Call graph and data flow analysis of a dynamic functional language *

Tamás Nagy, Zoltán Horváth, László Lövei, Anikó Nagyné Víg

Department of Programming Languages and Compilers
Eötvös Loránd University, Budapest, Hungary
e-mail: {hz,lovei,n_tamas,viganiko}@inf.elte.hu

Abstract

Refactoring is about improving the design of existing program code: making changes to the source code which preserves the meaning of the program in order to improve non-functional characteristics of the code like readability or maintainability.

In refactoring, the highest amount of work is usually the precondition checking, which makes sure that the refactoring does not change the behaviour of the system. Compared to precondition checking an uncomplicated transformation is almost straightforward. To check whether the preconditions are met, the type system of the language can provide very useful information. Our primary research areas are those functional languages which do not have a static type system. Our aim is to find static analysis techniques which can provide enough information to check whether preconditions are met. Two examples of such techniques are call graph and data flow analysis.

*Supported by ELTE IKKK, Ericsson Hungary