

# Inhaltsverzeichnis

<b>1</b>	<b>Historisches</b>	<b>1</b>
1.1	Sprache und Schrift . . . . .	1
1.2	Zahlen und Rechenmaschinen . . . . .	7
1.3	Informationstheorie . . . . .	15
<b>2</b>	<b>Computer</b>	<b>21</b>
2.1	Aussagenlogik . . . . .	21
2.2	Die Vollständigkeit der Aussagenlogik . . . . .	30
2.3	Selbstbezügliche Aussagen . . . . .	35
2.4	Universelle Maschinen . . . . .	44
2.5	Mikroprogrammierung . . . . .	50
2.6	"Feste Verdrahtung" . . . . .	57
2.7	Ein komplettes Computersystem . . . . .	67
<b>3</b>	<b>Programmiersprachen</b>	<b>73</b>
3.1	Assembler - Befehlsnamen und Adressen . . . . .	73
3.2	Basic - Anweisungen in Satzform . . . . .	81
3.3	Fortran - Module . . . . .	92
3.4	Algol - Rekursion . . . . .	101
3.5	Pascal - Datenstrukturen . . . . .	111
3.6	Noch einmal "feste Verdrahtung" . . . . .	121
3.7	Zwischenspiel: Simula, Smalltalk und C . . . . .	126
3.8	C++ - Klassen und Objekte . . . . .	135
3.9	Dynamische Speicherverwaltung . . . . .	147
<b>4</b>	<b>Compiler</b>	<b>159</b>
4.1	Die Backus-Naur-Form . . . . .	159
4.2	Syntaxtabellen . . . . .	165
4.3	Namenstabellen . . . . .	174
4.4	Die Analyse von Unterprogrammaufrufen . . . . .	186
4.5	Allgemeine Ausdrücke in Programmiersprachen . . . . .	191
4.6	Polymorphie . . . . .	199

4.7	Verbundanweisungen und Deklarationen . . . . .	205
4.8	Codegenerierung . . . . .	214
<b>5</b>	<b>”Künstliche Intelligenz”</b> . . . . .	<b>229</b>
5.1	Lisp . . . . .	229
5.2	Ein anderes Paradigma . . . . .	234
	<b>Index</b> . . . . .	<b>237</b>

# Tabellenverzeichnis

1.1	Der ASCII-Zeichensatz (ohne Steuerzeichen)	6
2.1	"und"-Verknüpfung	24
2.2	Liste aller binären Verknüpfungen	24
2.3	Die beiden Folgerungs-Richtungen	25
2.4	Eliminierbare Teilausdrücke	29
2.5	Allgemeine Wahrheitstabelle (für drei Teilaussagen)	31
2.6	Aufteilung der Wahrheitstabelle	32
2.7	Wahrheitstabelle der Auswahl	33
3.1	Rechenvorschrift eines binären Addierwerks	76
3.2	Zerlegte Rechenvorschrift eines binären Addierwerks	76
3.3	Maschinenprogramm	79
3.4	Assemblerprogramm	80
3.5	Bedingungsabfrage in Assembler: Sprung, falls $i > 10$	85
3.6	Addition über Stack	88
3.7	Addition über Stack	90
3.8	Normierung eines logischen Argumentes	91
3.9	Unterprogramme in Assembler	93
3.10	Unterprogramme in BASIC	94
3.11	Ein FORTRAN-Programm	96
3.12	Unterprogramm-Aufruf in Assembler: CALL <i>Verzweigung</i> (100, <i>erg</i> )	97
3.13	Einzeln übersetzte Unterprogramme in Maschinencode	99
3.14	ALGOL-Unterprogramm	108
3.15	Unterprogramm-Aufruf unter ALGOL	109
3.16	Funktionsdefinition und -aufruf unter ALGOL	110
3.17	Array- und Recorddefinitionen unter PASCAL	116
3.18	Zugriff auf Array- und Recordkomponenten unter PASCAL	119
3.19	Verwendung eines var-Parameters unter PASCAL	122
3.20	Operatordefinition unter C++	135
3.21	Polymorphe Unterprogramme unter C++	136
3.22	Klassendefinition unter C++	137

3.23	Klassendefinition unter C++ (ohne Vererbung)	138
3.24	Polymorphe Variablen unter C++	138
3.25	Virtuelle Methoden unter C++	141
3.26	Virtuelle Methoden unter C++ / Adressenvergabe	143
3.27	Virtuelle Methoden unter C++ / Aufrufmechanismus	145
3.28	Virtuelle Methoden unter C++ / Aufrufmechanismus mit Parametern	146
3.29	Werte, Adressen, Zeiger in Assembler	149
3.30	Objektlisten - Abstrakte Listenklasse	151
3.31	Konkrete Listeneinträge	152
4.1	Syntax-Tabelle	170
4.2	Restriktivere Syntax-Tabelle	170
4.3	Syntax-Tabelle mit Generator-Befehlen	173
4.4	Namens-Tabelle	176
4.5	Ein weiterer Name	176
4.6	"Lernende" Namens-Tabelle	177
4.7	"Lernende" Namens-Tabelle nach dem Lernvorgang	180
4.8	Syntaxtabelle für Funktionsaufrufe	187
4.9	Funktionsaufruftabelle mit Generator-Befehlen	190

# Abbildungsverzeichnis

2.1	Logische Schaltungen über Relais . . . . .	35
2.2	Speicherschaltkreis über Relais . . . . .	37
2.3	Flip-Flop über Relais . . . . .	38
2.4	Master-Slave-Flip-Flop über Relais . . . . .	41
2.5	Taktgenerator über Relais . . . . .	43
2.6	mikroprogrammierter von-Neumann-Computer . . . . .	57
2.7	Umriss eines "festverdrahteten" Computers . . . . .	62
3.1	Flußdiagramm-Symbole . . . . .	75
3.2	Schleifenbeispiel als Flußdiagramm . . . . .	75
3.3	Flußdiagramm von IF-THEN-ENDIF . . . . .	83
3.4	Flußdiagramm von IF-THEN-ELSE-ENDIF . . . . .	83
3.5	Flußdiagramm von DO-WHILE . . . . .	83
3.6	Flußdiagramm von WHILE-WEND . . . . .	84
3.7	Stack für Zwischenergebnisse . . . . .	87
3.8	Stackpointer und Basepointer . . . . .	105
3.9	Unterprogrammaufruf unter ALGOL . . . . .	106
3.10	Basepointer-relative Adressierung . . . . .	107
3.11	Ende eines Funktionsaufrufs unter ALGOL . . . . .	110
3.12	Die virtuelle Methodentabelle . . . . .	142
3.13	Objektliste . . . . .	150
3.14	Objektbaum . . . . .	153
3.15	Programmspeicheraufbau . . . . .	154
3.16	Programmspeicheraufbau . . . . .	154
3.17	Heap-Aufbau . . . . .	155
4.1	Mini-Compiler in PASCAL . . . . .	164
4.2	Aufbau des Compiler-Stacks . . . . .	172
4.3	Compiler-Stack während des Lernvorgangs . . . . .	179
4.4	Compiler-Stacks bei einem Unterprogramm-Aufruf . . . . .	189
4.5	Lernen von Unterprogramm-Aufrufen . . . . .	201
4.6	Relative Adressierung von Parametern . . . . .	222

# Weitere Verzeichnisse

## Schaltungsbeschreibungen

1	Flip-Flop . . . . .	37
2	Master-Slave-Flip-Flop . . . . .	41
3	Ringspeicher mit Resetleitung . . . . .	42
4	Multivibrator . . . . .	42
5	Primitives Rechenwerk, Teil 1 . . . . .	48
6	Primitives Rechenwerk, Teil 2 . . . . .	48
7	Datenaustausch mit einem Random Access Memory . . . . .	53
8	Adressdecodierung im RAM; Zwei-Bit-Speicher . . . . .	54
9	Adressdecodierung im RAM; Vier-Bit-Speicher . . . . .	54
10	Adressdecodierung im RAM; Acht-Bit-Speicher . . . . .	55
11	Zählwerk, Increment . . . . .	60
12	Zählwerk, Decrement . . . . .	60
13	Addierwerk . . . . .	77
14	ALU-Eingang . . . . .	123
15	ALU-Rechenwerk . . . . .	123
16	ALU-Ausgang . . . . .	124
17	Register-Eingänge und Datenbus . . . . .	124

## Signalbeschreibungen

1	Primitives Rechenwerk . . . . .	47
2	Einfacher RISC, Teil 1 . . . . .	63
3	Einfacher RISC, Teil 2 . . . . .	66
4	Einfacher RISC mit Addierwerk, Ergänzung . . . . .	77
5	Einfacher RISC mit Addierwerk, Bedingungsabfrage . . . . .	78
6	Einfacher RISC mit Stackpointer, Ergänzung . . . . .	89
7	RISC mit ALU, zusätzliche Signale . . . . .	125

## Befehlsbeschreibungen

1	Einfacher RISC, Teil 1 . . . . .	64
2	Einfacher RISC, Teil 2 . . . . .	67
3	Einfacher RISC mit Addierwerk, Ergänzung . . . . .	78
4	RISC mit ALU, Befehlsbeispiele . . . . .	125

## Syntaxbeispiele

1	Mathematischer Ausdruck . . . . .	161
2	Einfache Klammersausdrücke . . . . .	163
3	Umsetzung mit Regeln der Länge Zwei . . . . .	166
4	Umsetzung ohne Linksrekursion . . . . .	168
5	Umsetzung mit eindeutigen Regelköpfen . . . . .	169
6	Klammersausdrücke mit UPN-Ausgabe . . . . .	173
7	Eine Regel pro Name . . . . .	174
8	Eine Regel für alle Namen . . . . .	175
9	Scanner . . . . .	181
10	Funktionsaufrufe . . . . .	187
11	Parser, Pass 1 - Ausdrücke . . . . .	192
12	Parser, Pass 1 - Verbundanweisungen . . . . .	207
13	Parser, Pass 1 - Deklarationen . . . . .	210

## Compileralgorithmen

1	Grundalgorithmus . . . . .	171
2	Ausgabefunktion . . . . .	172
3	Lernende Namenstabelle . . . . .	177
4	Parameter zählen . . . . .	188
5	Spalten für Namen und ASCII-Zeichen einsparen . . . . .	191
6	Typanalyse in Ausdrücken, Fragment . . . . .	200
7	Typanalyse in Ausdrücken . . . . .	203
8	Labelgenerierung . . . . .	206
9	Generierung von Zugriffsoperatoren . . . . .	209