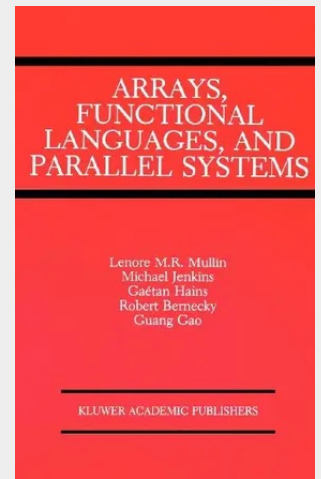


Arrays, Functional Languages, and Parallel Systems

During a meeting in Toronto last winter, Mike Jenkins, Bob Bernecky and I were discussing how the two existing theories on arrays influenced or were influenced by programming languages and systems. More's Army Theory was the basis for NIAL and APL2 and Mullin's A Mathematics of Arrays (MOA), is being used as an algebra of arrays in functional and A-calculus based programming languages. MOA was influenced by Iverson's initial and extended algebra, the foundations for APL and J respectively. We discussed that there is a lot of interest in the Computer Science and Engineering communities concerning formal methods for languages that could support massively parallel operations in scientific computing, a back-to-roots interest for both Mike and myself. Languages for this domain can no longer be informally developed since it is necessary to map languages easily to many multiprocessor architectures. Software systems intended for parallel computation require a formal basis so that modifications can be done with relative ease while ensuring integrity in design. List based languages are profiting from theoretical foundations such as the Bird-Meertens formalism. Their theory has been successfully used to describe list based parallel algorithms across many classes of architectures.

During a meeting in Toronto last winter, Mike Jenkins, Bob Bernecky and I were discussing how the two existing theories on arrays influenced or were influenced by programming languages and systems. More's Army Theory was the basis for NIAL and APL2 and Mullin's A Mathematics of Arrays (MOA), is being used as an algebra of arrays in functional and A-calculus based programming languages. MOA was influenced by Iverson's initial and extended algebra, the foundations for APL and J respectively. We discussed that there is a lot of interest in the Computer Science and Engineering communities concerning formal methods for languages that could support massively parallel operations in scientific computing, a back-to-roots interest for both Mike and myself. Languages for this domain can no longer be informally developed since it is necessary to map languages easily to many multiprocessor architectures. Software systems intended for parallel computation require a formal basis so that modifications can be done with relative ease while ensuring integrity in design. List based languages are profiting from theoretical foundations such as the Bird-Meertens formalism. Their theory has been successfully used to describe list based parallel algorithms across many classes of architectures.



160,49 €
149,99 € (zzgl. MwSt.)

Lieferfrist: bis zu 10 Tage

Artikelnummer: 9780792392132
Medium: Buch
ISBN: 978-0-7923-9213-2
Verlag: Humana
Erscheinungstermin: 31.10.1991
Sprache(n): Englisch
Auflage: 1991
Produktform: Gebunden
Gewicht: 1430 g
Seiten: 319
Format (B x H): 160 x 241 mm

