

Inhaltsverzeichnis

1.	Information und ihre Repräsentation	1
1.1	Der Begriff „Information“	3
1.2	Ein Beispiel für Information und Repräsentation: Aussageformen	6
1.2.1	Boolesche Terme	7
1.2.2	Die Boolesche Algebra der Wahrheitswerte	9
1.2.3	Interpretation von Booleschen Termen	11
1.2.4	Die Gesetze der Booleschen Algebra und der Booleschen Terme	13
1.2.5	Anwendung der Gesetze für Boolesche Terme: Reduktion	15
1.3	Information und ihre Repräsentation in Normalform	20
1.3.1	Der Übergang von der Repräsentation zur Information	20
1.3.2	Umformungen von Repräsentationen	21
1.3.3	Normalformen und eindeutige Normalformen	22
1.3.4	Zeichensequenzen	23
1.3.5	Formale Sprachen	25
2.	Rechenstrukturen und Algorithmen	27
2.1	Zum Begriff „Algorithmus“	27
2.1.1	Informelle Algorithmenbeschreibungen	28
2.1.2	Formale Algorithmenbeschreibung durch Textersetzung	31
2.1.3	Deterministische Textersetzungsalgorithmen	37
2.1.4	Durch Textersetzungsalgorithmen induzierte Abbildungen	39
2.2	Rechenstrukturen	41
2.2.1	Familien von Funktionen und Mengen als Rechenstrukturen	42
2.2.2	Signaturen	47
2.2.3	Grundterme	50
2.2.4	Rechnen mit Grundtermen: Formulare	52
2.2.5	Terme mit (freien) Identifikatoren	54
2.2.6	Interpretation von Termen mit (freien) Identifikatoren	55
2.2.7	Terme mit (freien) Identifikatoren als Formulare	56
2.3	Algorithmen als Termersetzungssysteme	57
2.3.1	Termersetzungsregeln	57
2.3.2	Termersetzungssysteme	58
2.3.3	Der Termersetzungsalgorithmus	59
2.3.4	Korrektheit von Termersetzungssystemen	61

2.4	Aussagenlogik und Prädikatenlogik	63
2.4.1	Aussagenlogik	64
2.4.2	Prädikatenlogik	68
3.	Programmiersprachen und Programmierung	73
3.1	Syntax: Beschreibung formaler Sprachen durch BNF	74
3.1.1	BNF-Notation	74
3.1.2	Syntaxdiagramme	78
3.1.3	Kontextbedingungen	79
3.2	Semantik	80
3.3	Zur Implementierung von Programmiersprachen	81
3.4	Methodik der Programmierung	81
3.4.1	Prinzipien der Programmierung	81
3.4.2	Rechenstrukturen	83
4.	Applikative Programmiersprachen	84
4.1	Elemente rein applikativer Programmiersprachen	84
4.1.1	Syntax von Ausdrücken und primitive Rechenstrukturen	86
4.1.2	Beschreibung der Bedeutung von Ausdrücken	86
4.1.3	Konstanten und Identifikatoren	89
4.1.4	Bedingte Ausdrücke	90
4.1.5	Funktionsapplikation	92
4.1.6	Bindung von (freien) Identifikatoren: Funktionsabstraktion	93
4.2	Erweiterung applikativer Sprachen: Deklarationen	99
4.2.1	Elementdeklaration	99
4.2.2	Deklarationen für Funktionen	102
4.3	Rekursive Funktionsdeklarationen	104
4.3.1	Induktive Deutung rekursiver Funktionsdeklarationen	109
4.3.2	Deutung als kleinster Fixpunkt	113
4.3.3	Rekursive Funktionsvereinbarungen in Termersetzungssystemen	119
4.3.4	Formulare für rekursive Funktionsdeklarationen	119
4.4	Rekursionsformen	120
4.4.1	Lineare Rekursion	121
4.4.2	Repetitive Rekursion	121
4.4.3	Kaskadenartige Rekursion	122
4.4.4	Vernestete Rekursion	124
4.4.5	Verschränkte Rekursion	125
4.5	Techniken applikativer Programmierung	126
4.5.1	Beschreibung der Aufgabenstellung: Spezifikation	126
4.5.2	Verallgemeinerung der Aufgabenstellung durch Einbettung	130
4.5.3	Zur Strukturierung von Aufgabenstellungen	132
4.5.4	Ableitung von Rekursionen aus Spezifikationen	133
4.5.5	Parameterunterdrückung, untergeordnete Rechenvorschriften	134
4.5.6	Effizienz applikativer Programme	136
4.5.7	Dokumentation	139

4.5.8	Test und Integration von Programmen	140
4.6	Beweistechniken für rekursive Programme	141
4.6.1	Induktion und Rekursion	141
4.6.2	Partielle Korrektheit	144
4.6.3	Terminierungsbeweise	145
5.	Zuweisungsorientierte Ablaufstrukturen	149
5.1	Zuweisungsorientierte Sprachelemente: Anweisungen	149
5.1.1	Syntax	150
5.1.2	Programmvariable und Zuweisung	150
5.1.3	Zustände	151
5.1.4	Funktionale Bedeutung von Anweisungen	152
5.1.5	Operationelle Semantik von Anweisungen	152
5.2	Einfache Anweisungen	153
5.2.1	Die „leere“ Anweisung nop	153
5.2.2	Die nichtterminierende Anweisung abort	153
5.2.3	Die Zuweisung	153
5.3	Zusammengesetzte Anweisungen	154
5.3.1	Sequentielle Komposition	155
5.3.2	Bedingte Anweisungen	156
5.3.3	Wiederholungsanweisungen	157
5.4	Variablendeklarationen und Blöcke	160
5.5	Prozeduren	163
5.5.1	Prozedurdeklaration	163
5.5.2	Prozeduraufruf	164
5.5.3	Globale Programmvariablen in Prozeduren	166
5.5.4	Rekursive Prozeduren	166
5.6	Abschnitte, Bindungen, Gültigkeit, Lebensdauer	167
5.7	Programmiertechniken für zuweisungsorientierte Programme	169
5.7.1	Die Methode der Zusicherung	170
5.7.2	Terminierungsbeweise	174
6.	Sortendeklarationen	176
6.1	Deklarationen von Sorten	176
6.1.1	Skalare Elemente: Enumeration	177
6.1.2	Direktes Produkt: Tupel	177
6.1.3	Direkte Summe: Variantensorten	180
6.1.4	Teilbereiche	184
6.2	Felder	184
6.2.1	Einstufige Felder	184
6.2.2	Felder in zuweisungsorientierten Sprachen: Selektives Ändern	186
6.2.3	Mehrstufige Felder und allgemeine Indexmengen	188
6.2.4	Dynamische und flexible Felder	189
6.3	Endliche Mengen als Rechenstrukturen	189

7.	Maschinennahe Sprachelemente: Sprünge und Referenzen ...	193
7.1	Kontrollfluß	193
7.1.1	Marken und Sprünge	193
7.1.2	Kontrollflußdiagramme	195
7.2	Referenzen und Zeiger	199
8.	Rekursive Sortendeklarationen	204
8.1	Sequenzartige Rechenstrukturen	204
8.1.1	Die Rechenstruktur der Sequenzen	204
8.1.2	Ein/Ausgabe und externe Rechenstrukturen	211
8.2	Baumartige Rechenstrukturen	215
8.3	Rekursive Vereinbarungen von Sorten	226
8.3.1	Induktive Deutung von rekursiven Sortendeklarationen	227
8.3.2	Fixpunktdeutung rekursiver Sortendeklarationen	228
8.3.3	Verwendung rekursiver Sortendeklarationen	228
8.4	Geflechte	229
8.4.1	Einfache Geflechte	229
8.4.2	Rekursiv vereinbarte Geflechtssorten	230
8.4.3	Implementierung von Sequenzen durch verkettete Listen	231
8.4.4	Zweifach verkettete Liste	232
8.4.5	Zyklische Geflechte	234
8.4.6	Gemeinsame Teilstrukturen	237
	Literaturangaben	243
	Stichwortverzeichnis	245