

# Informatik – Fachberichte

---

- Band 164: M. Eulenstein, Generierung portabler Compiler. X, 235 Seiten. 1988.
- Band 165: H.-U. Heiß, Überlast in Rechensystemen. IX, 176 Seiten. 1988.
- Band 166: K. Hörmann, Kollisionsfreie Bahnen für Industrieroboter. XII, 157 Seiten. 1988.
- Band 167: R. Lauber (Hrsg.), Prozeßrechensysteme '88. Stuttgart, März 1988. Proceedings. XIV, 799 Seiten. 1988.
- Band 168: U. Kastens, F. J. Rammig (Hrsg.), Architektur und Betrieb von Rechensystemen. 10. GI/ITG-Fachtagung, Paderborn, März 1988. Proceedings. IX, 405 Seiten. 1988.
- Band 169: G. Heyer, J. Krems, G. Görz (Hrsg.), Wissensarten und ihre Darstellung. VIII, 292 Seiten. 1988.
- Band 170: A. Jaeschke, B. Page (Hrsg.), Informatikanwendungen im Umweltbereich. 2. Symposium, Karlsruhe, 1987. Proceedings. X, 201 Seiten. 1988.
- Band 171: H. Lutterbach (Hrsg.), Non-Standard Datenbanken für Anwendungen der Graphischen Datenverarbeitung. GI-Fachgespräch, Dortmund, März 1988, Proceedings. VII, 183 Seiten. 1988.
- Band 172: G. Rahmstorf (Hrsg.), Wissensrepräsentation in Expertensystemen. Workshop, Herrenberg, März 1987. Proceedings. VII, 189 Seiten. 1988.
- Band 173: M. H. Schulz, Testmustergenerierung und Fehlersimulation in digitalen Schaltungen mit hoher Komplexität. IX, 165 Seiten. 1988.
- Band 174: A. Endrös, Rechtsprechung und Computer in den neunziger Jahren. XIX, 129 Seiten. 1988.
- Band 175: J. Hülsemann, Funktioneller Test der Auflösung von Zugriffskonflikten in Mehrrechnersystemen. X, 179 Seiten. 1988.
- Band 176: H. Trost (Hrsg.), 4. Österreichische Artificial-Intelligence-Tagung. Wien, August 1988. Proceedings. VIII, 207 Seiten. 1988.
- Band 177: L. Voelkel, J. Pliquet, Signaturanalyse. 223 Seiten. 1989.
- Band 178: H. Göttler, Graphgrammatiken in der Softwaretechnik. VIII, 244 Seiten. 1988.
- Band 179: W. Ameling (Hrsg.), Simulationstechnik. 5. Symposium. Aachen, September 1988. Proceedings. XIV, 538 Seiten. 1988.
- Band 180: H. Bunke, O. Kübler, P. Stucki (Hrsg.), Mustererkennung 1988. 10. DAGM-Symposium, Zürich, September 1988. Proceedings. XV, 361 Seiten. 1988.
- Band 181: W. Hoepfner (Hrsg.), Künstliche Intelligenz. GWAI-88, 12. Jahrestagung. Eringerfeld, September 1988. Proceedings. XII, 333 Seiten. 1988.
- Band 182: W. Barth (Hrsg.), Visualisierungstechniken und Algorithmen. Fachgespräch, Wien, September 1988. Proceedings. VIII, 247 Seiten. 1988.
- Band 183: A. Clauer, W. Purgathofer (Hrsg.), AUSTROGRAPHICS '88. Fachtagung, Wien, September 1988. Proceedings. VIII, 267 Seiten. 1988.
- Band 184: B. Gollan, W. Paul, A. Schmitt (Hrsg.), Innovative Informations-Infrastrukturen. I.I.I. – Forum, Saarbrücken, Oktober 1988. Proceedings. VIII, 291 Seiten. 1988.
- Band 185: B. Mitschang, Ein Molekül-Atom-Datenmodell für Non-Standard-Anwendungen. XI, 230 Seiten. 1988.
- Band 186: E. Rahm, Synchronisation in Mehrrechner-Datenbanksystemen. IX, 272 Seiten. 1988.
- Band 187: R. Valk (Hrsg.), GI – 18. Jahrestagung I. Vernetzte und komplexe Informatik-Systeme. Hamburg, Oktober 1988. Proceedings. XVI, 776 Seiten.
- Band 188: R. Valk (Hrsg.), GI – 18. Jahrestagung II. Vernetzte und komplexe Informatik-Systeme. Hamburg, Oktober 1988. Proceedings. XVI, 704 Seiten.
- Band 189: B. Wolfinger (Hrsg.), Vernetzte und komplexe Informatik-Systeme. Industrieprogramm zur 18. Jahrestagung der GI, Hamburg, Oktober 1988. Proceedings. X, 229 Seiten. 1988.
- Band 190: D. Maurer, Relevanzanalyse. VIII, 239 Seiten. 1988.
- Band 191: P. Levi, Planen für autonome Montageroboter. XIII, 259 Seiten. 1988.
- Band 192: K. Kansy, P. Wißkirchen (Hrsg.), Graphik im Büroereich. Proceedings, 1988. VIII, 187 Seiten. 1988.
- Band 193: W. Gotthard, Datenbanksysteme für Software-Produktionsumgebungen. X, 193 Seiten. 1988.
- Band 194: C. Lewerentz, Interaktives Entwerfen großer Programmsysteme. VII, 179 Seiten. 1988.
- Band 195: I. S. Bátori, U. Hahn, M. Pinkal, W. Wahlster (Hrsg.), Computerlinguistik und ihre theoretischen Grundlagen. Proceedings. IX, 218 Seiten. 1988.
- Band 197: M. Leszak, H. Eggert, Petri-Netz-Methoden und -Werkzeuge. XII, 254 Seiten. 1989.
- Band 198: U. Reimer, FRM: Ein Frame-Repräsentationsmodell und seine formale Semantik. VIII, 161 Seiten. 1988.
- Band 199: C. Beckstein, Zur Logik der Logik-Programmierung. IX, 246 Seiten. 1988.
- Band 200: A. Reinefeld, Spielbaum-Suchverfahren. IX, 191 Seiten. 1989.
- Band 201: A. M. Kotz, Triggermechanismen in Datenbanksystemen. VIII, 187 Seiten. 1989.
- Band 202: Th. Christaller (Hrsg.), Künstliche Intelligenz. 5. Frühjahrsschule, KIFS-87, Günne, März/April 1987. Proceedings. VII, 403 Seiten. 1989.
- 1989.
- Band 203: K. v. Luck (Hrsg.), Künstliche Intelligenz. 7. Frühjahrsschule, KIFS-89, Günne, März 1989. Proceedings. VII, 302 Seiten. 1989.
- Band 204: T. Härder (Hrsg.), Datenbanksysteme in Büro, Technik und Wissenschaft. GI/SI-Fachtagung, Zürich, März 1989. Proceedings. XII, 427 Seiten. 1989.
- Band 205: P. J. Kühn (Hrsg.), Kommunikation in verteilten Systemen. ITG/GI-Fachtagung, Stuttgart, Februar 1989. Proceedings. XII, 907 Seiten. 1989.
- Band 206: P. Horster, H. Isselhorst, Approximative Public-Key-Kryptosysteme. VII, 174 Seiten. 1989.
- Band 207: J. Knop (Hrsg.), Organisation der Datenverarbeitung an der Schwelle der 90er Jahre. 8. GI-Fachgespräch, Düsseldorf, März 1989. Proceedings. IX, 276 Seiten. 1989.
- Band 208: J. Retti, K. Leidlmair (Hrsg.), 5. Österreichische Artificial-Intelligence-Tagung, Igls/Tirol, März 1989. Proceedings. XI, 452 Seiten. 1989.
- Band 209: U. W. Lipeck, Dynamische Integrität von Datenbanken. VIII, 140 Seiten. 1989.
- Band 210: K. Drosten, Termersetzungssysteme. IX, 152 Seiten. 1989.
- Band 211: H. W. Meuer (Hrsg.), SUPERCOMPUTER '89. Mannheim, Juni 1989. Proceedings, 1989. VIII, 171 Seiten. 1989.
- Band 212: W.-M. Lippe (Hrsg.), Software-Entwicklung. Fachtagung, Marburg, Juni 1989. Proceedings. IX, 290 Seiten. 1989.
- Band 213: I. Walter, Datenbankgestützte Repräsentation und Extraktion von Episodenbeschreibungen aus Bildfolgen. VIII, 243 Seiten. 1989.
- Band 214: W. Görke, H. Sörensen (Hrsg.), Fehlertolerierende Rechensysteme / Fault-Tolerant Computing Systems. 4. Internationale GI/ITG/GMA-Fachtagung, Baden-Baden, September 1989. Proceedings. XI, 390 Seiten. 1989.

## Informatik-Fachberichte 261

---

Herausgeber: W. Brauer  
im Auftrag der Gesellschaft für Informatik (GI)

Wolf Zimmermann

# Automatische Komplexitätsanalyse funktionaler Programme



Springer-Verlag  
Berlin Heidelberg GmbH

**Autor**

Wolf Zimmermann  
Institut für Programmstrukturen und Datenorganisation  
Universität Karlsruhe  
Postfach 6980, W-7500 Karlsruhe 1

CR Subject Classification (1987): F.2, D.3.2

ISBN 978-3-540-53430-3      ISBN 978-3-662-05948-7 (eBook)  
DOI 10.1007/978-3-662-05948-7

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, bei auch 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 1990  
Ursprünglich erschienen bei Springer-Verlag Berlin Heidelberg New York 1990

## Geleitwort

Es gibt im Bereich der Softwaretechnik viele Werkzeuge, die die Entwicklung von Software unterstützen und es ermöglichen, auch große Systeme zu realisieren. Einige dieser Werkzeuge beziehen auch Korrektheitsfragen ein und versuchen, den Entwicklungsprozeß zu systematisieren und Korrektheit nicht nur durch nachträgliche Verifikation, sondern durch die Anwendung formaler Methoden während der Konstruktion zu garantieren. Neben der Korrektheit spielt natürlich auch die Effizienz eine wesentliche Rolle bei der Programmkonstruktion, aber im Gegensatz zur Korrektheit stehen für die Gewährleistung von Effizienz kaum geeignete Werkzeuge zur Verfügung, die bereits im Prozeß der Konstruktion angewandt werden könnten.

Die vorliegende Arbeit führt eine Methode ein, die es erlaubt, die Zeitkomplexität funktionaler Programme oder Spezifikationen automatisch zu ermitteln. Damit wird es möglich, bereits in frühen Phasen der Entwicklung durch Betrachtung der Effizienz die Auswahl, aber auch den Entwurf von Algorithmen zu unterstützen. Die Grundidee dieser Methode besteht darin, ein funktionales Programm in ein System von Rekurrenzgleichungen zu übersetzen, das das Zeitverhalten des Programms beschreibt. Anschließend wird dieses Rekurrenzsystm gelöst. Die Einführung von *bedingten Rekurrenzen* und *Rekurrenzfamilien* ermöglicht es, obere und untere Schranken für die Zeitkomplexität zu finden. Um die mittlere Zeitkomplexität zu bestimmen, werden mit Hilfe einer probabilistischen Semantik des Programms Wahrscheinlichkeiten dafür berechnet, daß im Programm vorkommende Bedingungen wahr bzw. falsch werden. Um möglichst genaue Schranken für die Zeitkomplexität zu erhalten, wird eine *Abhängigkeitsanalyse* durchgeführt. Das Verfahren ermöglicht eine genaue Analyse von Divide-and-Conquer-Programmen.

Der in dieser Arbeit vorgestellte Ansatz ermöglicht es, die oberen und unteren Schranken für die Zeitkomplexität, sowie die mittlere Zeitkomplexität einiger bekannter Algorithmen (*Quicksort*, Sortieren durch Einfügen, Horners Algorithmus, schnelle Fouriertransformation) automatisch zu analysieren. Das Analyseergebnis besteht dabei nicht nur in der asymptotischen Komplexität, sondern enthält vor allem auch die konstanten Faktoren, die wesentlich zur Auswahl geeigneter Algorithmen beitragen.

Karlsruhe, im Juli 1990

Stefan Jähnichen

## Vorwort

Die vorliegende Arbeit wurde als Dissertation der Fakultät für Informatik der Universität Karlsruhe (TH) vorgelegt und genehmigt.

Ich bedanke mich bei meinen beiden Betreuern Prof. Dr. Stefan Jähnichen und Prof. Dr. Thomas Beth für zahlreiche Diskussionen und Anregungen. Ich bedanke mich auch bei Prof. Dr. Gerhard Goos, der mich in der Anfangsphase betreute. Ich danke Prof. Dr. Philippe Flajolet (INRIA Rocquencourt) und Prof. Jähnichen für die Ermöglichung eines Kurzaufenthaltes bei INRIA Rocquencourt. Prof. Flajolet möchte ich an dieser Stelle auch für seine Gesprächsbereitschaft danken. Der Aufenthalt bei INRIA brachte mir zahlreiche Diskussionspartner. Die Diskussionen hatten sicherlich Einfluß auf meine Arbeit. Besonders danken möchte ich meinem Kollegen Paul Zimmermann (INRIA Rocquencourt), der diese Arbeit in einer früheren Version sorgfältig durchlas und die Beweise genau prüfte. Ich danke außerdem Bruno Salvy für seine Geduld bei meinen ersten Versuchen mit MAPLE.

Der Ursprung dieser Arbeit liegt im ESPRIT-Projekt 510 ToolUse. Ich danke allen Kollegen im Projekt für ihre Bereitschaft zum Zuhören und Diskutieren. Besonders bedanke ich mich bei René Jacquart (CERT Toulouse), der mich vor drei Jahren auf die Idee brachte, das Thema automatische Komplexitätsanalyse zu betrachten. Ich danke außerdem Jacques Sauloy (Midival Toulouse), der mich beim Beweis der Sätze 6.20 und 6.21 unterstützte.

Ich bedanke mich ferner bei meinen Kollegen Frank Bieler und Dr. Roland Dietrich von der GMD in Karlsruhe, die mir wertvolle Hinweise zur Gestaltung der Einleitung und zu den Schlußfolgerungen gaben.

Karlsruhe, im Juli 1990

Wolf Zimmermann

### *Hinweis für den Leser:*

Für Leser, die mit Lösungsverfahren für Rekurrenzen nicht vertraut sind, empfiehlt es sich, zuerst den Anhang A zu lesen. Dort sind die wesentlichen Lösungsverfahren zusammengefaßt. Im Anhang C finden sich Eigenschaften und Folgen von Funktionen, die in den Beweisen in Kapitel 5 benötigt werden. Wer mit den Eigenschaften monotoner und konvexer Folgen bzw. Funktionen nicht vertraut ist, sollte also vor Kapitel 5 den Anhang C lesen.

# Inhalt

<b>1</b>	<b>Einleitung</b>	<b>3</b>
1.1	Motivation und Problemstellung . . . . .	3
1.2	Verwandte Arbeiten . . . . .	4
1.3	Lösungsansatz und Aufbau der Arbeit . . . . .	5
<b>2</b>	<b>Ansätze zur automatischen Komplexitätsanalyse</b>	<b>8</b>
2.1	Der Ansatz mit erzeugenden Funktionen . . . . .	8
2.1.1	Algebraische Analyse . . . . .	9
2.1.1.1	Aufzählungen von Datenstrukturen . . . . .	9
2.1.1.2	Der Zeitaufwand von Programmen . . . . .	20
2.1.2	Analytische Analyse . . . . .	26
2.1.2.1	Bestimmung der Koeffizienten über Tabellen . . . . .	26
2.1.2.2	Grundlagen aus der Funktionentheorie . . . . .	27
2.1.2.3	Singularitätenanalyse . . . . .	29
2.1.2.4	Sattelpunktschätzungen . . . . .	33
2.1.2.5	Zusammenfassung . . . . .	36
2.1.3	Bewertung des Verfahrens . . . . .	36
2.2	Der Ansatz mit Rekurrenzen . . . . .	37
2.2.1	Das Ermitteln von Rekurrenzen . . . . .	38
2.2.2	Das Lösen von Rekurrenzen . . . . .	44
2.2.3	Übertragung auf FP-Programme . . . . .	47
2.2.4	Bewertung des Verfahrens . . . . .	48
2.3	Der probabilistische Ansatz . . . . .	48
2.4	Weitere Vorgehensweise . . . . .	57
<b>3</b>	<b>Das Maschinenmodell</b>	<b>58</b>
3.1	Die Sprache STYFL . . . . .	58
3.1.1	Typen . . . . .	58
3.1.2	Funktionen . . . . .	63
3.1.3	Syntaktische Eigenschaften von Programmen . . . . .	65
3.1.4	Semantik von Programmen . . . . .	69
3.2	Definition eines Zeitbegriffs . . . . .	76
<b>4</b>	<b>Das Abbilden auf Rekurrenzen</b>	<b>78</b>
4.1	Von Funktionen zu den rekursiven Gleichungen . . . . .	80
4.1.1	Vorverarbeitung . . . . .	80
4.1.2	Übersetzung in Zeitfunktionen und Normalisierung . . . . .	85
4.1.3	Symbolische Auswertung . . . . .	88

4.2	Von Zeitgleichungen zu Rekurrenzen . . . . .	92
4.2.1	Abstraktionsfunktionen . . . . .	93
4.2.2	Übersetzung in Maßfunktionen . . . . .	97
<b>5</b>	<b>Das Lösen von Rekurrenzen</b> . . . . .	<b>102</b>
5.1	Grundlegende Begriffe . . . . .	103
5.1.1	Rekurrenzen, Systeme von Rekurrenzen und Rekurrenzfamilien . . . . .	103
5.1.2	Arithmetische Ausdrücke . . . . .	106
5.2	Vorverarbeitung . . . . .	109
5.3	Bedingte Rekurrenzen . . . . .	111
5.3.1	Der beste und der schlechteste Fall . . . . .	112
5.3.2	Der mittlere Fall . . . . .	116
5.4	Abhängigkeitsanalyse . . . . .	118
5.5	Rekurrenzfamilien . . . . .	124
5.5.1	Der beste und der schlechteste Fall . . . . .	125
5.5.2	Der mittlere Fall . . . . .	131
<b>6</b>	<b>Probabilistische Semantik</b> . . . . .	<b>135</b>
6.1	Wahrscheinlichkeitsmaße auf allgemeinen Typen . . . . .	135
6.2	Wahrscheinlichkeitsmaßtransformationen . . . . .	147
6.3	Berechnung von Wahrscheinlichkeiten . . . . .	155
<b>7</b>	<b>Zusammenfassung und Ausblick</b> . . . . .	<b>157</b>
7.1	Einordnung dieser Arbeit . . . . .	157
7.2	Methode und Implementierung . . . . .	159
7.2.1	Reduktion auf Rekurrenzen . . . . .	159
7.2.2	Lösen von Rekurrenzen . . . . .	160
7.2.3	Implementierung . . . . .	160
7.3	Ausblick . . . . .	163
<b>A</b>	<b>Das Lösen von Rekurrenzen und Rekurrenzsystemen</b> . . . . .	<b>164</b>
A.1	Lineare Rekurrenzen 1-ter Ordnung . . . . .	164
A.2	Lineare Rekurrenzen $r$ -ter Ordnung mit konstanten Koeffizienten . . . . .	165
A.3	Rekurrenzsysteme . . . . .	168
A.4	Die Methode der erzeugenden Funktionen . . . . .	175
<b>B</b>	<b>Die Korrektheit der Übersetzungen</b> . . . . .	<b>177</b>
<b>C</b>	<b>Eigenschaften von Folgen und Funktionen</b> . . . . .	<b>183</b>
C.1	Folgen . . . . .	183
C.2	Funktionen . . . . .	186
	<b>Literatur</b> . . . . .	<b>189</b>
	<b>Index</b> . . . . .	<b>192</b>