

Informatik

Formale Sprachen, Compilerbau, Berechenbarkeit und Komplexität

Nach einer allgemeinen Diskussion formaler Sprachen, deren Beschreibungen und Grenzfällen der Erkennbarkeit werden die regulären Sprachen behandelt, welche in der lexikalischen Definition von Programmiersprachen ihre wichtigste Anwendung finden sowie die kontextfreien Sprachen, mit denen man die Syntax von Programmiersprachen definiert. Aus theoretischer Sicht befriedigend ist die eindeutige Entsprechung zwischen Sprachbeschreibung und Spracherkennung - den regulären Sprachen entsprechen die endlichen Automaten und den kontextfreien Sprachen die Stackmaschinen. Weitere Stufen der Chomsky-Hierarchie werden nur kurz behandelt, da sie in der Praxis von geringerer Bedeutung sind. Stattdessen zeigt ein eigenes Kapitel zum Thema Compilerbau weitere Techniken auf, die aus einer Sprachbeschreibung einen Parser, also das komplette »front-end« eines Compilers, entstehen lassen. Der Begriff des »Algorithmus« wird anhand verschiedener Maschinenmodelle erklärt und bestätigt wird auch die Churchsche These, dass jede vernünftige Definition von »Berechenbarkeit« auf die gleiche Klasse von Funktionen führt. Die Grenzen des algorithmisch Machbaren werden anhand des Halteproblems und des Satzes von Rice klar abgesteckt. Das abschließende Kapitel zur Komplexitätstheorie erkundet unter den lösbaren Problemen die Grenze zwischen denen, die mit einem vertretbaren (polynomiellen) Aufwand lösbar sind und solchen, deren Lösung nicht wesentlich effizienter ist, als ein systematisches Ausprobieren von Lösungskandidaten. Dieses Kapitel führt den Leser zu dem bekanntesten noch ungelösten Problem der Theoretischen Informatik: $P = NP?$

Der erste Band der erklärt die grundlegenden Konzepte: Programmierung, Algorithmen und Datenstrukturen. Der **zweite Band** ist technischen Themen gewidmet – insbesondere der Rechnerarchitektur, Betriebssystemen, Rechnernetzen und speziell dem Internet.

Das Buch richtet sich an alle Einsteiger, die sich ernsthaft mit Informatik beschäftigen wollen, sei es zum Selbststudium oder zur Begleitung von Vorlesungen. In den folgenden Bänden dieses Buches werden die Themen, Rechnerarchitektur, Betriebssysteme, Rechnernetze, Internet, Compilerbau und Theoretische Informatik vertieft.

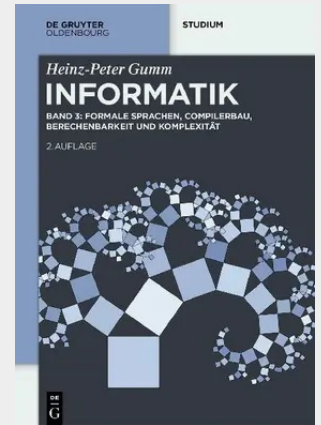
Prof. Dr. Heinz-Peter Gumm

ist Professor für Theoretische Informatik in Marburg. Nach dem Studium in Darmstadt und Winnipeg (Kanada) von 1970 bis 1975 und der Habilitation 1981 folgten Professuren in Hawaii, Kalifornien und New York. Seine Forschungsgebiete sind Formale Methoden, Allgemeine Algebren und Coalgebren.

Prof. Dr. Manfred Sommer

ist emeritierter Professor für Praktische Informatik in Marburg. Nach dem Studium in Göttingen und München von 1964 bis 1969, war er Assistent am ersten Informatik-Institut in Deutschland an der TU München. Es folgten zehn Jahre bei Siemens in München und von 1984 bis 2014 war er Informatik-Professor in Marburg.

Nach einer allgemeinen Diskussion formaler Sprachen, deren Beschreibungen und Grenzfällen der Erkennbarkeit werden die regulären Sprachen behandelt, welche in der lexikalischen Definition von Programmiersprachen ihre wichtigste Anwendung finden sowie die kontextfreien Sprachen, mit denen man die Syntax von Programmiersprachen definiert. Aus theoretischer Sicht befriedigend ist die eindeutige Entsprechung zwischen Sprachbeschreibung und Spracherkennung - den regulären Sprachen entsprechen die endlichen Automaten und den kontextfreien Sprachen die Stackmaschinen. Weitere Stufen der Chomsky-Hierarchie werden nur kurz behandelt, da sie in der Praxis von geringerer Bedeutung sind. Stattdessen zeigt ein eigenes Kapitel zum Thema



49,95 €

46,68 € (zzgl. MwSt.)

Lieferfrist: bis zu 10 Tage

Artikelnummer: 9783111634845

Medium: Buch

ISBN: 978-3-11-163484-5

Verlag: De Gruyter

Erscheinungstermin: 22.06.2026

Sprache(n): Deutsch

Auflage: 2. aktualisierte und erweiterte Auflage 2026

Serie: De Gruyter Studium

Produktform: Kartoniert

Gewicht: 512 g

Seiten: 287

Format (B x H): 168 x 239 mm

