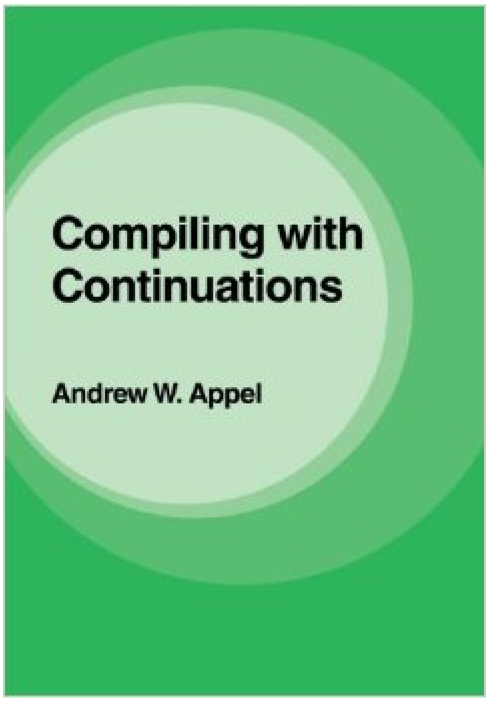
Compiling With Continuations



Compiling with Continuations

Andrew W. Appel

Synopsis

This book shows how continuation-passing style is used as an intermediate representation to perform optimizations and program transformations. Continuations can be used to compile most programming languages. The method is illustrated in a compiler for the programming language Standard ML. Prior knowledge of ML, however, is not necessary, as the author carefully explains each concept as it arises. This is the first book to show how concepts from the theory of programming languages can be applied to the production of practical optimizing compilers for modern languages like ML. All the details of compiling are covered, including the interface to a runtime system and garbage collector.

Book Information

Paperback: 272 pages

Publisher: Cambridge University Press; Revised ed. edition (February 12, 2007)

Language: English

ISBN-10: 052103311X

ISBN-13: 978-0521033114

Product Dimensions: 6.8 x 0.6 x 9.7 inches

Shipping Weight: 1.2 pounds (View shipping rates and policies)

Average Customer Review: 4.8 out of 5 starsÂ Â See all reviewsÂ (6 customer reviews)

Best Sellers Rank: #979,357 in Books (See Top 100 in Books)   #63 inÂ Books > Computers & Technology > Programming > Languages & Tools > Compiler Design   #168 inÂ Books > Computers & Technology > Programming > Languages & Tools > Compilers   #1228 inÂ Books > Textbooks > Computer Science > Software Design & Engineering

Customer Reviews

This is a very nice little book, and I found it to be surprisingly readable. The book is nicely written. Standard ML is used to illustrate the technique of compiling a functional language using continuations as the primary intermediate representation. Lack of familiarity with ML is not particularly burdensome. I would like to have seen more discussion of other languages, though (Scheme?).

I found this book while trying to build a interpreter for a distributed language. Appel’s approach not only solved my immediate issues (a uniform means of procedure call in the presence of mobility) but opened my mind to the utility of continuations in many areas of CS. It was a real mind opener, and the explanations were clear enough that I could adopt this approach with little difficulty.

This book was fantastic. It opened my mind to a different mindset towards the compilation process while at the same time demystifying many concepts that I had previously only partially understood.It is clear, concise, well-written, and unusually approachable for its genre. That said, I recommend some familiarity with ML-family languages as a prerequisite for approaching the book. He does include an appendix that's supposed to explain ML, but there are plenty of subtleties that I would have easily missed if that were my only resource.This book doesn't discuss the front end of the compiler at all--there is no discussion of lexical analysis, parsing or type inference. If that's what you're after, look elsewhere. This is text is limited to the back end of the compiler.The biggest quibble that I have with it is that the code generation chapter used MIPS/MAX/SPARC/68020 for case studies. In today's climate, ARM/x86/LLVM would be far more relevant and practical. This is an unfortunate consequence of the age of the text.

Compiling with Continuations