

## Haskell 프로그래밍을 위한 책 소개

변석우

순수 함수형 언어 (pure functional programming language) Haskell은 최근 빠른 발전을 거듭하고 있으며, 다목적 언어 (general-purpose language)로 성장하기 위한 여러 라이브러리와 프로그래밍 환경을 갖추어 가고 있다. 현재 Haskell 프로그램에서 기존의 C와 Java로 개발한 프로그램을 호출할 수 있으며, Visual Basic과 같이 Microsoft COM 환경에서 컴포넌트 프로그래밍을 할 수 있는 IDL이 개발되고 있다. 또한 웹브라우저에서 스크립트 프로그래밍을 할 수 있으며, CGI, DB, 그래픽스 등의 프로그래밍을 용이하게 할 수 있는 라이브러리들이 제공되고 있다. 현재 이러한 연구가 꾸준히 진행되고 있으므로 새로운 여러 흥미로운 프로그래밍 환경을 예상할 수 있다 (참조: <http://haskell.org/libraries/>).

Haskell의 발전 동향을 볼 때, Ericsson사의 Erlang과 같이, Haskell을 '산업적'으로 이용할 수 있을 것이라는 전망이 높아지고 있다. Haskell 컴파일러 개발에 주도적 역할을 하고 있는 Peyton-Jones 교수가 작년에 Microsoft사로 이직하면서 이러한 전망을 더욱 높여 주고 있다. 만약 이러한 전망이 실현된다면 이것은 컴퓨터 사이언스의 '이론'과 '실용' 사이의 벽을 허무는 중요한 의미를 갖는 것으로서, 그 파급효과는 매우 클 것으로 예상할 수 있다. 실용적이면서도 고려하지 않던 과도기적 수준에서 Haskell은 매우 흥미로운 특성을 가지고 있으며 교육적 가치가 크다고 보여진다.

본 고에서는 Haskell 프로그래밍에 관심을 갖는 독자들을 위해서 세 권의 책을 소개하려고 한다. 우선, Bird의 책은 예전부터 non-strict 함수형 프로그래밍의 대학 교재로서 널리 사용되었다. 1988년에 초판을 기반으로 새로운 내용과 Haskell의 새로운 기능을 보강시켜 제 2판이 출판되었다. 이 책은 초판의 경우와 마찬가지로 Oxford 대학 및 유럽 대학에서 주로 사용된 책으로서 함수형 프로그래밍의 전통적이면서 학문적인 분위기를 많이 느끼게 한다. 두 번째로, Thompson의 책은 1996년에 초판을 발간하고 1999년에 2판을 발간한 비교적 최근에 저술된 책이다. 이 책은 Hugs 인터프리터를 기반으로 사용자에게 좀 더 친근하고 프로그래밍에 유용한 여러 기능들을 쉽게 설명하고 있다. 최근 이 책은 미국 대학에서 1-2학년 용 대학 교재로서도 많이 사용되고 있다. 세 번째로, Hudak의 책은 지난 달에 출간된 책으로서 필자도 아직 직접 보지는 못하였다. 그러나, 지난 수년간의 Hudak의 연구와 웹에 소개된 내용으로 미루어볼 때, 이 책에서는 기존에 잘 다루어지지 않았던 interaction 등의 새로운 함수형 프로그래밍 기법을 소개하고 있는 것으로 예상된다. Hudak은 Conal Elliot와 함께 그래픽스 애니메이션 라이브러리인 Fran (Functional Reactive Animation)을 개발하였으며 음악을 처리하는 Haskore 등을 개발한 경험이 있는데, 이러한 내용이 소개되고 있는 것으로 보인다.

**Simon Thompson: Haskell: The Craft of Functional Programming**, Second Edition, Addison-Wesley, 507 pages, paperback, 1999. ISBN 0-201-34275-8. (국내 대리점: 그린출판사)

This book introduces Haskell at a level appropriate for those with little or no prior experience of functional programming. The emphasis is on the process of crafting programs, solving problems, and avoiding common errors.

The second edition of *Haskell: The Craft of Functional Programming* is essential reading for beginners to functional programming and newcomers to the Haskell programming language. The emphasis is on the process of crafting programs and the text contains many examples and running case studies, as well as advice on program design, testing, problem solving and how to avoid common pitfalls.

Building on the strengths of the first edition, the book includes many new and improved features:

Complete coverage of Haskell 98, the standard version of Haskell which will be stable and supported by implementations for years to come. An emphasis on software engineering principles, encouraging a disciplined approach to building reusable libraries of software components. Detailed coverage of the Hugs interpreter with an appendix covering other implementations. A running case study of pictures emphasizes the built-in functions which appear in the standard prelude and libraries. It is also used to give an early preview of some of the more complex language features, such as high-order functions. List comprehensions and the standard functions over lists are covered before recursion. Early coverage of polymorphism supporting the "toolkit" approach and encouraging the reuse of built-in functions and types.

**Richard Bird: Introduction to Functional Programming using Haskell**, 2nd edition, Prentice Hall Press, 1998, 460 pp., ISBN: 0-13-484346-0. (국내 대리점: 영한 출판사).

The book is self-contained, assuming no prior knowledge of programming, and is suitable as an introductory undergraduate text for first- or second-year students.

After the success of the first edition, *Introduction to Functional Programming using Haskell* has been thoroughly updated and revised to provide a complete grounding in the principles and techniques of

programming with functions. The second edition uses the popular language Haskell to express functional programs. There are new chapters on program optimisation, abstract datatypes in a functional setting, and programming in a monadic style. There are completely new case studies, and many new exercises. As in the first edition there is an emphasis on the fundamental techniques for reasoning about functional programs and for deriving them systematically from their specifications.

**Paul Hudak: The Haskell School of Expression,**

300pp, ISBN: 0 521 64408 9, September 1999, Cambridge University Press, (Price £ 17.95).

Functional programming is a style of programming that emphasizes the use of functions. It has become popular in recent years because of its simplicity, conciseness, and clarity. This book teaches functional programming as a way of thinking and problem solving, using Haskell, the most popular purely functional language. Rather than using the conventional (boring) mathematical examples commonly found in other programming language textbooks, the author uses examples drawn from multimedia applications, including graphics, animation, and computer music, thus rewarding the reader with working programs for inherently more interesting applications. Aimed at both beginning and advanced programmers, this tutorial begins with a gentle introduction to functional programming and moves rapidly on to more advanced topics. Details about programming in Haskell are presented in boxes throughout the text so they can be easily found and referred to.

Chapter Contents

1. Problem solving, programming, and calculation; 2. A module of shapes: part I; 3. Simple graphics; 4. Shapes II: drawing shapes; 5. Polymorphic and higher-order functions; 6. Shapes III: perimeters of shapes; 7. Trees; 8. A module of regions; 9. More about higher-order functions; 10. Drawing regions; 11. Proof by induction; 12. Qualified types; 13. A module of simple animations; 14. Programming with streams; 15. A module of reactive animations; 16. Communicating with the outside world; 17. Rendering reactive animations; 18. Higher-order types; 19. An imperative robot language; 20. Functional music composition; 21. Algebraic properties of multimedia; 22. Interpreting functional music; 23. A tour of the prelude list module; 24. A Tour of Haskell' s standard type classes.