Lisp In Small Pieces

DOWNLOAD EBOOK
This is a comprehensive account of the semantics and the implementation of the whole Lisp family of languages, namely Lisp, Scheme and related dialects. It describes 11 interpreters and 2 compilers, including very recent techniques of interpretation and compilation. The book is in two parts. The first starts from a simple evaluation function and enriches it with multiple name spaces, continuations and side-effects with commented variants, while at the same time the language used to define these features is reduced to a simple lambda-calculus. Denotational semantics is then naturally introduced. The second part focuses more on implementation techniques and discusses precompilation for fast interpretation: threaded code or bytecode; compilation towards C. Some extensions are also described such as dynamic evaluation, reflection, macros and objects. This will become the new standard reference for people wanting to know more about the Lisp family of languages: how they work, how they are implemented, what their variants are and why such variants exist. The full code is supplied (and also available over the Net). A large bibliography is given as well as a considerable number of exercises. Thus it may also be used by students to accompany second courses on Lisp or Scheme.
and even oldtimers as Cobol and Fortran tend to develop, or rather mature, into languages getting closer and closer to Lisp, Algol, and their ultimate offspring, Scheme. This is not without reason. But although the many qualities of Lisp have long since been known in academia, they need time and, more important, good reference material, to find their way into the real world. Lisp programmers know the value of everything, but the cost of nothing, it is said. Christian Queinnec neatly fills the gap in our knowledge in a book that is a hard read because of the density of the content, but also a fun book because all the source is there (available through the Internet, of course) to experiment with. You will not only gain insight into the workings of your Lisp system. You will gain insight into the basic elements of computer programming languages and their reason for being, their implementation, and the benefits and costs they will bring you. All in all, one of the best books on Lisp I have ever almost, but not completely grokked. I sincerely believe that tomorrow’s programming languages, whether they be called C** or Delphi 2010, will be closer to current Lisp than to current C or Pascal, and a way to efficiently implement these languages is available here and now. The book covers all standard material like direct interpretation, compilation towards a virtual machine using bytecodes, and compilation to C.

Download to continue reading...


Dmca