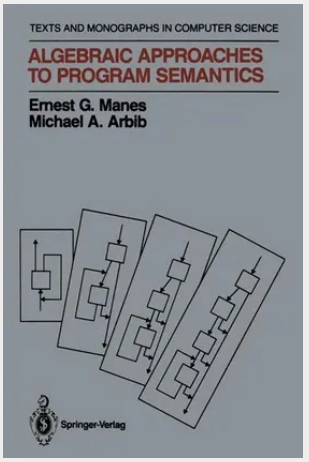
Manes / Arbib

Algebraic Approaches to Program Semantics

In the 1930s, mathematical logicians studied the notion of "effective comput ability" using such notions as recursive functions, A-calculus, and Turing machines. The 1940s saw the construction of the first electronic computers, and the next 20 years saw the evolution of higher-level programming languages in which programs could be written in a convenient fashion independent (thanks to compilers and interpreters) of the architecture of any specific machine. The development of such languages led in turn to the general analysis of questions of syntax, structuring strings of symbols which could count as legal programs, and semantics, determining the "meaning" of a program, for example, as the function it computes in transforming input data to output results. An important approach to semantics, pioneered by Floyd, Hoare, and Wirth, is called assertion semantics: given a specification of which assertions (preconditions) on input data should guarantee that the results satisfy desired assertions (postconditions) on output data, one seeks a logical proof that the program satisfies its specification. An alternative approach, pioneered by Scott and Strachey, is called denotational semantics: it offers algebraic techniques for characterizing the denotation of (i. e., the function computed by) a program-the properties of the program can then be checked by direct comparison of the denotation with the specification. This book is an introduction to denotational semantics. More specifically, we introduce the reader to two approaches to denotational semantics: the order semantics of Scott and Strachey and our own partially additive semantics.

In the 1930s, mathematical logicians studied the notion of "effective comput ability" using such notions as recursive functions, A-calculus, and Turing machines. The 1940s saw the construction of the first electronic computers, and the next 20 years saw the evolution of higher-level programming languages in which programs could be written in a convenient fashion independent (thanks to compilers and interpreters) of the architecture of any specific machine. The development of such languages led in turn to the general analysis of questions of syntax, structuring strings of symbols which could count as legal programs, and semantics, determining the "meaning" of a program, for example, as the function it computes in transforming input data to output results. An important approach to semantics, pioneered by Floyd, Hoare, and Wirth, is called assertion semantics: given a specification of which assertions (preconditions) on input data should guarantee that the results satisfy desired assertions (postconditions) on output data, one seeks a logical proof that the program satisfies its specification. An alternative approach, pioneered by Scott and Strachey, is called denotational semantics: it offers algebraic techniques for characterizing the denotation of (i. e. , the function computed by) a program-the properties of the program can then be checked by direct comparison of the denotation with the specification. This book is an introduction to denotational semantics. More specifically, we introduce the reader to two approaches to denotational semantics: the order semantics of Scott and Strachey and our own partially additive semantics.



93,08 €
86,99 € (zzgl. MwSt.)

Lieferfrist: bis zu 10 Tage

Artikelnummer: 9781461293774
Medium: Buch
ISBN: 978-1-4612-9377-4
Verlag: Springer
Erscheinungstermin: 17.01.2014
Sprache(n): Englisch
Auflage: Erscheinungsjahr 2014
Serie: Monographs in Computer Science
Produktform: Kartoniert
Gewicht: 563 g
Seiten: 353
Format (B x H): 155 x 235 mm

Kundenservice Verlag Dr. Otto Schmidt KG
Neumannstraße 10, 40235 Düsseldorf | kundenservice@otto-schmidt.de | 0221-93738-997 (Inland)