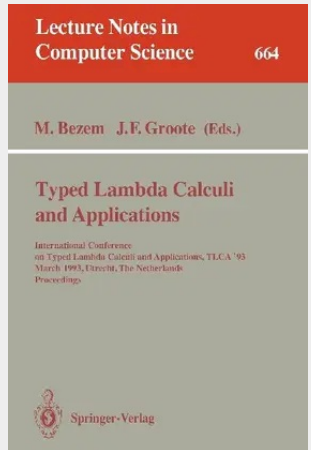


Typed Lambda Calculi and Applications

International Conference on Typed Lambda Calculi and Applications, TLCA '93, March 16-18, 1993, Utrecht, The Netherlands. Proceedings

The lambda calculus was developed in the 1930s by Alonzo Church. The calculus turned out to be an interesting model of computation and became the prototype for untyped functional programming languages. Operational and denotational semantics for the calculus served as examples for other programming languages. In typed lambda calculi, lambda terms are classified according to their applicative behavior. In the 1960s it was discovered that the types of typed lambda calculi are in fact appearances of logical propositions. Thus there are two possible views of typed lambda calculi: - as models of computation, where terms are viewed as programs in a typed programming language; - as logical theories, where the types are viewed as propositions and the terms as proofs. The practical spin-off from these studies are: - functional programming languages which are mathematically more succinct than imperative programs; - systems for automated proof checking based on lambda calculi. This volume is the proceedings of TLCA '93, the first international conference on Typed Lambda Calculi and Applications, organized by the Department of Philosophy of Utrecht University. It includes 29 papers selected from 51 submissions.

The lambda calculus was developed in the 1930s by Alonzo Church. The calculus turned out to be an interesting model of computation and became the prototype for untyped functional programming languages. Operational and denotational semantics for the calculus served as examples for other programming languages. In typed lambda calculi, lambda terms are classified according to their applicative behavior. In the 1960s it was discovered that the types of typed lambda calculi are in fact appearances of logical propositions. Thus there are two possible views of typed lambda calculi: - as models of computation, where terms are viewed as programs in a typed programming language; - as logical theories, where the types are viewed as propositions and the terms as proofs. The practical spin-off from these studies are: - functional programming languages which are mathematically more succinct than imperative programs; - systems for automated proof checking based on lambda calculi. This volume is the proceedings of TLCA '93, the first international conference on Typed Lambda Calculi and Applications, organized by the Department of Philosophy of Utrecht University. It includes 29 papers selected from 51 submissions.



53,49 €
49,99 € (zzgl. MwSt.)

Lieferfrist: bis zu 10 Tage

Artikelnummer: 9783540565178
Medium: Buch
ISBN: 978-3-540-56517-8
Verlag: Springer
Erscheinungstermin: 03.03.1993
Sprache(n): Englisch
Auflage: Erscheinungsjahr 1993
Serie: Lecture Notes in Computer Science
Produktform: Kartoniert
Gewicht: 1400 g
Seiten: 443
Format (B x H): 155 x 235 mm

