

## Bibliographische Schlussbemerkungen

Der Zweck dieses Anhangs ist die Auflistung von Werken, welche dem geneigten Leser entweder ein paar Schritte zurück, nämlich zu den wissenschaftlichen Quellen der Hauptresultate und ihrer ausführlichen Darstellung, oder dann ein paar Schritte vorwärts, nämlich zu dem im Augenblick am Erforschen Begriffenen, ermöglichen soll. Beide Wege verlieren sich ins Unbestimmte, das wir hier naturgemäss nur um Weniges aufhellen können.

### 0 Allgemeines und Geschichtliches

#### Zeitschriften für Theoretische Informatik

- *Journal of the Association for Computing Machinery (J. ACM)*, vierteljährlich, hrsg. von ACM.
- *Information and Computation* (früher: *Information and Control*), monatlich, hrsg. von Academic Press.
- *Theoretical Computer Science (TCS)*, hrsg. von North-Holland.

#### Newsletters

- *ACM Sigact News*, hrsg. von ACM special interest group on automata and computability theory; vierteljährlich.
- *Bulletin of the EATCS*, hrsg. von European Assoc. for Theoretical Computer Science.

#### Geschichte

- N. C. Metropolis (ed.), *A History of Computing in the Twentieth Century*. Academic Press, 1980.
- A. Hodges, *Alan Turing: The Enigma*. Simon and Schuster, 1983.
- K. Zuse, *Der Computer, mein Lebenswerk*. Springer, 1986.
- *Annals of the History of Computing*, hrsg. von AFIPS.

### 1 Berechenbarkeit und Aufzählbarkeit

Die ersten durchgeführten Ansätze zu einer Theorie der Berechenbarkeit entstanden im Zusammenhang mit dem sog. Grundlagenproblem der Mathematik und den daraus entspringenden Fragen nach Entscheidbarkeit und Unentscheidbarkeit, vgl. dazu:

- M. Davis (ed.), *The Undecidable: Basic Papers on Undecidable Propositions, Unsolvability Problems and Computable Functions*. Raven Press, 1965.

Dort finden sich die grundlegenden Arbeiten von Gödel, Church, Post, Turing etc. Die Berechnungstheorie in ihren Anfängen ist in folgenden einflussreichen Büchern zu finden:

- H. Hermes, *Aufzählbarkeit, Entscheidbarkeit, Berechenbarkeit*. Springer, 1961.

- S. C. Kleene, *Introduction to Metamathematics*. North-Holland, 1952.
- M. Davis, *Computability and Unsolvability*. McGraw-Hill, 1958.
- M. L. Minsky, *Computation: Finite and Infinite Machines*. Prentice-Hall, 1967.

Seither sind einige Lehrbücher erschienen, welche mit dem vorliegenden in Konkurrenz stehen; wir erwähnen

- M. D. Davis und E. J. Weyuker, *Computability, Complexity and Languages: Fundamentals of Theoretical Computer Science*. Academic Press, 1983.
- Z. Manna, *Mathematical Theory of Computation*. McGraw-Hill, 1974.
- H. R. Lewis und C. H. Papadimitriou, *Elements of the Theory of Computation*. Prentice-Hall, 1981.

Der Zusammenhang zwischen Programmiersprachen und Klassen von berechenbaren Funktionen (LOOP / primitiv-rekursive Funktionen; WHILE / partiell-rekursive Funktionen) ist nur ein kleines Kapitel in dem grossen Gebiet der Semantik von Programmiersprachen. Dazu empfehlen wir den hübschen Artikel:

- D. Harel, On Folk Theorems, *Comm. ACM* 23 (1980), S. 379-389.

und das neuere Werk zur Vertiefung:

- D. A. Schmidt, *Denotational Semantics: A Methodology for Language Development*. Allyn & Bacon, 1986.

## 2 Automaten und formale Sprachen

Dieses Gebiet hat verschiedene Wurzeln; einerseits die mathematische Linguistik, insbesondere aus der Schule von Chomsky, vgl. etwa

- N. Chomsky, Formal Properties of Grammars, in: *Handbook of Mathematical Psychology*, vol. 2 (Luce, et al., eds.). Wiley, 1963.

und andererseits die Neurologie, vgl. etwa die seminalen Arbeiten:

- S. C. Kleene, Representation of events in nerve nets and finite automata, in: *Automata Studies*. Princeton Univ. Press, 1956.
- M. O. Rabin und D. Scott, Finite Automata and Their Decision Problems, *IBM J. Res.* 3 (1959), S. 115-125.

Dieses Gebiet war vor allem in den 60er und frühen 70er Jahren aktiv. Einflussreichste Bücher sind die folgenden:

- S. Ginsburg, *The Mathematical Theory of Context-Free Languages*. McGraw-Hill, 1966.
- J. E. Hopcroft und J. D. Ullman, *Formal Languages and Their Relation to Automata*. Addison-Wesley, 1969.

Die wichtigsten Arbeiten sind zusammengetragen in:

- A. Salomaa, *Jewels of Formal Language Theory*. Computer Science Press, 1981.

Eine historische Übersicht gibt

- S. A. Greibach, *Formal Languages: Origins and Directions*, *Annals Hist. Comp.* **3** (1981), S. 14-41.

Der in unserem Buch dargestellte algebraische Zugang wurde vor allem vertreten in

- J. R. Büchi, *Mathematische Theorie des Verhaltens endlicher Automaten*, *Z. Angew. Math. Mech.* **42** (1962), S. 9-16. Eine detaillierte Darstellung der Theorie der Automaten findet man z.B. in
- W. Brauer, *Automatentheorie*. B. G. Teubner, Stuttgart, 1984.

### 3 Fixpunkttheorie

Die allgemeinste Form des Fixpunktsatzes erscheint zuerst in

- A. Tarski, *A Lattice-Theoretical Fixpoint Theorem and its Applications*, *Pacific J. Math.* **5** (1955), S. 285-309.

Während die meisten hier gemachten Anwendungen recht einfach sind, so ist doch der allgemeine Ansatz sehr fruchtbar, insbesondere in der sog. denotationellen Semantik von Programmiersprachen und Datenstrukturen. Vgl. dazu etwa:

- J. E. Stoy, *Denotational Semantics: The Scott-Strachey Approach to Programming Language Theory*. MIT Press, 1977.

### 4 Syntaktische Strukturen

Dieses Gebiet gehört eigentlich zur Universellen Algebra, wobei die Beziehung zu Sprachkonstruktionen wohl zuerst von L. Henkin ausgearbeitet wurde. Eine gute Einführung gibt

- S. Burris und H. P. Sankappanavar, *Universal Algebra*. Springer, 1981.

Mehr über die Beziehungen zwischen Universeller Algebra und Datenstrukturen findet sich in der Monographie

- P. Burmeister, *Model-Theoretic Oriented Approach to Partial Algebras*. Akademie Verlag Berlin, 1986.

### 5 Gödelnumerierung und Universalprogramme

Die Gödelnumerierung als technisches Hilfsmittel erscheint erstmals in:

- K. Gödel, *Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I*, *Monatshefte f. Math. u. Phys.* **38** (1931), S. 173-198.

Diese Arbeit ist, wie viele andere wichtige Arbeiten im schon erwähnten Sammelwerk enthalten:

- *The Undecidable* (M. Davis, ed.). Raven Press, 1965.

Dort sind auch die Arbeiten von Turing, insbesondere seine Erfindung einer Universalmaschine:

- A. Turing, *On Computable Numbers with an Application to the Entscheidungsproblem*, *Proc. London Math. Soc.* **42** (1936), S. 230-265, **43** (1937), S. 544-546.

Der Klassiker der Rekursionstheorie ist

- H. Rogers, *Theory of Recursive Functions and Effective Computability*. McGraw-Hill, 1967.

Als neuere Darstellung ist zu erwähnen:

- R. I. Soare, *Recursively Enumerable Sets and Degrees*. Springer 1987.

## 6 Unlösbare Probleme der Informatik

Die These von Church ist erstmals ausgesprochen in

- A. Church, An Unsolvable Problem of Elementary Number Theory, *Amer. J. Math.* **58** (1936), S. 345-363.

Sie stützt sich auf die Erfahrungen von Turing, Kleene, Post, Gödel und Herbrand mit Varianten des Berechnungsbegriffes, von denen einige in unserem Text zur Sprache kommen. Das Korrespondenzproblem stammt von Post und ist unentscheidbar gemäss

- E. L. Post, A Variant of a Recursively Unsolvable Problem, *Bull. AMS* **52** (1946), S. 264-268.

Die Tag-Maschine entspringt ebenfalls einer Idee von Post; die Unlösbarkeit des entsprechenden Halteproblems hat erstmals bewiesen:

- M. L. Minsky, Recursive Unsolvability of Post's Problem of 'tag' and other Topics in the Theory of Turing Machines, *Annals Math.* **74** (1961), S. 437-455.

Die Anwendung auf formale Sprachen (eine von vielen) stammt von D.G. Cantor. Über die Fragestellung nach Lösbarkeit / Unlösbarkeit hinaus geht die Komplexitätstheorie, die wir hier gar nicht aufnehmen. Einen Einstieg vermittelt etwa:

- M. R. Garey und D. S. Johnson, *Computers and Intractability: A Guide to NP-Completeness*. Freeman & Co., 1979.

## 7 Rekursive Prozeduren

Rekursive Prozeduren wurden erstmals systematisch als Grundelement des Programmierens verwendet durch McCarthy, vgl. z.B.:

- J. McCarthy in: *Computer Programming and Formal Systems* (Brafford & Hirschberg, eds.). North-Holland, 1963.

Die entsprechende, heute weitverbreitete Sprache LISP ist lehrbuchmässig dargestellt in:

- J. Allen, *The Anatomy of LISP*. McGraw-Hill, 1978.
- P. H. Winston und B. K. P. Horn, *LISP*. Addison-Wesley, 1984.

Zur Semantik rekursiver Prozeduren konsultiere man zusätzlich etwa

- Z. Manna, *Mathematical Theory of Computation*. McGraw-Hill, 1974.

Die hier dargestellte Theorie der Interpreter und Compiler geht zurück auf Ideen von A. Ershov, Y. Futamura und N. Jones, vgl.:

- A. P. Ershov, D. Bjørner und N. D. Jones (eds.). *Proc. of Workshop on Partial Evaluation and Mixed Computation*. North-Holland, 1988.

# Sachverzeichnis

- $A^*$  11
- $C^m$  20, 29, 30
- $D_i^m$  20, 29
- $\text{dom}(f)$  10, 27
- $e_p(x)$  78
- $E(\alpha)$  47
- $\mathbb{N}$  11
- $p_n$  78
- $\text{ran}(f)$  10, 27
- $\text{sg}(x)$  14
- $\overline{\text{sg}}(x)$  14
- $T^*$  32
- $\varepsilon$  32
- $\varepsilon$ -frei 39
- $\mu$ -Schema 21, 64
- $\chi_T$  18
- $\omega$  79
- $\dot{-}$  14
- $\rightarrow$  33
- $\doteq$  11
- $\equiv_L$  53, 57, 70
- $\Rightarrow$  33
- $\Rightarrow^*$  33
  
- abgeschlossene Menge 58, 61
- Ackermann-Funktion 15, 31
- ähnliche Strukturen 37, 51, 52, 67
- akzeptierender Zustand 37, 38
- Akzeptor
  - , deterministischer 40
  - , endlicher 37, 41, 70
  - , nichtdeterministischer 40, 42, 54
- algebraische Struktur 36, 51, 58, 70
- , mehrsortige 71, 74
- , verallgemeinerte 59, 61, 64
- Antwortfunktion 37, 52, 70
- Anweisungscode 79
- Atom 66, 67, 100
- Aufzählbarkeitssatz 81
- aussagenlogische Formel 68, 74
- Auswertung, symbolische 110
  
- berechenbar 97
- Berechenbarkeit, prinzipielle 76
- Bereich, reflexiver 77, 109
- busy beaver* 26
  
- charakteristische Funktion 18, 23, 26
- Chomsky, N. 34
  
- Church, These von 10, 16, 87
- Code 77
- Compilation 111
- Compiler 111
- Compiler-Generator 112
  
- Datenstruktur 11, 32
  - , reflexive 100
  - deterministischer Akzeptor 40
  - deterministische Turing-Maschine 95
  - Diagonalisierungsverfahren 29
  - dotted pair* 100
  
- eindeutige Lesbarkeit 66, 67
  - , Unlösbarkeit der 93
- endlicher Akzeptor 37, 41, 70
- endlicher Index, Kongruenz von 52, 53
- entscheidbares Prädikat 23
- Entscheidungsprobleme 50
- Erzeugnis 60, 61
- Evaluator, partieller 110
  
- Fixpunkt 58, 61, 63, 65, 84
  - , kleinster 106, 107, 108
- Fixpunktgleichung 103
- Fixpunktsatz der Rekursionstheorie 85, 98
- Fixpunktsemantik 103
- free-argument* 107
- freie  $m$ -Algebra 52, 67, 70
- full substitution* 107
- Funktion
  - , Ackermann- 15, 30
  - , charakteristische 18, 23, 26
  - , Graph einer 11, 28, 104
  - , monotone 58, 65
  - , partielle 10
  - , partiell-rekursive 22, 28, 63
  - , programmierbare 98
  - , primitiv-rekursive 17, 63, 66
  - , rekursive 22
  - , stetige 60, 65
- funktionale Menge 10, 104
- Funktionsvariable 102
  
- GOTO 88
  - , reduziertes 88
- Grammatik 33, 70
  - , kontextfreie 34, 93
  - , kontext-sensitive 35

- , links-reguläre 35, 56
- , nichtverkürzende 34
- , rechts-reguläre 35, 56
- , reguläre 34, 74
- Graph einer Funktion 11, 28, 104
- Gödelnumerierung 77
  
- Halteproblem 76, 82, 89
- Hauptsatz über syntaktische Strukturen 12, 52, 67
- Homomorphismus 51, 52, 54, 70
  
- Index 81
- , Kongruenz von endlichem 52, 53
- Induktion nach der Struktur 66
- induktive Struktur 66, 67
- Interpreter 111
- Iterationskette 60
- Iterationslemma 59
  
- k*-Kellerautomat 96
- Kellerautomat 96
- Kleene - Myhill, Satz von 47
- kleinster Fixpunkt 106, 107, 108
- Kompositionsschema 17, 64
- Konfigurationsfolge 83
- Kongruenz
- , von endlichem Index 52, 53
- , syntaktische 53, 57
- kontext-sensitive Grammatik 35
- kontextfreie Grammatik 34, 93
- kontextfreie Sprache 34, 68
- Kopf 94
  
- Label 78
- leftmost-innermost* 107
- leftmost-outermost* 107
- links-regulär 35, 56
- LISP 100
- Liste 100
- LOOP 11, 69
- LOOP-berechenbar 13, 66
- LOOP<sub>n</sub> 12, 34, 62, 66
  
- Maschine, selbstreproduzierende 85
- McCarthy, J. 100
- McCarthy-Schema 102
- mehrsortige Struktur 71, 74
- Menge
- , abgeschlossene 58, 61
- , funktionale Menge 10, 104
- , nicht-rekursive 82
- , rekursive 26
- Minimalakzeptor 54, 55, 57
- monoton 58, 65
  
- natürliche Fortsetzung 104
- Nerode, Satz von 53
- nichtdeterministischer Akzeptor 40, 42, 54
- nichtdeterministische Turing-Maschine 95
- Nichtterminalalphabet 33
- nichtverkürzende Grammatik 34
- Normalformensatz 82
- Numerierung 97
  
- parallel-innermost* 107
- parallel-outermost* 107
- Parametrisierungsproblem 84
- partiell-rekursive Funktion 22, 28, 63
- partielle Funktion 10
- Post'sches Korrespondenzproblem 90
- primitiv-rekursive Funktion 17, 63, 66
- primitiv-rekursives Prädikat 19
- prinzipielle Berechenbarkeit 76
- Prädikat
- , entscheidbares 23
- , primitiv-rekursives 19
- , rekursives 23
- Probleme
- , unentscheidbare 82
- , kombinatorische 88
- Produkt von Sprachen 44
- Produktion 33
- Produktionsgrammatik 33
- Programmcode 79
- Programme, LISP- 101
- Projektionsfunktion 16
- Prozedur, rekursive 100
- Pumping Lemma 49, 56
  
- Quellprogramm 111
  
- rechts-regulär 35, 56
- reflexiver Bereich 77, 109
- reflexive Datenstruktur 100
- reguläre Grammatik 34, 74
- reguläre Sprache 34, 41, 53
- regulärer Ausdruck 47, 69
- Rekursion 102
- , simultane 19, 62
- Rekursionsgleichung 103
- Rekursionsschema 17, 62, 64, 103, 104
- Rekursionstheorem 84
- Rekursionstheorie 81
- rekursiv aufzählbar 27, 28
- rekursiv aufzählbare Sprache 35
- rekursive Funktion 22
- rekursive Menge 26
- rekursive Sprache 35
- rekursives Prädikat 23
- rekursive Prozedur 100

- Residualprogramm 110
- Rice, Satz von 98
- Rosser, J. B. 28
  
- S-m-n*-Theorem 84, 109
- Schema der primitiven Rekursion 17
- Semantikfunktion 109
- Simulationssatz 80
- simultane Rekursion 19, 62
- Sprache 32
  - , Produkt von 44
  - , kontextfreie 34, 68
  - , reguläre 34, 41, 53
  - , rekursiv aufzählbare 35
  - , rekursive 35
- Sternoperator 44
- stetig 60, 65
- Strategie 107
- Struktur
  - , ähnliche 37, 51, 52, 67
  - , algebraische 36, 51, 58, 70
  - , mehrsortige 71, 74
  - , Induktion nach der 66
  - , induktive 66, 67
  - , syntaktische 52, 67, 68, 70
  - , verallgemeinerte algebraische 59, 61, 64
- symbolische Auswertung 110
- syntaktische Kategorie 11, 15, 34, 71
- syntaktische Kongruenz 53, 57
- syntaktische Struktur 52, 67, 68, 70
  - , Hauptsatz über 12, 52, 67
- Syntaxdiagramm 32, 71
  
- Tag-Maschine 88
- Term 104
- Terminalalphabet 33
- Turing, A. 87
- Turing-Maschine 35, 94
  - , deterministische 95
  - , nichtdeterministische 95
- Turing-Tabelle 94
  
- Unentscheidbarkeitssatz 82
- Universalmaschine 87
- Universalprogramm 79
  
- verallgemeinerte algebraische Struktur 59, 61, 64
  
- Warshall, Algorithmus von 48
- Wertecode 79
- WHILE 15, 69
- WHILE-berechenbar 15
- WHILE<sub>n</sub> 15
  
- Zielprogramm 111
  
- Zustand, akzeptierender 37, 38
- Zustandsdiagramm 37, 38

## Leitfäden der angewandten Informatik Fortsetzung

Kleine Büning/Schmitgen: **PROLOG**

304 Seiten. Kart. DM 34,—

Meier: **Methoden der grafischen und geometrischen Datenverarbeitung**

224 Seiten. Kart. DM 34,—

Meyer-Wegener: **Transaktionssysteme**

242 Seiten. DM 38,—

Mresse: **Information Retrieval – Eine Einführung**

280 Seiten. Kart. DM 38,—

Müller: **Entscheidungsunterstützende Endbenutzersysteme**

253 Seiten. Kart. DM 28,80

Mußtopf / Winter: **Mikroprozessor-Systeme**

Trends in Hardware und Software

302 Seiten. Kart. DM 32,—

Nebel: **CAD-Entwurfskontrolle in der Mikroelektronik**

211 Seiten. Kart. DM 32,—

Retti et al.: **Artificial Intelligence – Eine Einführung**

2. Aufl. X, 228 Seiten. Kart. DM 34,—

Schicker: **Datenübertragung und Rechnernetze**

2. Aufl. 242 Seiten. Kart. DM 32,—

Schmidt et al.: **Digitalschaltungen mit Mikroprozessoren**

2. Aufl. 208 Seiten. Kart. DM 25,80

Schmidt et al.: **Mikroprogrammierbare Schnittstellen**

223 Seiten. Kart. DM 34,—

Schneider: **Problemorientierte Programmiersprachen**

226 Seiten. Kart. DM 25,80

Schreiner: **Systemprogrammierung in UNIX**

Teil 1: Werkzeuge. 315 Seiten. Kart. DM 48,—

Teil 2: Techniken. 408 Seiten. Kart. DM 58,—

Singer: **Programmieren in der Praxis**

2. Aufl. 176 Seiten. Kart. DM 28,80

Specht: **APL-Praxis**

192 Seiten. Kart. DM 24,80

Vetter: **Aufbau betrieblicher Informationssysteme  
mittels konzeptioneller Datenmodellierung**

4. Aufl. 455 Seiten. Kart. DM 48,—

Weck: **Datensicherheit**

326 Seiten. Geb. DM 44,—

Wingert: **Medizinische Informatik**

272 Seiten. Kart. DM 25,80

Wißkirchen et al.: **Informationstechnik und Bürosysteme**

255 Seiten. Kart. DM 28,80

Wolf/Unkelbach: **Informationsmanagement in Chemie und Pharma**

244 Seiten. Kart. DM 34,—

Zehnder: **Informationssysteme und Datenbanken**

4. Aufl. 276 Seiten. Kart. DM 36,—

Zehnder: **Informatik-Projektentwicklung**

223 Seiten. Kart. DM 32,—

Preisänderungen vorbehalten



B. G. Teubner Stuttgart