1. Record Nr. UNISA996465290203316

Titolo	Functional Programming Languages and Computer Architecture [[electronic resource]] : 5th ACM Conference. Cambridge, MA, USA, August 26-30, 1991 Proceedings / / edited by John Hughes
Pubbl/distr/stampa	Berlin, Heidelberg : , : Springer Berlin Heidelberg : , : Imprint : Springer, , 1991
ISBN	3-540-47599-0
Edizione	[1st ed. 1991.]
Descrizione fisica	1 online resource (VIII, 672 p.)
Collana	Lecture Notes in Computer Science, , 0302-9743 ; ; 523
Disciplina	005.13
Soggetti	Architecture, Computer Programming languages (Electronic computers) Microprocessors Computer programming Computer logic Computer System Implementation Programming Languages, Compilers, Interpreters Processor Architectures Programming Techniques Logics and Meanings of Programs
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Note generali	Bibliographic Level Mode of Issuance: Monograph
Nota di contenuto	Type classes and overloading resolution via order-sorted unification -- On the complexity of ML typability with overloading -- Coercive type isomorphism -- Compiler-controlled multithreading for lenient parallel languages -- Multi-thread code generation for dataflow architectures from non-strict programs -- GAML: A parallel implementation of lazy ML -- Functional programming with bananas, lenses, envelopes and barbed wire -- A strongly-typed self-applicable partial evaluator -- Automatic online partial evaluation -- Assignments for applicative languages -- Linearity and laziness -- Syntactic detection of single-threading using continuations -- A projection model of types -- What is an efficient implementation of the ?-calculus? -- Outline of a proof theory of parametricity -- Reasoning about simple and exhaustive

demand in higher-order lazy languages -- Strictness analysis in logical form -- A note on abstract interpretation of polymorphic functions -- Incremental polymorphism -- Dynamics in ML -- Implementing regular tree expressions -- Efficient type inference for higher-order binding-time analysis -- Finiteness analysis -- For a better support of static data flow -- An architectural technique for cache-level garbage collection -- M-structures: Extending a parallel, non-strict, functional language with state -- List comprehensions in agna, a parallel persistent object system -- Generating efficient code for lazy functional languages -- Making abstract machines less abstract -- Unboxed values as first class citizens in a non-strict functional language.

Sommario/riassunto

This book offers a comprehensive view of the best and the latest work in functional programming. It is the proceedings of a major international conference and contains 30 papers selected from 126 submitted. A number of themes emerge. One is a growing interest in types: powerful type systems or type checkers supporting overloading, coercion, dynamic types, and incremental inference; linear types to optimize storage, and polymorphic types to optimize semantic analysis. The hot topic of partial evaluation is well represented: techniques for higher-order binding-time analysis, assuring termination of partial evaluation, and improving the residual programs a partial evaluator generates. The thorny problem of manipulating state in functional languages is addressed: one paper even argues that parallel programs with side-effects can be "more declarative" than purely functional ones. Theoretical work covers a new model of types based on projections, parametricity, a connection between strictness analysis and logic, and a discussion of efficient implementations of the lambda-calculus. The connection with computer architecture and a variety of other topics are also addressed.