

Two-Level Functional Languages

The authors describe here a framework in which the type notation of functional languages is extended to include a notation for binding times (that is run-time and compile-time) that distinguishes between them. Consequently the ability to specify code and verify program correctness can be improved. Two developments are needed, the first of which introduces the binding time distinction into the lambda calculus, in a manner analogous with the introduction of types into the untyped lambda calculus. Methods are also presented for introducing combinators for run-time. The second concerns the interpretation of the resulting language, which is known as the mixed lambda-calculus and combinatory logic. The notion of 'parametrized semantics' is used to describe code generation and abstract interpretation. The code generation is for a simple abstract machine designed for the purpose; it is close to the categorical abstract machine. The abstract interpretation focuses on a strictness analysis that generalises Wadler's analysis for lists. It is also shown how the results of abstract interpretation may be used to improve the code generation. The work is both an exposition and synthesis of recent research and as such will be valuable to research workers and graduate students working in formal methods and functional languages.

**148,80 €**

148,80 € (zzgl. MwSt.)

*Lieferfrist: bis zu 10 Tage***Artikelnummer:** 9780521403849**Medium:** Buch**ISBN:** 978-0-521-40384-9**Verlag:** Cambridge University Press**Erscheinungstermin:** 11.11.2014**Sprache(n):** Englisch**Auflage:** Erscheinungsjahr 2014**Serie:** Cambridge Tracts in Theoretical Computer Science**Produktform:** Gebunden**Gewicht:** 720 g**Seiten:** 311**Format (B x H):** 175 x 250 mm