Wolfgang Coy

Mehr als nur Programmieren ...

Eine Einführung in die Informatik

von Rainer Gmehlich und Heinrich Rust

Interaktive Systeme

Software-Entwicklung und Software-Ergonomie
von Christian Stary

Modernes Software Engineering

von Reiner Dumke

Management von Softwareprojekten

von Fritz Peter Elzer

Parallele Programmierung

von Thomas Bräunl

Konzepte und Praxis des Compilerbaus

von Volker Penner

Theoretische Informatik

von Werner Brecht

Formalisieren und Beweisen

von Dirk Siefkes

UNIX

von Werner Brecht

Verteilte Systeme unter UNIX

von Werner Brecht

Werner Brecht

Theoretische Informatik

Grundlagen und praktische Anwendungen

ISBN-13:978-3-528-05462-5
DOI: 10.1007/978-3-322-86827-5

e-ISBN-13:978-3-322-86827-5

Das in diesem Buch enthaltene Programm-Material ist mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Der Autor und der Verlag übernehmen infolgedessen keine Verantwortung und werden keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieses Programm-Materials oder Teilen davon entsteht.

Alle Rechte vorbehalten

© Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig/Wiesbaden, 1995

Der Verlag Vieweg ist ein Unternehmen der Bertelsmann Fachinformation GmbH.

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlags unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Umschlaggestaltung: Klaus Birk, Wiesbaden

Druck und buchbinderische Verarbeitung: Langelüddecke, Braunschweig

Gedruckt auf säurefreiem Papier

Inhaltsverzeichnis

Vorwort	VII
1 Mathematische Grundlagen	1
1.1 Mengen	1
1.2 Strukturen	10
1.3 Übungen	18
2 Algorithmen	19
2.1 Turing-Maschinen	19
2.2 Produktionssysteme	48
2.3 Formulierung von Algorithmen	58
2.4 Übungen	74
3 Berechen- und Entscheidbarkeit	75
3.1 Berechenbarkeit	75
3.2 Entscheidbarkeit	90
3.3 μ -rekursive Funktionen	95
3.4 These von Church	101
3.5 Grenzen algorithmischer Lösbarkeit	103
3.6 Übungen	114

4	Komplexität	115
4.1	Zeit- und Speicherkomplexität	115
4.2	Sortieralgorithmen	127
4.3	Übungen	140
5	Nebenläufigkeit	141
5.1	Nichtdeterminismus	141
5.2	Nebenläufige Prozesse	154
5.3	Übungen	168
6	Selbstmodifizierende und selbstreproduzierende Algorithmen	169
6.1	Selbstbezüglichkeit	169
6.2	Computerviren: Grundbegriffe	175
6.3	Arbeitsweise von Computerviren	180
6.4	Umgang mit Computerviren	200
6.5	Übungen	216
	Lösungen zu den Übungen	217
	Abbildungen	231
	Literaturverzeichnis	233
	Register	239

Vorwort

Die theoretische Informatik hat ihre Wurzeln in dem mathematisch-philosophischen Bereich der prinzipiellen Berechenbarkeit mathematischer Funktionen. Grundlegende Arbeiten sind im wesentlichen in den dreißiger Jahren dieses Jahrhunderts geleistet worden, also vor dem Erscheinen der ersten Rechenanlagen und lange bevor die Informatik den Rang eines eigenständigen Forschungs- und Lehrgebiets bekommen hat. Das mag einer der Gründe dafür sein, daß die theoretische Informatik in der Informatikausbildung meist der Mathematik und weniger den praktischen Anwendungen zugeordnet wird. Das schlägt sich in den Informatikstudiengängen an den Hochschulen entsprechend nieder. Im Bereich der (inner-) betrieblichen Aus- und Weiterbildung wird die theoretische Informatik fast vollständig ignoriert.

Eine stärkere Hinwendung zu den Anwendungsgebieten kann zum einen für die mathematisch orientierte theoretische Informatik eine sinnvolle Ergänzung sein, zum anderen können dadurch prinzipielle Gegebenheiten der Anwendungsgebiete auf einige wenige fundamentale Konzepte zurückgeführt werden. Dazu möchte das vorliegende Buch einen kleinen und bescheidenen Beitrag leisten. Es ist aus einem Vorlesungsmanuskript für Lehrveranstaltungen an der Technischen Fachhochschule in Berlin hervorgegangen. Der Autor, ein Informatiker und Mathematiker, ist dort Professor für Betriebssysteme und Systemprogrammierung und verfügt über eine langjährige Erfahrung als Rechenzentrumsleiter. Die Lehrveranstaltungen wurden Studierenden des Hauptstudiums der Studiengänge Allgemeine Informatik, Technische Informatik, Ingenieurinformatik und Mathematik angeboten.

Die Resonanz auf diese Angebote war ermutigend. Zum einen ist aus dem Kreis der Kolleginnen und Kollegen Zustimmung geäußert und auf einen grundsätzlichen Bedarf an theoretischer Informatik zur Unterstützung der eigenen Veranstaltungen hingewiesen worden. Zum anderen hat das Inter-

esse der (zahlreichen) Studierenden zu stellenweise sehr tiefgehenden Diskussionen geführt. Beides hat das Buch inhaltlich mitgestaltet. Es beginnt mit einer kurzen Zusammenstellung des notwendigen mathematischen Werkzeugs, das im Bereich der *Mengen und Strukturen* angesiedelt ist. Danach wird der *Algorithmusbegriff* mit Hilfe von Turing-Maschinen und den ihnen äquivalenten Produktionssystemen eingeführt. Die Behandlung der Turing-Maschinen schließt mit dem Bau einer Experimentiermaschine, die das fundamentale Algorithmenkonzept anfaßbar machen soll.

Eine solche Experimentiermaschine ist prinzipiell in jeder Programmiersprache realisierbar, und interessierte Leser(innen) sind aufgerufen, eine entsprechende Implementierung vorzunehmen. Um möglichst schnell zu einem funktionsfähigen Prototypen zu kommen, ist für das Buch eine Realisierung mit UNIX-Kommandoprozeduren, sogenannten Shell-Scripts, gewählt worden. Ziel bei dieser Realisierungsform war nicht nur eine schnelle Anwendungsentwicklung, sondern auch das Vorhaben, in dem Anwendungsgebiet, das sich mit Nebenläufigkeit beschäftigt, die Multitaskingfähigkeit des UNIX-Betriebssystems auszunutzen. Die Experimentiermaschine kann mit wenigen Befehlen so erweitert werden, daß mit ihr auch praktische Experimente mit nebenläufig arbeitenden Programmen durchgeführt werden können.

Damit ist bereits zu den Anwendungskapiteln übergeleitet worden. Vorher jedoch und für das Buch zentral wird die *Berechenbarkeit* mathematischer Funktionen und die Grenzen dieser (algorithmischen) Berechenbarkeit beschrieben. Dazu wird an dafür geeigneten Stellen eine höhere Programmiersprache als Formulierungshilfe für die algorithmischen Betrachtungen benutzt. Dies ist in der theoretischen Informatik durchaus üblich, und meist bedient man sich Pascal-ähnlicher Sprachen. Wegen der großen Verbreitung der Programmiersprache C und auch (das war der Hauptgrund) um den Anwendungsbezug hervorzuheben, wird als höhere Sprache ein C-ähnlicher Pseudocode verwendet. Die Beschränkung auf ganz wenige und intuitiv verständliche Sprachkonstrukte bewirkt, daß auch Leser(innen) ohne C-Kenntnisse die Programmbeispiele problemlos nachvollziehen können.

An die Berechenbarkeitsüberlegungen schließen sich beispielhaft drei Anwendungsgebiete an. Das erste und für die theoretische Informatik traditionelle Gebiet *Komplexität* hat eine sehr tiefgehende mathematische

Dimension, auf deren Beschreibung zugunsten praktischer Aufwandsanalysen konkreter Algorithmen verzichtet worden ist. Statt dessen ist versucht worden, das zugehörige theoretische Konzept praktisch greifbar zu machen und auf bekannte Sortierverfahren anzuwenden.

Im dem Kapitel über *Nebenläufigkeit* werden unter Zugrundelegung der Turing-Maschinen und nichtdeterministischer Programme grundlegende Konzepte der sogenannten *Parallelprogrammierung* aufgezeigt. Dabei werden Prozesse untersucht, die auf einer (einzigen) Maschine beziehungsweise in einem Netzwerk nebenläufig durchgeführt werden. Unter anderem wird dargestellt, welche Erweiterungen an den üblichen Programmiersprachen vorgenommen werden müßten, um mit ihnen nebenläufige Prozesse formulieren zu können.

Das letzte Kapitel schließlich befaßt sich mit der Arbeitsweise *selbstmodifizierender und selbstreproduzierender Algorithmen*. Anwendungen dieses Gebiets sind unter dem Schlagwort *Computerviren* bekannt. Es wird gezeigt, wie ein grundlegendes, auf dem Konzept der Turing-Maschinen beruhendes Verständnis dieser Algorithmen den Umgang mit ihnen praktikabel werden läßt. Das wird soweit getrieben, daß plakativ eine Handlungsanleitung für den Fall einer Virusinfektion angegeben wird, die durch Empfehlungen abgerundet ist, die dazu dienen, einem Befall durch Viren vorzubeugen.

Zu diesem Buch haben nicht nur die vielen Studierenden mit ihren Diskussionsbeiträgen beigetragen, sondern insbesondere meine beiden Kollegen an der Technischen Fachhochschule Berlin, Prof. Dr. Ulrich Grude und Prof. Dr. Andreas Solymosi, die das Manuskript gelesen und konstruktiv kritisiert haben. Beiden verdanke ich viele Anregungen. Besonders zu erwähnen ist in diesem Zusammenhang auch Hans-Christian Brecht, der als fortgeschrittener Student der Physik und Informatik Testleser war und viele didaktische Ungeschicklichkeiten entdeckt hat. Ihnen allen gebührt mein Dank. Danken möchte ich auch dem Vieweg-Verlag für die Möglichkeit der Realisierung dieses Buchprojekts und für die unbürokratische Begleitung.