8th International Conference on Applied Informatics
Eger, Hungary, January 27–30, 2010.

Measuring Software Complexity by Types

Gábor PÁLI[1], Tamás KOZSIK[2]

Faculty of Informatics, Department of Programming Languages and Compilers,
Eötvös Loránd University
[1] pgj@elte.hu [2] Tamas.Kozsik@elte.hu

Abstract

In typed programming languages, types play an important role equally in safety, optimization, documentation, and the employed abstractions. According to this merits, it is very likely that types can be also used as a tool of measuring software complexity. This paper discusses a potential approach to software measurement techniques for software written in strongly typed, especially functional programming languages.

Although there are initiatives for establishing metrics for Haskell programs [1][2], measuring complexity of programs written in functional languages is still a topic to be explored. As number and size of software systems implemented in such languages are continuously growing nowadays, existence of valuable software metrics is getting important.

In our opinion, it is worth to focus on types themselves when trying to say something about the quality of a software product. This work examines the possibilities of describing program quality and complexity through a static analysis of types by adapting the necessary foundamentals from type theory and deriving metrics from them.

Keywords: Functional Programming, Software Measurement, Software Quality, Metrics, Programming with Types, Haskell

MSC: 68N30 (Mathematical aspects of software engineering)

References

[1] Ryder, C. *Software Measurement for Functional Programming*, PhD thesis, Computing Lab, University of Kent, Canterbury, UK (August 2004)

[2] Ryder, C., Thompson, S. *Software Metrics: Measuring Haskell*, In Marko van Eekelen and Kevin Hammond, editors, Trends in Functional Programming (September 2005)