

Advanced Functional Programming

Third International School, AFP'98, Braga, Portugal, September 12-19, 1998, Revised Lectures

the theory behind many things commonly written by functional programmers. One of the main achievements of the project has been to make people aware of the fact that many algorithms can be described in a data-independent way. The PolyP system introduced in these notes is one of the translations to the Haskell world of this theoretical underpinning.

The third lecture, on Generic Program Transformation, can also be seen as an application of the theory introduced in lecture two. Many efficiency-improving program transformations can be performed in a mechanical way, and these would not have been possible without insight into the correctness of such transformations gained in the lecture on Generic Programming.

The fourth lecture, on Designing and Implementing Combinator Languages, introduces an easy way to write formalism for writing down the catamorphisms introduced in earlier chapters. It is shown how quite complicated catamorphisms, that at first sight seem rather forbidding by making extensive use of higher-order do- VI Preface mains, can actually be developed in a step-wise fashion, using an attribute grammar view; it is furthermore shown how to relate this way of programming with concepts from the object-oriented world thus making clear what the strengths and weaknesses of each world are. The

the lecture, titled Using MetaML: A Staged Programming Language, introduces the concept of partial evaluation. It serves as another instance of the quest for "the most generic of writing programs at the lowest cost". The staging techniques show how costs that were introduced by adding extra levels of abstraction, may be moved from run-time to compile-time.

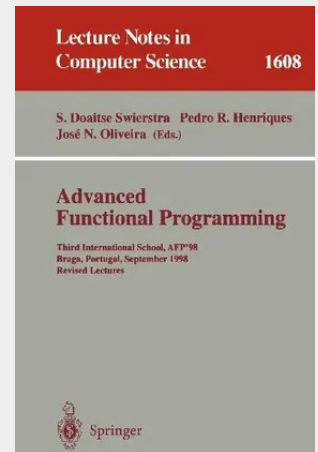
It has been common knowledge to users of modern functional languages that the type system can be a great help in shortening programs and reducing errors. In the extreme one might see a type as a predicate capturing the properties of any expression with that type. In the sixth lecture on Cayenne-Spice up your Programming with Dependent Types it is shown in what direction functional languages are most likely to develop, and what may be expected of the new type system to be introduced. The last lecture, titled Haskell as an Automation Controller, shows that writing functional programs does not have to imply that one is bound to remain isolated from the rest of the world. Being able to communicate with software written by others in a uniform way, is probably one of the most interesting new developments in current computer science. It appears that the concept of a monad together with the Haskell typing rules, is quite adequate to describe the interface between Haskell programs and the outer world.

Finally we want to thank everyone who contributed to this school and made it such a successful event: sponsors, local system managers, local organizers, students, and last but not least the lecturers. We are convinced that everyone present at the school enjoyed this event as much as we did, and we all hope that you will feel some of the spirit of this event when studying these lecture notes. March 1999

Doaitse Swierstra Pedro Henriques Jos'e Oliveira VII Sponsorship

The school has received generous sponsorship from: FCT-Fundac~o para a Ci~encia e Tecnologia, Minist'erio da Ci~encia e Tecnologia Adegacooperativa de Ponte de Lima Ag~encia Abreu CGD-Caixa Geral de Dep'ositos CIUM-Centro de Inform'atica da Universidade do Minho DI-Departamento de Inform'atica da Universidade do Minho GEPL-Grupo de Especi-

cac~ao e Processamento de Linguagens LESI-Direc~ao de Cursos de Engenharia de Sistemas de Inform'atica Enabler Lactolima Latic~o nios das Marinhas, Lda Novabase Porto-Sistemas de Informac~ao SA Primavera Software Projecto Camila-Grupo de M'etodos Formais Sidereus-



53,49 €
49,99 € (zzgl. MwSt.)

Lieferfrist: bis zu 10 Tage

Artikelnummer: 9783540662419
Medium: Buch
ISBN: 978-3-540-66241-9
Verlag: Springer
Erscheinungstermin: 07.07.1999
Sprache(n): Englisch
Auflage: 1. Auflage 1999
Serie: Lecture Notes in Computer Science
Produktform: Kartoniert
Gewicht: 476 g
Seiten: 298
Format (B x H): 155 x 235 mm

