

Inhaltsverzeichnis

1	Reguläre Sprachen	1
1.1	Alphabet und Sprache	2
1.1.1	Alphabet	2
1.1.2	Wörter	3
1.1.3	Sprache	4
1.1.4	Maschinen	5
1.2	Deterministische endliche Automaten	5
1.2.1	Zustandsdiagramm	5
1.2.2	Formale Definition	6
1.2.3	Automat für Wörter gerader Länge	9
1.2.4	Automat für Ganzzahlen	10
1.2.5	TCP Zustandsdiagramm	11
1.3	Reguläre Sprachen	11
1.3.1	Definition	13
1.3.2	Endliche Sprachen sind regulär	14
1.3.3	Durch drei teilbare Binärzahlen sind regulär	15
1.4	Mengenoperationen — Produktautomat	16
1.4.1	Produktzustände und Produktübergänge	16
1.4.2	Produktautomat	17
1.4.3	Mengenoperation für reguläre Sprachen	18
1.5	Der Minimalautomat	20
1.5.1	Automaten mit zu vielen Zuständen	20
1.5.2	Zustände unterscheiden	21
1.5.3	Reduktion auf eine minimale Zustandsmenge	22
1.5.4	Automaten vergleichen	26
1.6	Rekonstruktion des Automaten aus der Sprache	27
1.6.1	Charakterisierung von Zuständen	27
1.6.2	Der Myhill-Nerode-Automat	28
1.6.3	Myhill-Nerode-Automat für Wörter gerader Länge	31
1.6.4	Myhill-Nerode-Automat für durch drei teilbare Binärzahlen	32
	Übungsaufgaben	35

2	Nicht reguläre Sprachen	37
2.1	Nicht reguläre Sprachen	37
2.1.1	Limitierungen endlicher Automaten	37
2.1.2	Die Sprache $\{0^n 1^n \mid n \geq 0\}$	38
2.2	Pumping-Lemma für reguläre Sprachen	40
2.2.1	Wörter als Wege durch einen endlichen Automaten	40
2.2.2	Pumpeigenschaft	41
2.2.3	Das Pumping-Lemma für reguläre Sprachen	42
2.2.4	Die Sprache $\{0^n 1^n \mid n \geq 0\}$	42
2.2.5	Die Sprache $\{w \in \{0, 1\}^* \mid w _0 = w _1\}$	44
2.3	Eine pumpbare, nicht reguläre Sprache	44
	Übungsaufgaben	45
3	Nichtdeterministische endliche Automaten	49
3.1	Nichtdeterminismus	49
3.1.1	Fehlende Übergänge	50
3.1.2	Nicht eindeutige Übergänge	51
3.1.3	Formale Definition	53
3.2	Umwandlung in einen DEA	56
3.2.1	Zustandsmengen als Zustände	56
3.2.2	Der Thompson-NEA	58
3.3	ϵ -Übergänge	62
3.3.1	Übergänge ohne Zeichenverarbeitung	62
3.3.2	Erreichbare Zustände	64
3.3.3	Thompson-NEA für einen NEA_ϵ	65
3.4	Probabilistische endliche Automaten	67
3.4.1	Probabilistische Übergänge	67
3.4.2	Definition eines probabilistischen endlichen Automaten	68
3.4.3	Vergleich mit der NEA-Implementation	68
	Übungsaufgaben	69
4	Reguläre Operationen und reguläre Ausdrücke	71
4.1	Reguläre Operationen	71
4.1.1	Die Alternative	72
4.1.2	Die Verkettung	73
4.1.3	Die *-Operation	74
4.1.4	Reguläre Operationen und reguläre Sprachen	75
4.1.5	Abgeleitete Operationen	76
4.2	Reguläre Ausdrücke	77
4.2.1	Zeichenketten zur Spezifikation von Sprachen	79
4.2.2	Primitive Ausdrücke	79
4.2.3	Reguläre Operationen	80
4.2.4	Weitere Metazeichen für reguläre Ausdrücke	82
4.3	Einen regulären Ausdruck in einen DEA umwandeln	84
4.3.1	Konstruktion eines NEA	84
4.3.2	Performance von Regex-Implementationen	86

4.4	Regulärer Ausdruck eines DEA	91
4.4.1	Verallgemeinerter NEA	91
4.4.2	1. Schritt: VNEA vorbereiten	92
4.4.3	2. Schritt: Zwischenzustände entfernen	93
4.4.4	Beispiel 1	94
4.4.5	Beispiel 2: Durch drei teilbare Binärzahlen	95
4.5	Anwendungen	97
4.5.1	Scanner-Generator flex	97
4.5.2	Ragel, ein Zustandsmaschinencompiler	98
4.5.3	Eingebettete lexikalische Analysatoren mit re2c(1)	101
	Übungsaufgaben	104
5	Kontextfreie Grammatiken und Sprachen	107
5.1	Kontextfreie Grammatik und kontextfreie Sprache	107
5.1.1	Warum reguläre Sprachen nicht reichen	108
5.1.2	Klammerausdrücke erzeugen	109
5.1.3	Formale Definitionen	111
5.1.4	Grammatik für die Sprache $0^n 1^n$	112
5.1.5	Grammatik für die Sprache $1^n + 1^m = 1^{m+n}$	113
5.2	Syntaxbaum	113
5.2.1	Konstruktion eines Syntaxbaumes	114
5.2.2	Nicht eindeutige Syntaxbäume	115
5.3	Auswertung	117
5.3.1	Expression-Term-Factor-Grammatik	117
5.3.2	Ausdrücke auswerten	118
5.3.3	Kontrollstrukturen	121
5.4	Reguläre Operationen	124
5.4.1	Reguläre Operationen für Grammatiken	124
5.4.2	Reguläre Sprachen sind kontextfrei	126
5.5	Chomsky-Normalform	127
5.5.1	Anforderungen an eine Normalform	127
5.5.2	Umwandlung in Chomsky-Normalform	129
5.5.3	Beispiel für die Umwandlung in Chomsky-Normalform	131
	Übungsaufgaben	133
6	Parsing	137
6.1	Spezifikation von Programmiersprachen	137
6.1.1	Geschichte	137
6.1.2	Extended Backus-Naur-Form	139
6.1.3	Beispiele von Grammatiken moderner Programmiersprachen	139
6.1.4	Syntaxanalyse auf der Basis einer Grammatik	143
6.2	Der Cocke-Younger-Kasami-Algorithmus	143
6.2.1	Ideen	143
6.2.2	Pseudocode	144
6.3	Der CYK-Algorithmus in Tabellenform	146
6.3.1	Ableitungstabelle	147

6.3.2	Rekursion wird zu Iteration	149
6.3.3	Laufzeit	151
	Übungsaufgaben	152
7	Stackautomaten	155
7.1	Stackspeicher	155
7.1.1	Stackoperationen	155
7.1.2	Stackautomaten	157
7.1.3	Die Sprachen $\emptyset^n 1^n$ und $\{w \mid w _0 = w _1\}$	158
7.1.4	Palindrome	159
7.1.5	Klammerausdrücke	160
7.2	Grammatik und Stackautomat	161
7.2.1	Stack als Zwischenspeicher für die Regelanwendung	161
7.2.2	Stackautomat zu einer Grammatik	162
7.3	LR-Parser	164
7.3.1	Warum Syntaxanalyse viel schwieriger ist	164
7.3.2	Ein deterministischer $O(n)$ -Parser	165
7.3.3	Parsergenerator Bison	171
7.4	Grammatik eines Stackautomaten	179
7.4.1	Regeln als Wege durch das Zustandsdiagramm	180
7.4.2	Vorbereitung des Stackautomaten	180
7.4.3	Ablesen von Regeln	182
7.4.4	Beispiele	184
	Übungsaufgaben	187
8	Nicht kontextfreie Sprachen	189
8.1	Pumping-Lemma	189
8.1.1	Syntaxbäume	189
8.1.2	Scherenschnitt in Syntaxbäumen	193
8.1.3	Pumpeigenschaft von kontextfreien Sprachen	195
8.2	Die Sprache $a^n b^n c^n$	196
	Übungsaufgaben	198
9	Abzählbar und überabzählbar unendlich	201
9.1	Eine unendliche Geschichte	201
9.2	Abzählbar und überabzählbar unendliche Mengen	204
9.2.1	Endlich und abzählbar unendlich	205
9.2.2	Mengenoperationen	207
9.2.3	Überabzählbar unendliche Mengen	210
	Übungsaufgaben	213
10	Turing-Maschinen	215
10.1	Vom Stackautomaten zur Turing-Maschine	215
10.1.1	Speicher	216
10.1.2	Berechnung	219
10.1.3	Zustandsdiagramm	221

10.1.4	Formale Definition	223
10.1.5	Beispiele von Turing-Maschinen	224
10.1.6	Berechnungsgeschichte	227
10.2	Varianten von Turing-Maschinen	229
10.2.1	Verschiedene Bandalphabete	229
10.2.2	Mehrspurmaschine	232
10.2.3	Einseitig unendliches Band	234
10.2.4	Zusätzliche Kopfbewegungen	235
10.2.5	Mehrbandmaschine	236
10.2.6	Turing-Maschinen und Programme	241
10.3	Turing-erkennbare Sprachen	241
10.3.1	Erkennbarkeit	241
10.3.2	Aufzählbare Sprachen	245
10.4	Die universelle Turing-Maschine	247
10.4.1	Moderne Computer und Turing-Maschinen	247
10.4.2	Ladbare Programme für Turing-Maschinen	249
10.4.3	Universelle Turing-Maschine in Game of Life	250
	Übungsaufgaben	253
11	Entscheidbarkeit	257
11.1	Algorithmen und Entscheidbarkeit	257
11.1.1	Hilberts Vortrag am ICM 1900	257
11.1.2	Moderne Formulierung	258
11.1.3	Entscheidbarkeit	259
11.1.4	Berechenbarkeit	262
11.2	Entscheidbare Probleme	263
11.2.1	Entscheidbare Probleme für reguläre Sprachen	263
11.2.2	Entscheidbare Probleme für kontextfreie Sprachen	265
11.3	Das Akzeptanzproblem für Turing-Maschinen	266
11.3.1	Das Lügner-Paradoxon	266
11.3.2	Das Akzeptanzproblem A_{TM} für Turing-Maschinen	268
11.4	Reduktion	269
11.4.1	Das Halteproblem	270
11.4.2	Vergleich von Sprachen: Reduktion	272
11.4.3	Implementationseigenschaften	273
11.5	Spracheigenschaften und der Satz von Rice	274
11.5.1	Spracheigenschaften	274
11.5.2	Leerheitsproblem	275
11.5.3	Regularitätsproblem	276
11.5.4	Nichttriviale Eigenschaften	278
11.5.5	Der Satz von Rice	278
11.6	Nicht entscheidbare Probleme und die Berechnungsgeschichte	280
11.6.1	Das Post-Korrespondenz-Problem	281
11.6.2	Das Gleichheitsproblem für kontextfreie Sprachen	284
	Übungsaufgaben	287

12 Komplexität	289
12.1 Laufzeit von Turing-Maschinen	290
12.1.1 Maßzahl für die Laufzeit	290
12.1.2 Polynomielle Laufzeit klassischer Maschinen	291
12.1.3 Polynomielle Reduktion	292
12.1.4 Vertex-Coloring und Stundenplan	293
12.2 Nichtdeterminismus	296
12.2.1 Nichtdeterministische Turing-Maschinen	297
12.2.2 Verifizierer	301
12.2.3 Polynomielle Ausfüllrätsel	304
12.3 Komplexitätsklassen P und NP	306
12.3.1 Polynomielle Probleme	306
12.3.2 Die Klasse NP	307
12.3.3 Das P vs. NP-Problem	308
Übungsaufgaben	309
13 NP-Vollständigkeit	313
13.1 NP-vollständige Sprachen	313
13.1.1 Die schwierigsten Probleme in der Klasse NP	313
13.1.2 Formale Definition	314
13.1.3 Alle NP-vollständigen Probleme sind äquivalent	315
13.1.4 Der Umgang mit NP-Vollständigkeit in der Praxis	316
13.2 Der Satz von Cook-Levin	318
13.2.1 SAT: Erfüllbarkeit für logische Formeln	318
13.2.2 Reduktion auf ein polynomielles Ausfüllrätsel	319
13.2.3 Reduktion auf SAT	322
13.2.4 3SAT ist NP-vollständig	322
13.3 Weitere NP-vollständige Probleme	324
13.3.1 3SAT und <i>k-CLIQUE</i>	324
13.3.2 Das Färbeproblem und das Stundenplanproblem	326
13.3.3 <i>SUBSET-SUM</i> und Kryptographie	329
13.3.4 <i>CIRCUIT</i> und <i>MINESWEEPER-CONSISTENCY</i>	337
13.4 Ein Katalog von NP-vollständigen Problemen	342
13.4.1 Der Reduktionsbaum von Karp	343
13.4.2 <i>BIP</i> : Binary Integer Programming	344
13.4.3 NP-vollständige Probleme zu Graph-Überdeckungen	345
13.4.4 NP-vollständige Probleme über Familien von Mengen	347
13.4.5 <i>FEEDBACK-NODE-SET</i> und <i>FEEDBACK-ARC-SET</i>	352
13.4.6 NP-vollständige Probleme über Pfade in Graphen	352
13.4.7 Probleme mit einer Aufteilung in zwei Teilmengen	353
13.4.8 <i>3D-MATCHING</i>	356
13.4.9 Kombinatorische Optimierungsprobleme	357
Übungsaufgaben	358

14 Programmiersprachen und Turing-Vollständigkeit	361
14.1 Turing-Vollständigkeit	361
14.1.1 Programmiersprachen	361
14.1.2 Turing-Vollständigkeit	362
14.1.3 Moderne Programmiersprachen	363
14.2 Die Sprache LOOP	365
14.2.1 Grundelemente	365
14.2.2 Die LOOP-Kontrollstruktur	366
14.2.3 Das Halteproblem für LOOP	369
14.2.4 Anwendungen	370
14.3 Die Sprache WHILE	373
14.3.1 Die WHILE-Schleife	373
14.3.2 WHILE als Erweiterung von LOOP	374
14.3.3 Turing-Vollständigkeit von WHILE	374
14.3.4 Brainfuck	375
14.3.5 Office	377
14.4 Die Sprache GOTO	377
14.4.1 Die GOTO-Sprache	377
14.4.2 WHILE-Programm in GOTO übersetzen	379
14.4.3 GOTO-Programm in WHILE übersetzen	379
14.4.4 Turing-Maschinensimulator in GOTO	380
Übungsaufgaben	383
15 Quantencomputer	387
15.1 Analogcomputer	387
15.1.1 Berechnung und natürliche Prozesse	388
15.1.2 Differentialgleichung und Analogcomputer	389
15.1.3 Quantenprozesse	389
15.2 Schaltungen und Quantenschaltungen	395
15.2.1 Universelle Quantencomputerbausteine	396
15.2.2 Die Hadamard-Operation	396
15.2.3 CNOT	397
15.2.4 Das Toffoli-Gate	398
15.2.5 Phasenverschiebung	398
15.2.6 Der Approximationssatz	399
15.2.7 Aufbau eines universellen Quantencomputers	399
15.3 Komplexitätsklassen	400
15.3.1 Probabilistische Algorithmen und BPP	400
15.3.2 Die Klasse BQP	406
A Grundlagen und Bezeichnungen	409
A.1 Logik	409
A.1.1 Prädikate	409
A.1.2 Logische Verknüpfungen	410
A.1.3 Rechenregeln	411
A.1.4 Normalformen	411

A.2	Mengen	411
A.2.1	Konstruktion von Mengen	412
A.2.2	Mengenoperationen	412
A.2.3	Kartesisches Produkt	415
A.2.4	Abbildungen	415
A.3	Graphen	416
A.3.1	Ungerichtete Graphen	416
A.3.2	Gerichtete Graphen	417
A.3.3	Beschriftete Graphen	417
A.4	Wahrscheinlichkeitsrechnung	418
A.4.1	Ereignisse	418
A.4.2	Wahrscheinlichkeit	419
A.4.3	Bedingte Wahrscheinlichkeit und Unabhängigkeit	419
A.4.4	Bernoulli-Experimente und Binomialverteilung	420
A.5	<i>O</i> -Notation	420
Literatur		423
Index		427