

# Inhaltsverzeichnis

<b>Vorwort zur 1. Auflage</b>	<b>xi</b>
<b>Vorwort zur 2. Auflage</b>	<b>xiii</b>
<b>Vorwort zur 3. Auflage</b>	<b>xiv</b>
<b>Vorwort zur 4. Auflage</b>	<b>xv</b>
<b>Vorwort zur 5. Auflage</b>	<b>xvi</b>
<b>Vorwort zur 6. Auflage</b>	<b>xvii</b>
<b>1 Einführung und Übersicht</b>	<b>1</b>
1.1 Ausgangspunkte für das Themengebiet . . . . .	1
1.2 Anwendungen theoretischer Erkenntnisse . . . . .	3
1.3 Stoffübersicht und -abgrenzung . . . . .	4
1.4 Externe Lernhilfen und Web-Seiten . . . . .	5
1.5 Allgemeine Bibliographische Hinweise . . . . .	7
<b>I Endliche Automaten und reguläre Sprachen</b>	<b>9</b>
<b>2 Endliche Automaten</b>	<b>11</b>
2.1 Deterministische endliche Automaten . . . . .	11
2.1.1 Beispiel: Der Automat $A_{\text{Eintritt}}$ . . . . .	12
2.1.2 Alphabete, Wörter, Sprachen . . . . .	15
2.1.3 Zustände und Zustandsübergänge . . . . .	21
2.1.4 Deterministische endliche Automaten und reguläre Sprachen .	22
2.1.5 Vollständige Automaten . . . . .	27
2.1.6 Zusammenfassung . . . . .	28
2.2 Nichtdeterministische endliche Automaten . . . . .	28
2.2.1 Definitionen . . . . .	28
2.2.2 Äquivalenz von deterministischen und nichtdeterministischen endlichen Automaten . . . . .	34

2.2.3	Zusammenfassung . . . . .	39
2.3	Endliche Automaten mit $\varepsilon$ -Übergängen . . . . .	39
2.3.1	Definitionen . . . . .	39
2.3.2	Äquivalenz von $\varepsilon$ -Automaten zu nichtdeterministischen endlichen Automaten . . . . .	40
2.3.3	Zusammenfassung . . . . .	45
2.4	Verallgemeinerte endliche Automaten . . . . .	45
2.4.1	Definitionen . . . . .	45
2.4.2	Äquivalenz von verallgemeinerten und endlichen Automaten . . . . .	46
2.4.3	Weitere Varianten endlicher Automaten . . . . .	46
2.5	Minimierung endlicher Automaten . . . . .	47
2.5.1	Isomorphie endlicher Automaten . . . . .	48
2.5.2	Der Satz von Myhill und Nerode . . . . .	49
2.5.3	Verfahren zur Minimierung endlicher Automaten . . . . .	52
2.6	Anwendungen endlicher Automaten . . . . .	57
2.6.1	Rechnersysteme und Systemprogrammierung . . . . .	58
2.6.2	Teilworterkennung . . . . .	59
2.6.3	Weitere Anwendungen . . . . .	63
2.7	Bibliographische Hinweise und Ergänzungen . . . . .	63
2.8	Übungen . . . . .	65
<b>3</b>	<b>Reguläre Sprachen</b>	<b>69</b>
3.1	Reguläre Ausdrücke . . . . .	70
3.1.1	Definitionen und Eigenschaften . . . . .	70
3.1.2	Anwendung regulärer Ausdrücke: Syntaxprüfung von Programmeingaben . . . . .	76
3.1.3	Äquivalenz von endlichen Automaten und regulären Ausdrücken . . . . .	78
3.1.4	Anwendung: Scanner-Generatoren . . . . .	83
3.1.5	Zusammenfassung . . . . .	86
3.2	Typ-3-Grammatiken . . . . .	86
3.2.1	Rechtslineare Grammatiken . . . . .	87
3.2.2	Linkslineare Grammatiken . . . . .	91
3.2.3	Äquivalenz rechtslinearer und linkslinearer Grammatiken . . . . .	91
3.2.4	Verallgemeinerte Typ-3-Grammatiken . . . . .	93
3.2.5	Äquivalenz von endlichen Automaten und Typ-3-Grammatiken . . . . .	94
3.2.6	Zusammenfassung . . . . .	96
3.3	Eigenschaften regulärer Sprachen . . . . .	97
3.3.1	Abschlusseigenschaften von $REG_{\Sigma}$ . . . . .	97
3.3.2	Das Pumping-Lemma für reguläre Sprachen . . . . .	106
3.3.3	Entscheidbarkeitsprobleme . . . . .	108
3.3.4	Grenzen endlicher Automaten . . . . .	111
3.4	Bibliographische Hinweise und Ergänzungen . . . . .	114
3.5	Übungen . . . . .	115

<b>4</b>	<b>Endliche Maschinen und Automatenetze</b>	<b>119</b>
4.1	Endliche Maschinen . . . . .	120
4.1.1	Erweiterung des endlichen Automaten $A_{\text{Eintritt}}$ . . . . .	120
4.1.2	Mealy-Maschinen . . . . .	122
4.1.3	Ein formales Vorgehensmodell bei der Problemlösung . . . . .	129
4.1.4	Moore-Maschinen . . . . .	132
4.1.5	Äquivalenz von Mealy- und Moore-Berechenbarkeit . . . . .	135
4.1.6	Grenzen endlicher Maschinen . . . . .	139
4.2	Mealy-Maschinen und Schaltwerke . . . . .	142
4.3	Endliche Transducer . . . . .	147
4.4	Beispiele für Automatenetze . . . . .	149
4.4.1	Synchrone Automaten: Zellulare Automaten . . . . .	149
4.4.2	Asynchrone Automaten: Petri-Netze . . . . .	154
4.4.3	Anwendungen und Varianten von Petri-Netzen . . . . .	163
4.5	Anwendungen endlicher Maschinen . . . . .	170
4.5.1	Software- und Systementwurf. Statecharts . . . . .	170
4.5.2	Modellierung von (Geschäfts-) Prozessen . . . . .	172
4.5.3	Elektronischer Handel . . . . .	176
4.6	Bibliographische Hinweise und Ergänzungen . . . . .	177
4.7	Übungen . . . . .	178

## II Kontextfreie Sprachen und Kellerautomaten 183

<b>5</b>	<b>Kontextfreie Sprachen</b>	<b>185</b>
5.1	Kontextfreie Grammatiken . . . . .	185
5.1.1	Beispiele und Definitionen . . . . .	186
5.1.2	Normalformen . . . . .	188
5.2	Eigenschaften kontextfreier Sprachen . . . . .	195
5.2.1	Mehrdeutigkeit . . . . .	195
5.2.2	Das Pumping-Lemma für kontextfreie Sprachen . . . . .	197
5.2.3	Abschlusseigenschaften . . . . .	200
5.3	Bibliographische Hinweise . . . . .	204
5.4	Übungen . . . . .	205
<b>6</b>	<b>Kellerautomaten</b>	<b>207</b>
6.1	Nichtdeterministische Kellerautomaten . . . . .	208
6.1.1	Grundlegende Definitionen . . . . .	208
6.1.2	Akzeptieren mit leerem Keller . . . . .	212
6.2	Äquivalenz von kontextfreien Grammatiken und Kellerautomaten . . . . .	213
6.3	Deterministische Kellerautomaten . . . . .	217
6.4	Bibliographische Hinweise . . . . .	221
6.5	Übungen . . . . .	221

<b>7</b>	<b>Anwendungen kontextfreier Sprachen</b>	<b>223</b>
7.1	Ableitungs- und Syntaxbäume . . . . .	223
7.2	Compilerbau . . . . .	226
7.3	Syntax von Programmiersprachen . . . . .	235
7.3.1	Erweiterte Backus-Naur-Form . . . . .	235
7.3.2	Syntaxdiagramme . . . . .	239
7.4	Reguläre Definitionen . . . . .	242
7.5	Beispielanwendung: XML . . . . .	243
7.6	Bibliographische Hinweise . . . . .	247
7.7	Übungen . . . . .	247
<b>III</b>	<b>Berechenbarkeit und Komplexität</b>	<b>249</b>
<b>8</b>	<b>Typ-1- und Typ-0-Sprachen</b>	<b>251</b>
8.1	Die Chomsky-Hierarchie . . . . .	251
8.1.1	Typ-1-Sprachen (kontextsensitive Sprachen) . . . . .	252
8.1.2	Typ-0-Sprachen (rekursiv-aufzählbare Sprachen) . . . . .	260
8.1.3	Die Hierarchie . . . . .	265
8.1.4	Das Wortproblem . . . . .	267
8.2	Turingautomaten . . . . .	269
8.2.1	Definitionen und Beispiele . . . . .	269
8.2.2	Varianten von Turingautomaten . . . . .	274
8.2.3	Äquivalenz von deterministischen und nichtdeterministischen Turingautomaten . . . . .	279
8.2.4	Linear beschränkte Automaten . . . . .	280
8.2.5	Äquivalenz zwischen Typ-1-Grammatiken und linear beschränk- ten Automaten . . . . .	281
8.2.6	Äquivalenz zwischen Typ-0-Grammatiken und Turingautoma- ten . . . . .	283
8.2.7	Entscheidbare Sprachen . . . . .	283
8.3	Zusammenfassung . . . . .	284
8.4	Bibliographische Hinweise . . . . .	287
8.5	Übungen . . . . .	288
<b>9</b>	<b>Berechenbarkeit</b>	<b>291</b>
9.1	Turing-Berechenbarkeit . . . . .	291
9.1.1	Definition und Beispiele . . . . .	292
9.1.2	Die Programmiersprache TURING . . . . .	296
9.2	Loop-, While- und Goto-Berechenbarkeit . . . . .	301
9.2.1	Die Programmiersprache LOOP . . . . .	301
9.2.2	Die Programmiersprache WHILE . . . . .	307
9.2.3	Die Programmiersprache GOTO . . . . .	309
9.3	Primitiv rekursive und $\mu$ -rekursive Funktionen . . . . .	312
9.3.1	Primitiv-rekursive Funktionen . . . . .	312

9.3.2	$\mu$ -rekursive Funktionen . . . . .	317
9.4	Die Churchsche These . . . . .	320
9.5	Die Ackermannfunktion . . . . .	326
9.6	Zusammenfassung . . . . .	329
9.7	Universelle Turingmaschinen . . . . .	330
9.7.1	Codierung von Turingmaschinen . . . . .	330
9.7.2	Nummerierung von Turingmaschinen . . . . .	333
9.7.3	Eine Standardnummerierung für $\mathcal{P}$ . . . . .	336
9.7.4	Fundamentale Anforderungen an Programmiersprachen . . . . .	338
9.7.5	Das utm-Theorem . . . . .	338
9.7.6	Das smn-Theorem . . . . .	339
9.7.7	Anwendungen von utm- und smn-Theorem . . . . .	341
9.7.8	Bedeutung von utm- und smn-Theorem . . . . .	345
9.8	Bibliographische Hinweise . . . . .	347
9.9	Übungen . . . . .	349
<b>10</b>	<b>Entscheidbarkeit</b>	<b>353</b>
10.1	Existenz unentscheidbarer Probleme . . . . .	353
10.2	Entscheidbare und semi-entscheidbare Sprachen . . . . .	354
10.3	Reduktion von Sprachen . . . . .	358
10.4	Entscheidbare und semi-entscheidbare Mengen . . . . .	359
10.5	Unentscheidbare Mengen . . . . .	360
10.5.1	Das Halteproblem . . . . .	360
10.5.2	Anwendungen des Halteproblems: Straßenbahnen, autonome Roboter und fahrerlose Autos . . . . .	363
10.5.3	Unentscheidbare Sprachen . . . . .	365
10.5.4	Der Satz von Rice . . . . .	366
10.5.5	Das Korrektheitsproblem . . . . .	368
10.5.6	Das Äquivalenzproblem . . . . .	369
10.5.7	Der erweiterte Satz von Rice . . . . .	369
10.5.8	Das Postsche Korrespondenzproblem . . . . .	372
10.5.9	Anwendungen des Postschen Korrespondenzproblems . . . . .	375
10.6	Zusammenfassung . . . . .	378
10.7	Bibliographische Hinweise . . . . .	378
10.8	Übungen . . . . .	379
<b>11</b>	<b>Komplexität</b>	<b>383</b>
11.1	Die O-Notation . . . . .	383
11.2	Komplexität von Algorithmen . . . . .	386
11.3	Wichtige Komplexitätsklassen . . . . .	389
11.4	Die Klassen P und NP . . . . .	390
11.4.1	Die Klasse P . . . . .	391
11.4.2	Die Klasse NP . . . . .	391
11.4.3	Die Klassen EXPTIME und NEXPTIME . . . . .	393
11.4.4	Das P-NP-Problem . . . . .	394

11.4.5	NP-Vollständigkeit . . . . .	394
11.5	Konkrete NP-vollständige Probleme . . . . .	397
11.5.1	SAT – Das Erfüllbarkeitsproblem der Aussagenlogik . . . . .	397
11.5.2	Weitere NP-vollständige Probleme . . . . .	402
11.6	Weitere Komplexitätsklassen . . . . .	410
11.6.1	Die Klasse PSPACE . . . . .	410
11.6.2	Komplementäre Komplexitätsklassen . . . . .	416
11.7	Zusammenfassung . . . . .	418
11.8	Bibliographische Hinweise und Ergänzungen . . . . .	419
11.9	Übungen . . . . .	420
<b>12</b>	<b>Approximative und probabilistische Ansätze und deren Anwendungen</b>	<b>423</b>
12.1	Approximative Algorithmen für NP-vollständige Probleme . . . . .	425
12.1.1	Approximierbarkeit . . . . .	425
12.1.2	Lokale Verbesserung am Beispiel <i>TSP</i> . . . . .	428
12.1.3	Untere Schranken für das Approximieren . . . . .	432
12.1.4	<i>TSP</i> in der Praxis . . . . .	433
12.2	Randomisierte Algorithmen und probabilistische Komplexitätsklassen	434
12.2.1	Die Klasse RP . . . . .	434
12.2.2	Die Klasse ZPP . . . . .	440
12.2.3	Die Klasse BPP . . . . .	443
12.2.4	Anwendung: Verschlüsselung . . . . .	444
12.3	Interaktive Beweissysteme . . . . .	453
12.4	Zero Knowledge Beweise. Anwendung: Authentifikation . . . . .	456
12.5	Probabilistisch überprüfbare Beweise . . . . .	459
12.6	Bemerkungen zur P-NP-Frage . . . . .	463
12.7	Zusammenfassung . . . . .	465
12.8	Bibliographische Hinweise und Ergänzungen . . . . .	466
12.9	Übungen . . . . .	468
	<b>Literaturverzeichnis</b>	<b>469</b>
	<b>Index</b>	<b>479</b>

Grundkurs Theoretische Informatik

Eine anwendungsbezogene Einführung - Für  
Studierende in allen Informatik-Studiengängen

Vossen, G.; Witt, K.-U.

2016, XVIII, 485 S. 142 Abb., Softcover

ISBN: 978-3-8348-1770-9