



**it**  
informatik

John E. Hopcroft  
Rajeev Motwani  
Jeffrey D. Ullman

# Einführung in Automatentheorie, Formale Sprachen und Berechenbarkeit

3., aktualisierte Auflage

**John E. Hopcroft,  
Rajeev Motwani,  
Jeffrey D. Ullman**

# **Einführung in Automatentheorie, Formale Sprachen und Berechenbarkeit**

**3., aktualisierte Auflage**



---

ein Imprint von Pearson Education  
München • Boston • San Francisco • Harlow, England  
Don Mills, Ontario • Sydney • Mexico City  
Madrid • Amsterdam

# Einführung in Automatentheorie, Formale Sprachen und Berechenbarkeit

## Inhaltsverzeichnis

Einführung in Automatentheorie, Formale Sprachen und Berechenbarkeit

Impressum

Inhaltsübersicht

Inhaltsverzeichnis

**Kapitel 1 Automaten: Die Grundlagen und Methoden 23**

1.1 Wozu dient das Studium der Automatentheorie? 25

1.1.1 Einführung in endliche Automaten 25

1.1.2 Strukturelle Repräsentationen 27

1.1.3 Automaten und Komplexität 28

1.2 Einführung in formale Beweise 28

1.2.1 Deduktive Beweise 29

1.2.2 Reduktion auf Definitionen 32

1.2.3 Andere Formen von Sätzen 34

1.2.4 Sätze, die keine Wenn-dann-Aussagen zu sein scheinen 37

1.3 Weitere Formen von Beweisen 37

1.3.1 Beweise der Äquivalenz von Mengen 38

1.3.2 Die Umkehrung 39

1.3.3 Beweis durch Widerspruch 41

1.3.4 Gegenbeispiele 41

1.4 Induktive Beweise 43

1.4.1 Induktive Beweise mit ganzen Zahlen 44

1.4.2 Allgemeinere Formen der Induktion mit ganzen Zahlen 47

1.4.3 Strukturelle Induktion 48

1.4.4 Gegenseitige Induktion 51

1.5 Die zentralen Konzepte der Automatentheorie 53

1.5.1 Alphabete 54

1.5.2 Zeichenreihen 54

1.5.3 Sprachen 56

1.5.4 Probleme 57

**Kapitel 2 Endliche Automaten 61**

2.1 Eine informelle Darstellung endlicher Automaten 63

2.1.1 Die Grundregeln 63

2.1.2 Das Protokoll 64

2.1.3 Die Automaten dazu befähigen, Eingaben zu ignorieren 66

2.1.4 Das gesamte System aus Automaten darstellen 68

2.1.5 Mithilfe des Produktautomaten die Gültigkeit des Protokolls überprüfen  
70

2.2 Deterministische endliche Automaten 71

2.2.1 Definition eines deterministischen endlichen Automaten 71

2.2.2 Wie ein DEA Zeichenreihen verarbeitet 72

2.2.3 Einfachere Notationen für DEAs 74

# Inhaltsverzeichnis

2.2.4 Die Übergangsfunktion auf Zeichenreihen erweitern 75

2.2.5 Die Sprache eines DEA 79

2.2.6 Übungen zum Abschnitt 2.2 79

## 2.3 Nichtdeterministische endliche Automaten 82

2.3.1 Eine informelle Sicht auf nichtdeterministische endliche Automaten 83

2.3.2 Definition nichtdeterministischer endlicher Automaten 85

2.3.3 Die erweiterte Übergangsfunktion 86

2.3.4 Die Sprache eines NEA 87

2.3.5 Äquivalenz deterministischer und nichtdeterministischer endlicher Automaten 88

2.3.6 Ein ungünstiger Fall für die Teilmengenkonstruktion 93

2.3.7 Übungen zum Abschnitt 2.3 95

## 2.4 Eine Anwendung: Textsuche 97

2.4.1 Zeichenreihen in Texten finden 97

2.4.2 Nichtdeterministische endliche Automaten für die Textsuche 98

2.4.3 Ein DEA, um die Menge von Schlüsselwörtern zu erkennen 99

2.4.4 Übungen zum Abschnitt 2.4 101

## 2.5 Endliche Automaten mit e-Übergängen 101

2.5.1 Verwendungen von e-Übergängen 102

2.5.2 Die formale Notation eines e-NEA 103

2.5.3 e-Hüllen 104

2.5.4 Erweiterte Übergänge und Sprachen für e-NEAs 105

2.5.5 e-Übergänge eliminieren 107

2.5.6 Übungen zum Abschnitt 2.5 110

## Kapitel 3 Reguläre Ausdrücke und Sprachen 113

### 3.1 Reguläre Ausdrücke 114

3.1.1 Die Operatoren regulärer Ausdrücke 115

3.1.2 Reguläre Ausdrücke bilden 117

3.1.3 Auswertungsreihenfolge der Operatoren regulärer Ausdrücke 120

3.1.4 Übungen zum Abschnitt 3.1 121

### 3.2 Endliche Automaten und reguläre Ausdrücke 122

3.2.1 Von DEAs zu regulären Ausdrücken 122

3.2.2 DEA durch die Eliminierung von Zuständen in reguläre Ausdrücke umwandeln 128

3.2.3 Reguläre Ausdrücke in Automaten umwandeln 134

3.2.4 Übungen zum Abschnitt 3.2 138

### 3.3 Anwendungen regulärer Ausdrücke 140

3.3.1 Reguläre Ausdrücke in Unix 140

3.3.2 Lexikalische Analyse 142

3.3.3 Textmuster finden 144

3.3.4 Übungen zum Abschnitt 3.3 146

### 3.4 Algebraische Gesetze für reguläre Ausdrücke 147

3.4.1 Assoziativität und Kommutativität 147

3.4.2 Identitäten und Annihilatoren 148

3.4.3 Distributivgesetze 149

3.4.4 Das Idempotenzgesetz 150

# Inhaltsverzeichnis

- 3.4.5 Gesetze bezüglich der Hüllenbildung 150
- 3.4.6 Gesetze für reguläre Ausdrücke entdecken 151
- 3.4.7 Test eines für reguläre Ausdrücke geltenden Gesetzes der Algebra 154
- 3.4.8 Übungen zum Abschnitt 3.4 156

## Kapitel 4 Eigenschaften regulärer Sprachen 159

- 4.1 Beweis der Nichtregularität von Sprachen 160
  - 4.1.1 Das Pumping-Lemma für reguläre Sprachen 161
  - 4.1.2 Anwendungen des Pumping-Lemmas 162
  - 4.1.3 Übungen zum Abschnitt 4.1 164
- 4.2 Abschluss-Eigenschaften regulärer Sprachen 166
  - 4.2.1 Abgeschlossenheit regulärer Sprachen bezüglich Boolescher Operationen 166
  - 4.2.2 Spiegelung 173
  - 4.2.3 Homomorphismus 174
  - 4.2.4 Inverser Homomorphismus 176
  - 4.2.5 Übungen zum Abschnitt 4.2 182
- 4.3 Entscheidbarkeits-Eigenschaften regulärer Sprachen 185
  - 4.3.1 Wechsel zwischen Repräsentationen 186
  - 4.3.2 Prüfen, ob eine reguläre Sprache leer ist 189
  - 4.3.3 Zugehörigkeit zu einer regulären Sprache prüfen 190
  - 4.3.4 Übungen zum Abschnitt 4.3 191
- 4.4 Äquivalenz und Minimierung von Automaten 191
  - 4.4.1 Prüfen, ob Zustände äquivalent sind 192
  - 4.4.2 Prüfen, ob reguläre Sprachen äquivalent sind 195
  - 4.4.3 Minimierung von DEAs 198
  - 4.4.4 Warum minimierte DEAs unschlagbar sind 201
  - 4.4.5 Übungen zum Abschnitt 4.4 203

## Kapitel 5 Kontextfreie Grammatiken und Sprachen 205

- 5.1 Kontextfreie Grammatiken 206
  - 5.1.1 Ein informelles Beispiel 206
  - 5.1.2 Definition kontextfreier Grammatiken 208
  - 5.1.3 Ableitungen mithilfe einer Grammatik 210
  - 5.1.4 Links- und rechtsseitige Ableitungen 213
  - 5.1.5 Die Sprache einer Grammatik 215
  - 5.1.6 Satzformen 216
  - 5.1.7 Übungen zum Abschnitt 5.1 217
- 5.2 Parse-Bäume 219
  - 5.2.1 Parse-Bäume aufbauen 219
  - 5.2.2 Der Ergebnis eines Parse-Baums 221
  - 5.2.3 Inferenz, Ableitungen und Parse-Bäume 222
  - 5.2.4 Von Inferenzen zu Bäumen 223
  - 5.2.5 Von Bäumen zu Ableitungen 225
  - 5.2.6 Von Ableitungen zu rekursiven Inferenzen 228
  - 5.2.7 Übungen zum Abschnitt 5.2 230
- 5.3 Anwendungen kontextfreier Grammatiken 231
  - 5.3.1 Parser 231

# Inhaltsverzeichnis

- 5.3.2 Der YACC-Parsergenerator 234
- 5.3.3 Markup-Sprachen 235
- 5.3.4 XML und Dokumenttypdefinitionen 238
- 5.3.5 Übungen zum Abschnitt 5.3 244

## 5.4 Mehrdeutigkeit von Grammatiken und Sprachen 245

- 5.4.1 Mehrdeutige Grammatiken 246
- 5.4.2 Mehrdeutigkeit aus Grammatiken tilgen 248
- 5.4.3 Linksseitige Ableitungen als Möglichkeit zur Beschreibung von Mehrdeutigkeit 251
- 5.4.4 Inhärente Mehrdeutigkeit 252
- 5.4.5 Übungen zum Abschnitt 5.4 255

## Kapitel 6 Pushdown-Automaten 259

### 6.1 Definition des Pushdown-Automaten 260

- 6.1.1 Informelle Einführung 260
- 6.1.2 Die formale Definition von Pushdown-Automaten 262
- 6.1.3 Eine grafische Notation für PDAs 264
- 6.1.4 Unmittelbare Beschreibungen eines PDA 265
- 6.1.5 Übungen zum Abschnitt 6.1 269

### 6.2 Die Sprachen eines PDA 270

- 6.2.1 Akzeptanz durch Endzustand 270
- 6.2.2 Akzeptanz durch leeren Stack 272
- 6.2.3 Vom leeren Stack zum Endzustand 272
- 6.2.4 Vom Endzustand zum leeren Stack 276
- 6.2.5 Übungen zum Abschnitt 6.2 278

### 6.3 Äquivalenz von PDAs und kontextfreien Grammatiken 279

- 6.3.1 Von Grammatiken zu PDAs 280
- 6.3.2 Von PDAs zu Grammatiken 283
- 6.3.3 Übungen zum Abschnitt 6.3 288

### 6.4 Deterministische Pushdown-Automaten 289

- 6.4.1 Definition eines deterministischen PDA 290
- 6.4.2 Reguläre Sprachen und deterministische PDAs 291
- 6.4.3 DPDAs und kontextfreie Sprachen 292
- 6.4.4 DPDAs und mehrdeutige Grammatiken 293
- 6.4.5 Übungen zum Abschnitt 6.4 294

## Kapitel 7 Eigenschaften kontextfreier Sprachen 297

### 7.1 Normalformen kontextfreier Grammatiken 298

- 7.1.1 Eliminierung unnützer Symbole 298
- 7.1.2 Berechnung der erzeugenden und erreichbaren Symbole 301
- 7.1.3 e-Produktionen eliminieren 302
- 7.1.4 Einheitsproduktionen eliminieren 306
- 7.1.5 Chomsky-Normalform 311
- 7.1.6 Übungen zum Abschnitt 7.1 316

### 7.2 Das Pumping-Lemma für kontextfreie Sprachen 319

- 7.2.1 Die Größe von Parse-Bäumen 319
- 7.2.2 Aussage des Pumping-Lemmas 320
- 7.2.3 Anwendungen des Pumping-Lemmas für kontextfreie Sprachen 323

# Inhaltsverzeichnis

7.2.4 Übungen zum Abschnitt 7.2 326

## 7.3 Abschluss-Eigenschaften kontextfreier Sprachen 328

7.3.1 Substitutionen 328

7.3.2 Anwendungen des Substitutions-Theorems 331

7.3.3 Spiegelung 332

7.3.4 Durchschnitt mit einer regulären Sprache 332

7.3.5 Inverse Homomorphismen 337

7.3.6 Übungen zum Abschnitt 7.3 339

## 7.4 Entscheidbarkeits-Eigenschaften kontextfreier Sprachen 341

7.4.1 Komplexität der Umwandlung von kFGs in PDAs und umgekehrt 342

7.4.2 Ausführungszeit der Umwandlung in Chomsky- Normalform 343

7.4.3 Prüfen, ob eine kontextfreie Sprache leer ist 345

7.4.4 Die Zugehörigkeit zu einer kontextfreien Sprache prüfen 347

7.4.5 Vorschau auf unentscheidbare kFL-Probleme 351

7.4.6 Übungen zum Abschnitt 7.4 352

## Kapitel 8 Einführung in Turing-Maschinen 355

### 8.1 Probleme, die Computer nicht lösen können 356

8.1.1 Programme, die »Hello, World« ausgeben 357

8.1.2 Der hypothetische »Hello, World«-Tester 359

8.1.3 Ein Problem auf ein anderes Problem reduzieren 362

8.1.4 Übungen zum Abschnitt 8.1 365

### 8.2 Die Turing-Maschine 366

8.2.1 Das Streben danach, alle mathematischen Fragen zu entscheiden 367

8.2.2 Die Notation der Turing-Maschine 368

8.2.3 Unmittelbare Beschreibungen für Turing-Maschinen 369

8.2.4 Übergangsdigramme für Turing-Maschinen 373

8.2.5 Die Sprache einer Turing-Maschine 376

8.2.6 Turing-Maschinen und das Halteproblem 377

8.2.7 Übungen zum Abschnitt 8.2 378

### 8.3 Programmiertechniken für Turing-Maschinen 379

8.3.1 Speicher im Zustand 380

8.3.2 Mehrere Spuren 381

8.3.3 Unterprogramme 383

8.3.4 Übungen zum Abschnitt 8.3 386

### 8.4 Erweiterungen für die einfache Turing-Maschine 386

8.4.1 Turing-Maschinen mit mehreren Bändern 386

8.4.2 Äquivalenz zwischen ein- und mehrbändigen TMn 388

8.4.3 Ausführungszeit und die Viele-Bänder-in-eins-Konstruktion 390

8.4.4 Nichtdeterministische Turing-Maschinen 391

8.4.5 Übungen zum Abschnitt 8.4 393

### 8.5 Beschränkte Turing-Maschinen 396

8.5.1 Turing-Maschinen mit semi-unendlichen Bändern 397

8.5.2 Maschinen mit mehreren Stacks 400

8.5.3 Zählermaschinen 403

8.5.4 Die Leistungsfähigkeit von Zählermaschinen 404

8.5.5 Übungen zum Abschnitt 8.5 406

# Inhaltsverzeichnis

## 8.6 Turing-Maschinen und Computer 407

- 8.6.1 Eine Turing-Maschine mit einem Computer simulieren 407
- 8.6.2 Einen Computer mit einer Turing-Maschine simulieren 409
- 8.6.3 Laufzeitvergleich zwischen Computern und Turing-Maschinen 413

## Kapitel 9 Unentscheidbarkeit 419

### 9.1 Eine nicht rekursiv aufzählbare Sprache 421

- 9.1.1 Binärzeichenreihen aufzählen 421
- 9.1.2 Codes für Turing-Maschinen 422
- 9.1.3 Die Diagonalisierungssprache 423
- 9.1.4 Der Beweis, dass  $L_d$  nicht rekursiv aufzählbar ist 425
- 9.1.5 Übungen zum Abschnitt 9.1 425

### 9.2 Ein unentscheidbares Problem, das rekursiv aufzählbar ist 426

- 9.2.1 Rekursive Sprachen 426
- 9.2.2 Komplemente rekursiver und rekursiv aufzählbarer Sprachen 427
- 9.2.3 Die universelle Sprache 430
- 9.2.4 Unentscheidbarkeit der universellen Sprache 433
- 9.2.5 Übungen zum Abschnitt 9.2 434

### 9.3 Unentscheidbare Probleme über Turing-Maschinen 436

- 9.3.1 Reduktionen 436
- 9.3.2 Turing-Maschinen, die die leere Sprache akzeptieren 438
- 9.3.3 Der Satz von Rice und Eigenschaften der rekursiv aufzählbaren Sprachen 441
- 9.3.4 Probleme bezüglich Spezifikationen von Turing-Maschinen 444
- 9.3.5 Übungen zum Abschnitt 9.3 444

### 9.4 Das Postsche Korrespondenz-Problem 446

- 9.4.1 Definition des Postschen Korrespondenz-Problems 446
- 9.4.2 Das »modifizierte« PKP 449
- 9.4.3 Fertigstellung des Beweises der PKP-Unentscheidbarkeit 452
- 9.4.4 Übungen zum Abschnitt 9.4 458

### 9.5 Andere unentscheidbare Probleme 459

- 9.5.1 Probleme bei Programmen 459
- 9.5.2 Unentscheidbarkeit der Mehrdeutigkeit kontextfreier Grammatiken 459
- 9.5.3 Das Komplement einer Listensprache 462
- 9.5.4 Übungen zum Abschnitt 9.5 465

## Kapitel 10 Nicht handhabbare Probleme 469

### 10.1 Die Klassen P und NP 471

- 10.1.1 Mit polynomialem Zeitaufwand lösbare Probleme 471
- 10.1.2 Beispiel: Der Kruskal-Algorithmus 472
- 10.1.3 Nichtdeterministischer polynomialer Zeitaufwand 476
- 10.1.4 Ein NP-Beispiel: Das Problem des Handlungsreisenden 477
- 10.1.5 Polynomzeit-Reduktionen 478
- 10.1.6 NP-vollständige Probleme 480
- 10.1.7 Übungen zum Abschnitt 10.1 482

### 10.2 Ein NP-vollständiges Problem 483

- 10.2.1 Das Erfüllbarkeitsproblem 484
- 10.2.2 SAT-Instanzen repräsentieren 485



# Inhaltsverzeichnis

10.2.3 NP-Vollständigkeit des SAT-Problems 486

10.2.4 Übungen zum Abschnitt 10.2 493

## 10.3 Ein eingeschränktes Erfüllbarkeitsproblem 493

10.3.1 Normalformen für Boolesche Ausdrücke 494

10.3.2 Ausdrücke in KNF konvertieren 495

10.3.3 NP-Vollständigkeit von CSAT 498

10.3.4 NP-Vollständigkeit von 3SAT 503

10.3.5 Übungen zum Abschnitt 10.3 504

## 10.4 Weitere NP-vollständige Probleme 505

10.4.1 NP-vollständige Probleme beschreiben 506

10.4.2 Das Problem unabhängiger Mengen 506

10.4.3 Das Problem der Knotenüberdeckung 511

10.4.4 Das Problem des gerichteten Hamiltonschen Kreises 512

10.4.5 Ungerichtete Hamiltonsche Kreise und das Problem des Handlungsreisenden 519

10.4.6 Zusammenfassung NP-vollständiger Probleme 521

10.4.7 Übungen zum Abschnitt 10.4 521

## Kapitel 11 Zusätzliche Problemklassen 529

### 11.1 Komplemente von Sprachen, die in NP enthalten sind 531

11.1.1 Die Sprachklasse Co-NP 531

11.1.2 NP-vollständige Probleme und Co-NP 532

11.1.3 Übungen zum Abschnitt 11.1 533

### 11.2 Probleme, die mit polynomialem Speicherplatz lösbar sind 534

11.2.1 Turing-Maschinen mit polynomialer Platzbegrenzung 534

11.2.2 Beziehung von PS und NPS zu früher definierten Klassen 535

11.2.3 Deterministischer und nichtdeterministischer polynomialer Speicherplatz 537

### 11.3 Ein für PS vollständiges Problem 540

11.3.1 PS-Vollständigkeit 540

11.3.2 Quantifizierte Boolesche Formeln 541

11.3.3 Quantifizierte Boolesche Formeln auswerten 542

11.3.4 PS-Vollständigkeit des QBF-Problems 544

11.3.5 Übungen zum Abschnitt 11.3 550

### 11.4 Sprachklassen basierend auf Randomisierung 550

11.4.1 Quicksort: Ein Beispiel für einen zufallsabhängigen Algorithmus 551

11.4.2 Ein auf Zufallsabhängigkeit basierendes Modell einer Turing-Maschine 552

11.4.3 Die Sprache einer zufallsabhängigen Turing-Maschine 554

11.4.4 Die Klasse RP 556

11.4.5 In RP enthaltene Sprachen erkennen 558

11.4.6 Die Klasse ZPP 559

11.4.7 Beziehung zwischen RP und ZPP 560

11.4.8 Beziehungen zu den Klassen P und NP 562

### 11.5 Die Komplexität des Primzahltests 562

11.5.1 Die Bedeutung des Primzahltests 563

11.5.2 Einführung in Modular-Arithmetik 565

11.5.3 Die Komplexität modular-arithmetischer Berechnungen 567

# Inhaltsverzeichnis

11.5.4 Zufallsabhängig-polynomiales Primzahl-Testen 568

11.5.5 Nichtdeterministische Primzahltests 570

11.5.6 Übungen zum Abschnitt 11.5 573

## Vorwort

Verwendung des Buches

Voraussetzungen

Übungen

Unterstützung im World Wide Web

Danksagung

## Vorwort zur deutschen Auflage

## 1 Automaten: Die Grundlagen und Methoden

### 1.1 Wozu dient das Studium der Automatentheorie?

1.1.1 Einführung in endliche Automaten

1.1.2 Strukturelle Repräsentationen

1.1.3 Automaten und Komplexität

### 1.2 Einführung in formale Beweise

1.2.1 Deduktive Beweise

1.2.2 Reduktion auf Definitionen

1.2.3 Andere Formen von Sätzen

1.2.4 Sätze, die keine Wenn-dann-Aussagen zu sein scheinen

### 1.3 Weitere Formen von Beweisen

1.3.1 Beweise der Äquivalenz von Mengen

1.3.2 Die Umkehrung

1.3.3 Beweis durch Widerspruch

1.3.4 Gegenbeispiele

### 1.4 Induktive Beweise

1.4.1 Induktive Beweise mit ganzen Zahlen

1.4.2 Allgemeinere Formen der Induktion mit ganzen Zahlen

1.4.3 Strukturelle Induktion

1.4.4 Gegenseitige Induktion

### 1.5 Die zentralen Konzepte der Automatentheorie

1.5.1 Alphabete

1.5.2 Zeichenreihen

1.5.3 Sprachen

1.5.4 Probleme

## 2 Endliche Automaten

### 2.1 Eine informelle Darstellung endlicher Automaten

2.1.1 Die Grundregeln

2.1.2 Das Protokoll

2.1.3 Die Automaten dazu befähigen, Eingaben zu ignorieren

# Inhaltsverzeichnis

2.1.4 Das gesamte System aus Automaten darstellen

2.1.5 Mithilfe des Produktautomaten die Gültigkeit des Protokolls überprüfen

## 2.2 Deterministische endliche Automaten

2.2.1 Definition eines deterministischen endlichen Automaten

2.2.2 Wie ein DEA Zeichenreihen verarbeitet

2.2.3 Einfachere Notationen für DEAs

2.2.4 Die Übergangsfunktion auf Zeichenreihen erweitern

2.2.5 Die Sprache eines DEA

2.2.6 Übungen zum Abschnitt 2.2

## 2.3 Nichtdeterministische endliche Automaten

2.3.1 Eine informelle Sicht auf nichtdeterministische endliche Automaten

2.3.2 Definition nichtdeterministischer endlicher Automaten

2.3.3 Die erweiterte Übergangsfunktion

2.3.4 Die Sprache eines NEA

2.3.5 Äquivalenz deterministischer und nichtdeterministischer endlicher Automaten

2.3.6 Ein ungünstiger Fall für die Teilmengenkonstruktion

2.3.7 Übungen zum Abschnitt 2.3

## 2.4 Eine Anwendung: Textsuche

2.4.1 Zeichenreihen in Texten finden

2.4.2 Nichtdeterministische endliche Automaten für die Textsuche

2.4.3 Ein DEA, um die Menge von Schlüsselwörtern zu erkennen

2.4.4 Übungen zum Abschnitt 2.4

## 2.5 Endliche Automaten mit e-Übergängen

2.5.1 Verwendungen von e-Übergängen

2.5.2 Die formale Notation eines e-NEA

2.5.3 e-Hüllen

2.5.4 Erweiterte Übergänge und Sprachen für e-NEAs

2.5.5 e-Übergänge eliminieren

2.5.6 Übungen zum Abschnitt 2.5

## 3 Reguläre Ausdrücke und Sprachen

### 3.1 Reguläre Ausdrücke

3.1.1 Die Operatoren regulärer Ausdrücke

3.1.2 Reguläre Ausdrücke bilden

3.1.3 Auswertungsreihenfolge der Operatoren regulärer Ausdrücke

3.1.4 Übungen zum Abschnitt 3.1

### 3.2 Endliche Automaten und reguläre Ausdrücke

3.2.1 Von DEAs zu regulären Ausdrücken

3.2.2 DEA durch die Eliminierung von Zuständen in reguläre Ausdrücke umwandeln

3.2.3 Reguläre Ausdrücke in Automaten umwandeln

# Inhaltsverzeichnis

3.2.4 Übungen zum Abschnitt 3.2

## 3.3 Anwendungen regulärer Ausdrücke

3.3.1 Reguläre Ausdrücke in Unix

3.3.2 Lexikalische Analyse

3.3.3 Textmuster finden

3.3.4 Übungen zum Abschnitt 3.3

## 3.4 Algebraische Gesetze für reguläre Ausdrücke

3.4.1 Assoziativität und Kommutativität

3.4.2 Identitäten und Annihilatoren

3.4.3 Distributivgesetze

3.4.4 Das Idempotenzgesetz

3.4.5 Gesetze bezüglich der Hüllenbildung

3.4.6 Gesetze für reguläre Ausdrücke entdecken

3.4.7 Test eines für reguläre Ausdrücke geltenden Gesetzes der Algebra

3.4.8 Übungen zum Abschnitt 3.4

## 4 Eigenschaften regulärer Sprachen

### 4.1 Beweis der Nichtregularität von Sprachen

4.1.1 Das Pumping-Lemma für reguläre Sprachen

4.1.2 Anwendungen des Pumping-Lemmas

4.1.3 Übungen zum Abschnitt 4.1

### 4.2 Abschluss-Eigenschaften regulärer Sprachen

4.2.1 Abgeschlossenheit regulärer Sprachen bezüglich Boolescher Operationen

4.2.2 Spiegelung

4.2.3 Homomorphismus

4.2.4 Inverser Homomorphismus

4.2.5 Übungen zum Abschnitt 4.2

### 4.3 Entscheidbarkeits-Eigenschaften regulärer Sprachen

4.3.1 Wechsel zwischen Repräsentationen

4.3.2 Prüfen, ob eine reguläre Sprache leer ist

4.3.3 Zugehörigkeit zu einer regulären Sprache prüfen

4.3.4 Übungen zum Abschnitt 4.3

### 4.4 Äquivalenz und Minimierung von Automaten

4.4.1 Prüfen, ob Zustände äquivalent sind

4.4.2 Prüfen, ob reguläre Sprachen äquivalent sind

4.4.3 Minimierung von DEAs

4.4.4 Warum minimierte DEAs unschlagbar sind

4.4.5 Übungen zum Abschnitt 4.4

## 5 Kontextfreie Grammatiken und Sprachen

### 5.1 Kontextfreie Grammatiken

5.1.1 Ein informelles Beispiel

5.1.2 Definition kontextfreier Grammatiken

# Inhaltsverzeichnis

5.1.3 Ableitungen mithilfe einer Grammatik

5.1.4 Links- und rechtsseitige Ableitungen

5.1.5 Die Sprache einer Grammatik

5.1.6 Satzformen

5.1.7 Übungen zum Abschnitt 5.1

## 5.2 Parse-Bäume

5.2.1 Parse-Bäume aufbauen

5.2.2 Der Ergebnis eines Parse-Baums

5.2.3 Inferenz, Ableitungen und Parse-Bäume

5.2.4 Von Inferenzen zu Bäumen

5.2.5 Von Bäumen zu Ableitungen

5.2.6 Von Ableitungen zu rekursiven Inferenzen

5.2.7 Übungen zum Abschnitt 5.2

## 5.3 Anwendungen kontextfreier Grammatiken

5.3.1 Parser

5.3.2 Der YACC-Parsergenerator

5.3.3 Markup-Sprachen

5.3.4 XML und Dokumenttypdefinitionen

5.3.5 Übungen zum Abschnitt 5.3

## 5.4 Mehrdeutigkeit von Grammatiken und Sprachen

5.4.1 Mehrdeutige Grammatiken

5.4.2 Mehrdeutigkeit aus Grammatiken tilgen

5.4.3 Linksseitige Ableitungen als Möglichkeit zur Beschreibung von Mehrdeutigkeit

5.4.4 Inhärente Mehrdeutigkeit

5.4.5 Übungen zum Abschnitt 5.4

## 6 Pushdown-Automaten

### 6.1 Definition des Pushdown-Automaten

6.1.1 Informelle Einführung

6.1.2 Die formale Definition von Pushdown-Automaten

6.1.3 Eine grafische Notation für PDAs

6.1.4 Unmittelbare Beschreibungen eines PDA

6.1.5 Übungen zum Abschnitt 6.1

### 6.2 Die Sprachen eines PDA

6.2.1 Akzeptanz durch Endzustand

6.2.2 Akzeptanz durch leeren Stack

6.2.3 Vom leeren Stack zum Endzustand

6.2.4 Vom Endzustand zum leeren Stack

6.2.5 Übungen zum Abschnitt 6.2

### 6.3 Äquivalenz von PDAs und kontextfreien Grammatiken

6.3.1 Von Grammatiken zu PDAs

# Inhaltsverzeichnis

6.3.2 Von PDAs zu Grammatiken

6.3.3 Übungen zum Abschnitt 6.3

## 6.4 Deterministische Pushdown-Automaten

6.4.1 Definition eines deterministischen PDA

6.4.2 Reguläre Sprachen und deterministische PDAs

6.4.3 DPDAs und kontextfreie Sprachen

6.4.4 DPDAs und mehrdeutige Grammatiken

6.4.5 Übungen zum Abschnitt 6.4

## 7 Eigenschaften kontextfreier Sprachen

### 7.1 Normalformen kontextfreier Grammatiken

7.1.1 Eliminierung unnützer Symbole

7.1.2 Berechnung der erzeugenden und erreichbaren Symbole

7.1.3 e-Produktionen eliminieren

7.1.4 Einheitsproduktionen eliminieren

7.1.5 Chomsky-Normalform

7.1.6 Übungen zum Abschnitt 7.1

### 7.2 Das Pumping-Lemma für kontextfreie Sprachen

7.2.1 Die Größe von Parse-Bäumen

7.2.2 Aussage des Pumping-Lemmas

7.2.3 Anwendungen des Pumping-Lemmas für kontextfreie Sprachen

7.2.4 Übungen zum Abschnitt 7.2

### 7.3 Abschluss-Eigenschaften kontextfreier Sprachen

7.3.1 Substitutionen

7.3.2 Anwendungen des Substitutions-Theorems

7.3.3 Spiegelung

7.3.4 Durchschnitt mit einer regulären Sprache

7.3.5 Inverse Homomorphismen

7.3.6 Übungen zum Abschnitt 7.3

### 7.4 Entscheidbarkeits-Eigenschaften kontextfreier Sprachen

7.4.1 Komplexität der Umwandlung von kfGs in PDAs und umgekehrt

7.4.2 Ausführungszeit der Umwandlung in Chomsky-Normalform

7.4.3 Prüfen, ob eine kontextfreie Sprache leer ist

7.4.4 Die Zugehörigkeit zu einer kontextfreien Sprache prüfen

7.4.5 Vorschau auf unentscheidbare kfL-Probleme

7.4.6 Übungen zum Abschnitt 7.4

## 8 Einführung in Turing-Maschinen

### 8.1 Probleme, die Computer nicht lösen können

8.1.1 Programme, die »Hello, World« ausgeben

8.1.2 Der hypothetische »Hello, World«-Tester

8.1.3 Ein Problem auf ein anderes Problem reduzieren

8.1.4 Übungen zum Abschnitt 8.1

# Inhaltsverzeichnis

## 8.2 Die Turing-Maschine

- 8.2.1 Das Streben danach, alle mathematischen Fragen zu entscheiden
- 8.2.2 Die Notation der Turing-Maschine
- 8.2.3 Unmittelbare Beschreibungen für Turing-Maschinen
- 8.2.4 Übergangdiagramme für Turing-Maschinen
- 8.2.5 Die Sprache einer Turing-Maschine
- 8.2.6 Turing-Maschinen und das Halteproblem
- 8.2.7 Übungen zum Abschnitt 8.2

## 8.3 Programmiertechniken für Turing-Maschinen

- 8.3.1 Speicher im Zustand
- 8.3.2 Mehrere Spuren
- 8.3.3 Unterprogramme
- 8.3.4 Übungen zum Abschnitt 8.3

## 8.4 Erweiterungen für die einfache Turing-Maschine

- 8.4.1 Turing-Maschinen mit mehreren Bändern
- 8.4.2 Äquivalenz zwischen ein- und mehrbändigen TMn
- 8.4.3 Ausführungszeit und die Viele-Bänder-in-eins-Konstruktion
- 8.4.4 Nichtdeterministische Turing-Maschinen
- 8.4.5 Übungen zum Abschnitt 8.4

## 8.5 Beschränkte Turing-Maschinen

- 8.5.1 Turing-Maschinen mit semi-unendlichen Bändern
- 8.5.2 Maschinen mit mehreren Stacks
- 8.5.3 Zählermaschinen
- 8.5.4 Die Leistungsfähigkeit von Zählermaschinen
- 8.5.5 Übungen zum Abschnitt 8.5

## 8.6 Turing-Maschinen und Computer

- 8.6.1 Eine Turing-Maschine mit einem Computer simulieren
- 8.6.2 Einen Computer mit einer Turing-Maschine simulieren
- 8.6.3 Laufzeitvergleich zwischen Computern und Turing-Maschinen

## 9 Unentscheidbarkeit

### 9.1 Eine nicht rekursiv aufzählbare Sprache

- 9.1.1 Binärzeichenreihen aufzählen
- 9.1.2 Codes für Turing-Maschinen
- 9.1.3 Die Diagonalisierungssprache
- 9.1.4 Der Beweis, dass  $L_d$  nicht rekursiv aufzählbar ist
- 9.1.5 Übungen zum Abschnitt 9.1

### 9.2 Ein unentscheidbares Problem, das rekursiv aufzählbar ist

- 9.2.1 Rekursive Sprachen
- 9.2.2 Komplemente rekursiver und rekursiv aufzählbarer Sprachen
- 9.2.3 Die universelle Sprache
- 9.2.4 Unentscheidbarkeit der universellen Sprache

# Inhaltsverzeichnis

9.2.5 Übungen zum Abschnitt 9.2

## 9.3 Unentscheidbare Probleme über Turing-Maschinen

9.3.1 Reduktionen

9.3.2 Turing-Maschinen, die die leere Sprache akzeptieren

9.3.3 Der Satz von Rice und Eigenschaften der rekursiv aufzählbaren Sprachen

9.3.4 Probleme bezüglich Spezifikationen von Turing-Maschinen

9.3.5 Übungen zum Abschnitt 9.3

## 9.4 Das Postsche Korrespondenz-Problem

9.4.1 Definition des Postschen Korrespondenz-Problems

9.4.2 Das »modifizierte« PKP

9.4.3 Fertigstellung des Beweises der PKP-Unentscheidbarkeit

9.4.4 Übungen zum Abschnitt 9.4

## 9.5 Andere unentscheidbare Probleme

9.5.1 Probleme bei Programmen

9.5.2 Unentscheidbarkeit der Mehrdeutigkeit kontextfreier Grammatiken

9.5.3 Das Komplement einer Listensprache

9.5.4 Übungen zum Abschnitt 9.5

## 10 Nicht handhabbare Probleme

### 10.1 Die Klassen P und NP

10.1.1 Mit polynomialem Zeitaufwand lösbare Probleme

10.1.2 Beispiel: Der Kruskal-Algorithmus

10.1.3 Nichtdeterministisch-polynomialer Zeitaufwand

10.1.4 Ein NP-Beispiel: Das Problem des Handlungsreisenden

10.1.5 Polynomzeit-Reduktionen

10.1.6 NP-vollständige Probleme

10.1.7 Übungen zum Abschnitt 10.1

### 10.2 Ein NP-vollständiges Problem

10.2.1 Das Erfüllbarkeitsproblem

10.2.2 SAT-Instanzen repräsentieren

10.2.3 NP-Vollständigkeit des SAT-Problems

10.2.4 Übungen zum Abschnitt 10.2

### 10.3 Ein eingeschränktes Erfüllbarkeitsproblem

10.3.1 Normalformen für Boolesche Ausdrücke

10.3.2 Ausdrücke in KNF konvertieren

10.3.3 NP-Vollständigkeit von CSAT

10.3.4 NP-Vollständigkeit von 3SAT

10.3.5 Übungen zum Abschnitt 10.3

### 10.4 Weitere NP-vollständige Probleme

10.4.1 NP-vollständige Probleme beschreiben

10.4.2 Das Problem unabhängiger Mengen

10.4.3 Das Problem der Knotenüberdeckung



# Inhaltsverzeichnis

- 10.4.4 Das Problem des gerichteten Hamiltonschen Kreises
- 10.4.5 Ungerichtete Hamiltonsche Kreise und das Problem des Handlungsreisenden
- 10.4.6 Zusammenfassung NP-vollständiger Probleme
- 10.4.7 Übungen zum Abschnitt 10.4

## 11 Zusätzliche Problemklassen

### 11.1 Komplemente von Sprachen, die in NP enthalten sind

- 11.1.1 Die Sprachklasse Co-NP
- 11.1.2 NP-vollständige Probleme und Co-NP
- 11.1.3 Übungen zum Abschnitt 11.1

### 11.2 Probleme, die mit polynomialem Speicherplatz lösbar sind

- 11.2.1 Turing-Maschinen mit polynomialer Platzbegrenzung
- 11.2.2 Beziehung von PS und NPS zu früher definierten Klassen
- 11.2.3 Deterministischer und nichtdeterministischer polynomialer Speicherplatz

### 11.3 Ein für PS vollständiges Problem

- 11.3.1 PS-Vollständigkeit
- 11.3.2 Quantifizierte Boolesche Formeln
- 11.3.3 Quantifizierte Boolesche Formeln auswerten
- 11.3.4 PS-Vollständigkeit des QBF-Problems
- 11.3.5 Übungen zum Abschnitt 11.3

### 11.4 Sprachklassen basierend auf Randomisierung

- 11.4.1 Quicksort: Ein Beispiel für einen zufallsabhängigen Algorithmus
- 11.4.2 Ein auf Zufallsabhängigkeit basierendes Modell einer Turing-Maschine
- 11.4.3 Die Sprache einer zufallsabhängigen Turing-Maschine
- 11.4.4 Die Klasse RP
- 11.4.5 In RP enthaltene Sprachen erkennen
- 11.4.6 Die Klasse ZPP
- 11.4.7 Beziehung zwischen RP und ZPP
- 11.4.8 Beziehungen zu den Klassen P und NP

### 11.5 Die Komplexität des Primzahltests

- 11.5.1 Die Bedeutung des Primzahltests
- 11.5.2 Einführung in Modular-Arithmetik
- 11.5.3 Die Komplexität modular-arithmetischer Berechnungen
- 11.5.4 Zufallsabhängig-polynomiales Primzahl-Testen
- 11.5.5 Nichtdeterministische Primzahltests
- 11.5.6 Übungen zum Abschnitt 11.5

Literaturverzeichnis

Stichwortverzeichnis

Ins Internet: Weitere Infos zum Buch, Downloads, etc.

Copyright

# Copyright

Daten, Texte, Design und Grafiken dieses eBooks, sowie die eventuell angebotenen eBook-Zusatzdaten sind urheberrechtlich geschützt. Dieses eBook stellen wir lediglich als **persönliche Einzelplatz-Lizenz** zur Verfügung!

Jede andere Verwendung dieses eBooks oder zugehöriger Materialien und Informationen, einschließlich

- der Reproduktion,
- der Weitergabe,
- des Weitervertriebs,
- der Platzierung im Internet, in Intranets, in Extranets,
- der Veränderung,
- des Weiterverkaufs und
- der Veröffentlichung

bedarf der **schriftlichen Genehmigung** des Verlags. Insbesondere ist die Entfernung oder Änderung des vom Verlag vergebenen Passwortschutzes ausdrücklich untersagt!

Bei Fragen zu diesem Thema wenden Sie sich bitte an: [info@pearson.de](mailto:info@pearson.de)

## Zusatzdaten

Möglicherweise liegt dem gedruckten Buch eine CD-ROM mit Zusatzdaten bei. Die Zurverfügungstellung dieser Daten auf unseren Websites ist eine freiwillige Leistung des Verlags. **Der Rechtsweg ist ausgeschlossen.**

## Hinweis

Dieses und viele weitere eBooks können Sie rund um die Uhr und legal auf unserer Website herunterladen:

**<http://ebooks.pearson.de>**