
Inhaltsverzeichnis

1	Einleitung	1
2	Mathematische Grundlagen	9
2.1	Berechenbare Funktionen	9
2.1.1	Einleitung	9
2.1.2	Primitiv-rekursive Funktionen	11
2.1.3	Primitiv-rekursive Prädikate	14
2.1.4	Partiell rekursive Funktionen	16
2.1.5	Vergleich der betrachteten Klassen von Funktionen	18
2.1.6	Effektive Bereiche	19
2.2	Der λ -Kalkül	23
2.2.1	Einleitung	23
2.2.2	Der klassische λ -Kalkül	25
2.2.2.1	Elementare Begriffe	26
2.2.2.2	Reduktionsregeln des λ -Kalküls	31
2.2.2.3	Extensionale Gleichheit von Funktionen	32
2.2.2.4	Die Church-Rosser Eigenschaft	34
2.2.3	Wahrheitswerte und logische Verknüpfungen	39
2.2.4	Arithmetik und λ -Definierbarkeit	40
2.2.5	Terme mit undefinierter Bedeutung	45
2.2.6	Fixpunkte	48
2.2.7	Reduktionsstrategien	50
2.2.8	Angewandter λ -Kalkül	53
2.2.9	Typsysteme	54
2.2.9.1	Getypter λ -Kalkül	55
2.2.9.2	Abstrakte Datentypen	57
2.2.9.3	Polymorphie	60
2.3	Kombinatorische Logik	62
2.3.1	Einleitung	62
2.3.2	Elementare Begriffe	62

2.3.3	Die Beziehung zum λ -Kalkül	68
2.3.4	Anwendungen der kombinatorischen Logik	69
3	Programmiersprachen	73
3.1	FP-systeme	73
3.1.1	Einleitung	73
3.1.2	Schemasprache	74
3.1.3	Ein FP-System	76
3.1.4	Beispiele für FP-Programme	79
3.1.5	Die Algebra der FP-Programme	86
3.1.6	FFP-Systeme	92
3.1.7	Beispiele für FFP-Programme	94
3.1.8	FP-Programme als Kombinatoren	97
3.2	LISP	108
3.2.1	Einleitung	108
3.2.2	Pure-LISP	109
3.2.2.1	Daten	109
3.2.2.2	Basisfunktionen zur Verarbeitung von S-Ausdrücken	111
3.2.2.3	Vereinfachte Darstellung aufeinanderfolgender car's und cdr's	112
3.2.2.4	Basisprädikate für S-Ausdrücke	112
3.2.2.5	Bedingte Ausdrücke	113
3.2.2.6	Abstraktionen und Applikationen	113
3.2.2.7	Rekursive Lambda-Ausdrücke	114
3.2.2.8	Funktionale	114
3.2.2.9	Zusammenfassung der Syntax	115
3.2.2.10	Übersetzung von Programmen der M-Sprache in S-Ausdrücke	116
3.2.2.11	Beispiele	117
3.2.2.12	Der Interpretierer	121
3.2.2.13	Interpretation eines Beispiels	124
3.2.3	LISP-Programmiersysteme	126
3.2.3.1	Datenstrukturen	127
3.2.3.2	Pseudofunktionen	131
3.2.3.3	Standardfunktionen	133
3.2.3.4	Konzeptionelle Erweiterungen	134
3.2.3.5	Die INTERLISP-Programmierungsumgebung	137
3.2.4	Kuriositäten	141
3.2.5	Die Beziehung zum λ -Kalkül	143
3.2.5.1	LISP als angewandter λ -Kalkül	143
3.2.5.2	Der Interpretierer <i>eval1</i>	144
3.2.5.3	Der Interpretierer <i>eval2</i>	146
3.2.5.4	Statische und dynamische Bindung von Variablen	148

	3.2.5.5	Der Interpretierer <i>eval3</i>	149
	3.2.5.6	Der Interpretierer <i>eval4</i>	151
3.3		Weitere Applikative Programmiersprachen	154
	3.3.1	Einleitung	154
	3.3.2	SASL	155
	3.3.3	KRC	160
	3.3.4	EFPL	169
	3.3.5	BRL	172
	3.3.5.1	Sprachbeschreibung	172
	3.3.5.2	Ein Programm zur Unifikation von Termen . . .	179
	3.3.6	Scheme	187
	3.3.6.1	Datentypen	187
	3.3.6.2	Globale Definitionen	188
	3.3.6.3	Lokale Deklarationen	189
	3.3.6.4	Prozeduren	193
	3.3.6.5	Fallunterscheidungen	194
	3.3.6.6	Rekursionen	196
	3.3.6.7	Programmierbeispiele	197
	3.3.6.8	Scheme-Systeme	200
	3.3.7	Miranda	201
	3.3.8	Haskell	206
	3.3.8.1	Sprachkonzepte	207
	3.3.8.2	Das Typsystem	212
	3.3.8.3	Beispielprogramme	221
	3.3.9	ML	222
	3.3.9.1	Entwicklung	222
	3.3.9.2	Sprachelemente	223
	3.3.10	Hope	229
	3.3.11	Curry	234
	3.3.11.1	Einführung	234
	3.3.11.2	Sprachkonzepte	235
	3.3.11.3	Ein Programmbeispiel	237
	3.3.12	Weitere Sprachen	238
	3.3.12.1	ASpecT	238
	3.3.12.2	Caml	239
	3.3.12.3	Cayenne	239
	3.3.12.4	CELP	239
	3.3.12.5	Clean	239
	3.3.12.6	Eden	240
	3.3.12.7	Erlang	240
	3.3.12.8	Escher	241
	3.3.12.9	FALCON	241
	3.3.12.10	Goffin	242
	3.3.12.11	λ -Prolog	243
	3.3.12.12	L_λ	243

3.3.12.13	Leda	243
3.3.12.14	Mercury	244
3.3.12.15	Oz	245
3.3.12.16	Scala	245
3.3.12.17	TyPiCal	246
4	Implementierungstechniken	247
4.1	Interpretierer	247
4.1.1	Einleitung	247
4.1.2	Shallow-Binding	249
4.1.3	Optimierung von einfachen Postrekursionen	257
4.1.4	Optimierung von verdeckten Postrekursionen	260
4.2	Übersetzer	265
4.2.1	Einleitung	265
4.2.2	Ein Laufzeitsystem mit kellerartiger Speicherplatzverwaltung	267
4.2.3	Optimierungen	272
4.3	Hardware – Unterstützte Implementierungen	278
4.3.1	Einleitung	278
4.3.1.1	Reduktionsmaschinen	279
4.3.1.2	Datenflußmaschinen	283
4.3.2	Die GMD-Reduktionsmaschine (Berkling-Maschine)	291
4.3.2.1	Einleitung	291
4.3.2.2	Der interne Aufbau der GMD-Maschine	293
4.3.2.3	Kooperierende Reduktionsmaschinen	303
4.3.3	Die S-K-I-Graph-Reduktionsmaschine von Turner	307
4.3.3.1	Einleitung	307
4.3.3.2	Übersetzung von SASL-Programmen	308
4.3.3.3	Der Graph Reduktionsmechanismus	313
4.3.4	Die Manchester-Datenflußmaschine	324
4.3.4.1	Einleitung	324
4.3.4.2	Rechnerstruktur	330
	Literaturverzeichnis	337
	Sachverzeichnis	351

Funktionale und Applikative Programmierung
Grundlagen, Sprachen, Implementierungstechniken

Lippe, W.

2009, X, 353 S., Hardcover

ISBN: 978-3-540-89091-1