

# Springer-Lehrbuch

---

**Springer-Verlag Berlin Heidelberg GmbH**

Gerhard Goos

# Vorlesungen über Informatik

Band 4: Paralleles Rechnen und  
nicht-analytische Lösungsverfahren

Mit 115 Abbildungen  
und 12 Tabellen



Springer

Prof. Dr. Gerhard Goos  
Institut für Programmstrukturen  
und Datenorganisation  
Universität Karlsruhe  
Postfach 6980  
D-76128 Karlsruhe

Die Deutsche Bibliothek – CIP-Einheitsaufnahme

**Goos, Gerhard:**

Vorlesungen über Informatik / Gerhard Goos. – Berlin; Heidelberg; New York; Barcelona; Budapest; Hongkong; London; Mailand; Paris; Santa Clara; Singapur; Tokio: Springer (Springer-Lehrbuch)

Bd. 4. Paralleles Rechnen und nicht-analytische Lösungsverfahren. – 1998

ISBN 978-3-540-60650-5 ISBN 978-3-642-58821-1 (eBook)

DOI 10.1007/978-3-642-58821-1

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland vom 9. September 1965 in der jeweils geltenden Fassung zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtsgesetzes.

© Springer-Verlag Berlin Heidelberg 1998

Ursprünglich erschienen bei Springer-Verlag Berlin Heidelberg New York 1998

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, daß solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Satz: Reproduktionsfertige Autorenvorlage

Umschlaggestaltung: design & production GmbH, Heidelberg

SPIN: 10525604 33/3142-543210 – Gedruckt auf säurefreiem Papier

# Inhaltsverzeichnis

<b>Vorwort</b>	<b>ix</b>
<b>19 Paralleles Rechnen</b>	<b>1</b>
19.1 Parallele Rechenmodelle . . . . .	2
19.1.1 Die parallele Registermaschine . . . . .	2
19.1.2 Datenflußmodell . . . . .	11
19.1.3 Abstrakte Netzwerkmodelle . . . . .	20
19.2 Technische Modelle . . . . .	31
19.2.1 Klassifikation von Parallelrechnern . . . . .	32
19.2.2 Vektorrechner . . . . .	33
19.2.3 Gemeinsamer Speicher . . . . .	35
19.2.4 Verbindungsnetzwerke . . . . .	36
19.2.5 Abbildung von PRAM- auf LogP-Algorithmen . . . . .	44
19.2.6 PVM und MPI . . . . .	51
19.3 Entwurfstechniken für parallele Algorithmen . . . . .	52
19.3.1 Die Komplexitätsklasse $NC$ . . . . .	53
19.3.2 Paralleles Teile-und-Herrsche . . . . .	55
19.3.3 Ausgewogene Bäume . . . . .	61
19.3.4 Präfixgraphen . . . . .	64
19.3.5 Zeigerspringen . . . . .	66
19.3.6 Die Euler-Tour . . . . .	70
19.4 Anmerkungen und Verweise . . . . .	74
<b>20 Zellularautomaten</b>	<b>75</b>
20.1 Grundlagen . . . . .	76
20.2 LIFE: Das Spiel des Lebens . . . . .	81
20.3 Turing-Mächtigkeit . . . . .	86
20.4 Anwendungen . . . . .	91
20.4.1 Ein Synchronisierungsproblem . . . . .	92
20.4.2 Modelle aus Differentialgleichungen . . . . .	95
20.4.3 Gittergase . . . . .	97
20.4.4 Verkehrssimulation . . . . .	100
20.5 Anmerkungen und Verweise . . . . .	102

<b>21 Künstliche neuronale Netze</b>	<b>105</b>
21.1 Neuronen . . . . .	107
21.1.1 Das biologische Vorbild . . . . .	107
21.1.2 Künstliche Neuronen . . . . .	109
21.2 Zur Konstruktion von KNNs . . . . .	113
21.2.1 Einsatzbedingungen . . . . .	113
21.2.2 Lernen statt Programmieren . . . . .	115
21.3 Vorwärtsgerichtete Netze . . . . .	118
21.3.1 Das einfache Perzeptron . . . . .	119
21.3.2 Das mehrschichtige Perzeptron . . . . .	125
21.4 Unüberwachtes Lernen . . . . .	137
21.4.1 Wettlernen . . . . .	137
21.4.2 Topologie-erhaltende Karten . . . . .	141
21.5 Hopfield-Netze . . . . .	146
21.5.1 Synchrone Hopfield-Netze . . . . .	146
21.5.2 Asynchrone Hopfield-Netze . . . . .	150
21.6 Anwendungen . . . . .	160
21.7 Anmerkungen und Verweise . . . . .	163
<b>22 Zufallsgesteuerte Optimierung</b>	<b>165</b>
22.1 Optimierungsaufgaben . . . . .	166
22.1.1 Schrittweise Optimierung . . . . .	168
22.1.2 Gradientenverfahren . . . . .	171
22.2 Stochastische Optimierung . . . . .	174
22.2.1 Simuliertes Tempern . . . . .	175
22.2.2 Tempern mit deterministischer Akzeptanz . . . . .	178
22.3 Evolutionäre Algorithmen . . . . .	182
22.3.1 Evolutionsstrategien . . . . .	185
22.3.2 Genetische Algorithmen . . . . .	191
22.3.3 Evolutionsstrategien und genetische Algorithmen im Vergleich . . . . .	198
22.4 Anmerkungen und Verweise . . . . .	199

<b>23 Unschärfe Informationsverarbeitung</b>	<b>201</b>
23.1 Unschärfe Mengen . . . . .	206
23.1.1 Grundoperationen auf unscharfen Mengen . . . . .	210
23.1.2 Allgemeinere unscharfe Mengen . . . . .	218
23.2 Unschärfe Relationen . . . . .	220
23.2.1 Max-Min Komposition und Erweiterungsprinzip . . . . .	224
23.2.2 Relationengleichungen . . . . .	228
23.3 Unschärfe Regelung . . . . .	229
23.3.1 Entwurf unscharfer Regler . . . . .	238
23.4 Unschärfe Maße . . . . .	247
23.4.1 Maßbasen . . . . .	250
23.4.2 Glaubwürdigkeits- und Plausibilitätsmaße . . . . .	251
23.4.3 Möglichkeits- und Notwendigkeitsmaße . . . . .	253
23.4.4 Klassifikation unscharfer Maße . . . . .	254
23.4.5 Möglichkeitsmaße und unscharfe Mengen . . . . .	255
23.4.6 Unschärfe Maße in der Anwendung . . . . .	256
23.5 Anmerkungen und Verweise . . . . .	259
<b>Literaturverzeichnis</b>	<b>261</b>
<b>D Fourierreihen und Fouriertransformation</b>	<b>269</b>
D.1 Fourierreihen . . . . .	269
D.2 Fouriertransformation . . . . .	272
D.2.1 Eigenschaften von Fouriertransformierten und Fourierreihen . . . . .	275
D.2.2 Das Abtasttheorem . . . . .	277
D.3 Diskrete Fouriertransformation . . . . .	279
D.3.1 Polynommultiplikation . . . . .	280
D.3.2 Schnelle Fouriertransformation . . . . .	281
D.4 Anmerkungen und Verweise . . . . .	282
<b>Programmverzeichnis</b>	<b>283</b>
<b>Stichwortverzeichnis</b>	<b>285</b>

# Vorwort

Der vorliegende vierte Band der Reihe „Vorlesungen über Informatik“ entstand aus Vorlesungen an der Universität Karlsruhe. Er behandelt parallele Rechenverfahren und Problemlösungsmethoden, die nicht auf ein mathematisches Lösungsmodell im üblichen Sinne zurückgreifen, und baut in vielfältiger Weise auf den Ergebnissen der Vorgängerbände auf, die meist nur mit Angabe der Abschnittsnummern zitiert werden.

Kap. 19 ist dem Thema paralleles Rechnen gewidmet. Zerlegung in parallel ausführbare Teile ist wie rekursive Problemzerlegung eines der Grundprinzipien der Informatik. Dementsprechend stehen in diesem Kapitel nicht so sehr die technischen Möglichkeiten von Parallelrechnern, sondern die abstrakten Modelle solcher Rechner als Grundlage für den Entwurf paralleler Algorithmen im Vordergrund. Daran schließen sich Entwurfsmethoden an, die vor allem zeigen, daß parallele Algorithmen nicht nur durch Parallelisierung von Schleifen aus sequentiellen Programmen entstehen.

Viele parallele Algorithmen kann man in Netzen von Arbeitsplatzrechnern ausführen, wenn die erforderliche Kommunikationsbandbreite zur Verfügung steht. Insoweit besteht zwischen einem echten Parallelrechner und der Parallelarbeit in einem verteilten System kein prinzipieller Unterschied. Die ausführliche Behandlung der Abbildung planbarer paralleler Algorithmen auf das LogP-Modell resultiert aus der Erfahrung, daß ein sehr großer Teil der parallelen Programme im wissenschaftlichen Rechnen planbar ist; die vorgestellte Methodik liefert zusätzliche Leistungssteigerungen gegenüber herkömmlichen Implementierungsmethoden.

Der Engpaß vieler paralleler Programme ist nicht die Rechenleistung, sondern die zum Transport und zur Speicherung immenser Datenmengen erforderliche Kommunikationsbandbreite. Nicht die geschickte Verteilung der Arbeit auf die Prozessoren, sondern die geschickte Verteilung der Daten entscheidet in den meisten Fällen über den Erfolg. Daher sind optimale Lösungen auch gewöhnlich nicht für beliebige Prozessorzahlen  $p$  skalierbar, wie man dies fälschlicherweise vielleicht aufgrund der vorgestellten theoretischen Resultate vermuten möchte: Mit steigender Prozessorzahl ändern sich die Engpässe des Kommunikationssystems; Lösungen, die vielleicht bei  $p = 16$  tauglich erschienen, sind es für 256 Prozessoren vielleicht nicht mehr.



Die in Kap. 20 behandelten Zellularautomaten erscheinen vordergründig als Spezialfall eines massiv parallelen Rechenmodells im Sinne des vorangehenden Kapitels. Entscheidend ist aber der folgende Ansatz zum Problemlösen: Die Zellen eines Zellularautomaten unterscheiden nicht zwischen dem Ausführungszustand ihres Programms und den dazu benötigten Daten. Auch können sie wegen der ausschließlich lokalen Kommunikationsbeziehungen keinerlei Überblick über den globalen Zustand einer Berechnung gewinnen. Daß sich aus lokalen Verhaltensweisen global sichtbare systematische Entwicklungen ergeben, ist eine Erfahrung, die man in vielen physikalischen und sozialen Systemen machen kann. Zellularautomaten sind ein formales Hilfsmittel, um solche Phänomene zu studieren. Die Anwendungen zeigen, daß man mit ihnen vor allem auch solche Systeme simulieren kann.

Das Thema der weiteren Kapitel heißt heute im Amerikanischen *Soft Computing* oder *Computational Intelligence*. Gemeinhin zählt man hierzu die Gebiete *künstliche neuronale Netze*, *evolutionäre Algorithmen* sowie *unscharfes Schließen*, denen auch die Kapitel 21–23 gewidmet sind. Den Themenbereichen ist das natürliche Vorbild gemein: Auslöser ist die Beobachtung von Mechanismen in Biologie, Physik, Chemie und sozialen Systemen, die sich natürlich entwickelt und als erfolgreich erwiesen haben. Grundtechnik ist die Nachahmung dieser Mechanismen.

Biologen, Philosophen und Physiker befassen sich mit diesem Thema und versuchen damit menschliches Denken aus seiner vermuteten biologischen Arbeitsweise heraus zu erklären. Für den Informatiker und Ingenieur sind neben dem Erkenntnisgewinn vor allem Gesichtspunkte wichtig, wie sie LOTFI A. ZADEH (1994) in seiner Definition von *Soft Computing* betont:

*The real world is pervasively imprecise and uncertain. Precision and certainty carry a cost. The guiding principle of soft computing is: Exploit the tolerance for imprecision, uncertainty, and partial truth to achieve tractability, robustness, and low solution cost.*

Hierauf weist auch der Untertitel *nicht-analytische Lösungsverfahren* des Buches hin: Analytische Modellbildung auf mathematischer Grundlage verlangt bei vielen Problemen nur schwer oder gar nicht zugängliche Informationen, ohne daß die Genauigkeit der Lösung irgendeinen signifikanten Vorteil gegenüber einer ungenaueren Lösung hätte, die mit wesentlich geringerem Aufwand zu erreichen ist. Methoden des *Soft Computing* sind darüber hinaus auch bei Problemen einsetzbar, für die überhaupt keine analytischen Modelle verfügbar oder denkbar sind. Man spricht daher oft von *modellfreien* Verfahren, obwohl in Wahrheit klar definierte Rechenmodelle vorliegen.

Bei der Auswahl des Stoffes für das Kap. 21 standen naturgemäß die praktisch wichtigen Verfahren zum überwachten Lernen in vorwärtsgerichteten Netzen

im Vordergrund. Beim unüberwachten Lernen werden Kohonen-Karten und Hopfield-Netze ausführlich behandelt; viele weiterführende Konzepte bleiben unberücksichtigt.

Kap. 22 über zufallsgesteuerte Optimierung beginnt mit Gradientenverfahren. Diese deterministischen Verfahren gehören eigentlich nicht unter die Kapitelüberschrift, werden aber als Grundlage in weiteren Verfahren benötigt; beim Gebrauch in Vorlesungen kann man diesen Abschnitt auch als Exkurs vor 21.3.1 einschieben, wo sie ebenfalls angewandt werden. Der praktische Einsatz der weiteren Verfahren wie simuliertes Tempern, Evolutionsstrategien und genetische Algorithmen verlangt Erfahrung bei der Problem-Codierung und der zweckmäßigen Wahl der Parameter, die in einer Einführung nur unzureichend vermittelt werden kann.

Kap. 23 über unscharfe Informationsverarbeitung konzentriert sich zuerst auf unscharfe Mengen und Relationen, um diese dann bei Aufgaben der unscharfen Regelung einzusetzen. Abschließend werden kursorisch unscharfe Maße behandelt. In Vorlesungen kann man auch unscharfe Maße an den Anfang stellen und von da zu unscharfen Mengen und unscharfer Logik übergehen.

Viele weiterführende Hinweise und praktische Beispiele zu den hier behandelten Themen findet man heute auf dem Internet. Wir haben daher an einigen Stellen Netz-Adressen angegeben, obwohl solche Adressen schneller sind als ein Buch.

Beim Schreiben dieses Buches haben mich meine Mitarbeiter ANDREAS HERBERLE und MARTIN TRAPP tatkräftig unterstützt. Das erste Kapitel geht auf eine erste Ausarbeitung der Herren Dr. WOLF ZIMMERMANN und Dr. WELF LÖWE, das zweite Kapitel auf Herrn Dr. THOMAS WORSCH und die restlichen Kapitel auf Herrn Dr. JOACHIM WEISBROD und Herrn MARTIN SPOTT zurück. Ihnen und den Mitarbeitern Dr. UWE ASSMANN, JÖRN EISENBIEGLER, THILO GAUL, DANIELA GENIUS, THOMAS LINDNER, ANDREAS LUDWIG und RAINER NEUMANN danke ich für die Mitarbeit und Durchsicht der einzelnen Kapitel, aus denen viele Korrekturen und Verbesserungen resultierten. Den Kollegen WOLFGANG BANZHAF, PETER LOCKEMANN, ALFRED SCHMITT, HANS-PAUL SCHWEFEL, THEO UNGERER und ROLAND VOLLMAR sowie den Herren Dr. JÜRGEN GOOS und Dr. MARTIN RIEDMILLER danke ich für die Durchsicht von Teilen des Manuskripts, zahlreiche nützliche Hinweise und Verbesserungsvorschläge. Die Mitarbeiter des Springer-Verlags, allen voran Herr ENGESSER, Frau GEORGIADIS und Herr STRASSER, haben mich auch bei diesem Band und seiner endgültigen Gestaltung tatkräftig unterstützt. Nicht zuletzt bedanke ich mich bei meiner Frau für die Durchsicht des Manuskripts und die Nachsicht und Geduld, mit der sie die Arbeit auch an diesem Buch ertrug.