

**Leitfäden und Monographien
der Informatik**

**Ingo Wegener
Theoretische Informatik**

Leitfäden und Monographien der Informatik

Herausgegeben von

Prof. Dr. Hans-Jürgen Appelrath, Oldenburg

Prof. Dr. Volker Claus, Stuttgart

Prof. Dr. Günter Hotz, Saarbrücken

Prof. Dr. Klaus Waldschmidt, Frankfurt

Die Leitfäden und Monographien behandeln Themen aus der Theoretischen, Praktischen und Technischen Informatik entsprechend dem aktuellen Stand der Wissenschaft. Besonderer Wert wird auf eine systematische und fundierte Darstellung des jeweiligen Gebietes gelegt. Die Bücher dieser Reihe sind einerseits als Grundlage und Ergänzung zu Vorlesungen der Informatik und andererseits als Standardwerke für die selbständige Einarbeitung in umfassende Themenbereiche der Informatik konzipiert. Sie sprechen vorwiegend Studierende und Lehrende in Informatik-Studiengängen an Hochschulen an, dienen aber auch in Wirtschaft, Industrie und Verwaltung tätigen Informatikern zur Fortbildung im Zuge der fortschreitenden Wissenschaft.

Theoretische Informatik

Eine algorithmenorientierte Einführung

Von Prof. Dr. math. Ingo Wegener
Universität Dortmund



B.G. Teubner Stuttgart 1993

Prof. Dr. math. Ingo Wegener

Geboren 1950 in Bremen, Studium der Mathematik und Soziologie in Bielefeld, Diplom 1976, Promotion 1978, Habilitation 1981. Von 1980 bis 1987 zunächst als Gastprofessor dann als C3-Professor am Fachbereich Informatik der Johann Wolfgang Goethe-Universität in Frankfurt am Main, seit 1987 als C4-Professor für das Gebiet Komplexitätstheorie und Effiziente Algorithmen am Fachbereich Informatik der Universität Dortmund.

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Wegener, Ingo:

Theoretische Informatik : eine algorithmenorientierte
Einführung / von Ingo Wegener. - Stuttgart : Teubner, 1993
(Leitfäden und Monographien der Informatik)

ISBN 978-3-519-02123-0 ISBN 978-3-322-94687-4 (eBook)

DOI 10.1007/978-3-322-94687-4

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

© B. G. Teubner Stuttgart 1993

Gesamtherstellung: Zehnersche Buchdruckerei GmbH, Speyer
Einband: P.P.K.S.-Konzepte, T. Koch, Ostfildern/Stuttgart

Vorwort

Die Theoretische Informatik ist älter als die Praktische, Angewandte oder Technische Informatik. Daher ist sie als wissenschaftliche Disziplin bereits weiter ausgebaut als andere Bereiche der Informatik, und ihre Ergebnisse sind schwerer zugänglich, da sie auf ein größeres und tieferes Fundament aufbauen. Stark verästelte Theorien tendieren dazu, sich als Selbstzweck aufzufassen und als *l'art pour l'art* betrieben zu werden. In der vorliegenden Einführung in die Theoretische Informatik begegnen wir dieser Gefahr, indem wir die Orientierung moderner Theorien an den Anwendungen in den Mittelpunkt stellen. Schon Novalis (1772–1801) hat darauf hingewiesen, daß die Theorie häufig den Anwendungen vorauseilt: „Wenn die Theorie auf die Erfahrung warten sollte, so käme sie nie zustande.“

Nicht immer sind die Anwendungen von Ergebnissen der Theoretischen Informatik so direkt zu sehen wie die Anwendungen anderer Zweige der Informatik. Dies gilt insbesondere für negative Resultate. Dabei sind deren Konsequenzen klar. Wenn wir beweisen, daß es bestimmte für die Praxis wünschenswerte Werkzeuge oder Algorithmen nicht geben kann, muß die unsinnige, weil hoffnungslose Arbeit an diesen Werkzeugen oder Algorithmen eingestellt und statt dessen die Suche nach bestmöglichen Auswegen begonnen werden.

Andererseits sind positive Resultate nicht automatisch anwendungsorientiert. Existenzaussagen oder Algorithmen mit exponentieller oder noch größerer Laufzeit sind häufig praktisch wertlos. Das Neue an der vorliegenden Einführung in die Theoretische Informatik ist die konsequent algorithmenorientierte Sichtweise (zum didaktischen Hintergrund siehe Wegener (1992)). Stets wurde bei positiven Resultaten eine Umsetzung in praktisch und theoretisch effiziente Algorithmen angestrebt. Um den Rahmen nicht zu sprengen, geben wir uns in einigen wenigen Fällen mit einfachen effizienten Algorithmen zufrieden, wenn eine weitere Effizienzsteigerung nur mit komplizierten Datenstrukturen möglich und die Verbesserung marginal ist.

Das Buch enthält eine Einführung in die zentralen Gebiete der Theoretischen Informatik: Komplexitätstheorie mit den Bereichen Entscheidbarkeit und NP-Vollständigkeit, Automatentheorie, Grundlagen von Programmiersprachen und Syntaxanalyse. Die Kernbereiche des Buches bilden den Stoff einer vierstündigen Einführungsvorlesung in die Theoretische Informatik, so wie sie an den deutschen Universitäten vorgesehen ist. Als Umfang einer derartigen Vorlesung schlage ich die folgenden Kapitel vor: 1, 2.1–2.5, 2.6 bis 2.6.8, 2.7, 2.8, 3.1–3.4, 4.1–4.3, 4.4. bis 4.4.6, 4.6, 5.1, 5.2, 5.3 bis 5.3.4, 6.1, 6.2 bis 6.2.2, 6.3–6.5, 6.6 bis 6.6.2, 6.7, 7.1 bis 7.1.3, 7.2–7.4 und 9.1. Die weiteren Kapitel bieten Stoff für sinnvolle Ergänzungen, die als spezielle Schwerpunkte in die Vorlesung eingebaut werden können. Es kann aber auch empfohlen werden, einige der ergänzenden Abschnitte in einem die Vorlesung

VI

begleitenden Proseminar zu behandeln. Die vorgeschlagenen Übungsaufgaben beziehen sich vor allem auf die Kernbereiche und sind in ihrem Schwierigkeitsgrad auf Studierende im Grundstudium zugeschnitten. Das Buch schließt mit einer Zusammenfassung der erzielten Resultate, um Querverbindungen aufzuzeigen, und einer Liste von typischen Prüfungsfragen über den behandelten Stoff.

Für die sorgfältige und rasche Erstellung des Manuskripts bedanke ich mich bei Matthias Bußmann, Werner Traidl, Ludwig Voß und vor allem bei Heike Griehl. Für Anregungen, Diskussionen, Beispiele und Korrekturen danke ich Bea Bollig, Martin Dietzfelbinger, Matthias Frommknecht, Heike Griehl, Jozef Gruska, Thomas Hofmeister, Friedhelm Meyer auf der Heide, Uwe Schöning, Ricki Wegner, Derick Wood und Detlef Wotschke. Christa danke ich dafür, daß sie mich anregt, nicht nur die Theoretische Informatik von immer neuen Seiten zu betrachten.

Dortmund/Bielefeld, im Oktober 1992

Ingo Wegener

Inhaltsverzeichnis

1	Einleitung	1
2	Turingmaschinen, Churchsche These und Entscheidbarkeit	7
2.1	Registermaschinen und deterministische Turingmaschinen	7
2.2	Techniken zur Programmierung von Turingmaschinen	14
2.3	Simulationen zwischen Turingmaschinen und Registermaschinen . . .	16
2.4	Universelle Turingmaschinen	18
2.5	Die Churchsche These	20
2.6	Die Unentscheidbarkeit des Halteproblems	21
2.7	Eigenschaften rekursiver und rekursiv aufzählbarer Sprachen	26
2.8	Die Unentscheidbarkeit des Postschen Korrespondenzproblems	28
	Übungen	33
3	Die NP-Vollständigkeitstheorie	36
3.1	Die Klasse P	36
3.2	Nichtdeterministische Turingmaschinen und die Klasse NP	39
3.3	NP-Vollständigkeit	43
3.4	Die NP-Vollständigkeit wichtiger Probleme	51
3.5	Pseudopolynomielle Algorithmen und starke NP-Vollständigkeit . . .	59
3.6	Turing-Reduzierbarkeit, NP-harte, NP-leichte und NP-äquivalente Probleme	63
3.7	Eine Komplexitätstheorie für Approximationsalgorithmen	68
3.8	Eine Komplexitätstheorie für probabilistische Algorithmen	73
3.9	Die Struktur von NP und die polynomielle Hierarchie	80
	Übungen	83
4	Endliche Automaten	86
4.1	Schaltwerke und endliche Automaten	86
4.2	Die Minimierung endlicher Automaten	92
4.3	Das Pumping-Lemma für endliche Automaten	101

VIII Inhaltsverzeichnis

4.4	Nichtdeterministische endliche Automaten	103
4.5	Zwei-Wege Automaten	108
4.6	Effiziente Algorithmen für die Konstruktion endlicher Automaten und die Entscheidung von Eigenschaften regulärer Sprachen	114
	Übungen	122
5	Grammatiken, die Chomsky-Hierarchie und das Wortproblem	124
5.1	Grammatiken und die Chomsky-Hierarchie	124
5.2	Chomsky-0-Grammatiken und rekursiv aufzählbare Sprachen	127
5.3	Chomsky-3-Grammatiken, reguläre Sprachen und Ausdrücke, lexika- lische Analyse	129
5.4	Kontextsensitive Grammatiken und Sprachen	136
	Übungen	144
6	Kontextfreie Grammatiken und Sprachen	146
6.1	Beispiele kontextfreier Sprachen und Syntaxbäume	146
6.2	Die Chomsky-Normalform für kontextfreie Grammatiken	149
6.3	Der Cocke-Younger-Kasami Algorithmus	153
6.4	Das Pumping-Lemma und Ogden's Lemma für kontextfreie Sprachen	154
6.5	Effiziente Algorithmen für die Konstruktion kontextfreier Grammati- ken und die Entscheidung von Eigenschaften kontextfreier Sprachen .	158
6.6	Unentscheidbare Probleme	162
6.7	Eine inhärent mehrdeutige kontextfreie Sprache	168
	Übungen	170
7	Kellerautomaten und kontextfreie Sprachen	172
7.1	Die Greibach-Normalform für kontextfreie Grammatiken	172
7.2	Kellerautomaten	182
7.3	Kellerautomaten und kontextfreie Sprachen	185
7.4	Weitere effiziente Algorithmen im Zusammenhang mit kontextfreien Sprachen	189
	Übungen	191
8	Deterministisch kontextfreie Sprachen	192
8.1	Deterministische Kellerautomaten	192

8.2	Bottom-up Syntaxanalysealgorithmen	199
8.3	Eine weitere Charakterisierung von LR(k)-Grammatiken	206
8.4	Die Konstruktion eines LR(k)-Parsers	210
8.5	Deterministische Kellerautomaten und LR(k)-Grammatiken	215
	Übungen	216
9	Zusammenfassung und Testfragen	218
9.1	Zusammenfassung	218
9.2	Testfragen	223
	Schriftenverzeichnis	228
	Index	232