

Refactoring Erlang Programs¹

Zoltán Horváth, László Lövei, Tamás Kozsik, Anikó Víg, and Tamás Nagy

We present here the prototype of a refactoring toolset for Erlang programs where one can incrementally carry out programmer-guided meaning-preserving program transformations. We discuss an approach to the problems of storing and extracting the syntactic and also the static semantic information in order to be flexible enough to perform the desired transformations. In our approach the program to be redesigned is stored in a relational database. The Erlang ODBC interface is used to access the database. The refactoring tool is integrated into a development environment using Emacs as an interface. The refactoring steps are implemented in Erlang, using the standard Erlang parser to construct the database from the source code. This backend is connected to Emacs through Distel. The tool has two different modes. In the first mode, the programmer can choose from the safe refactor steps, which results an other safe position. Editing the source code is prohibited in this mode, which eliminates the need for reparsing and rebuilding the database. In the second mode, editing is possible in two ways: the programmer can choose from menu points like "insert a function", which is partly controlled, or can edit the source code freely. Further refactor steps need syntactically correct source code.

References

- [1] H. Li, C. Reinke, and S. Thompson. Tool Support for Refactoring Functional Programs, *Haskell Workshop: Proceedings of the ACM SIGPLAN workshop on Haskell*, Uppsala, Sweden, Pages: 27–38, 2003.
- [2] Á. Fóthi, Z. Horváth, and J. Nyéky-Gaizler. A Relational Model of Transformation in Programming, *Proceedings of the 3rd International Conference on Applied Informatics*, Eger-Noszvaj, Hungary, Aug. 26-28, 1997. 335-349.
- [3] M. Fowler, K. Beck, J. Brant, W. Opdyke, and D. Roberts. Refactoring: Improving the Design of Existing Code, *Addison-Wesley*, 1999.
- [4] *Martin Fowler's refactoring site*, <http://www.refactoring.com>
- [5] R. Szabó-Nacsa, P. Diviánszky, and Z. Horváth. Prototype Environment for Refactoring Clean programs, *CSCS 2004, The Fourth Conference of PhD Students in Computer Science Szeged*, Hungary, July 1-4, 2004.
- [6] Distel: Distributed Emacs Lisp, <http://fresh.homeunix.net/~luke/distel>
- [7] J. Armstrong, R. Virding, M. Williams, C. Wikstrom. Concurrent Programming in Erlang, *Prentice Hall*, 1996

¹Supported by GVOP-3.3.3-2004-07-0005/3.0 ELTE IKKK, Ericsson and ELTE CNL