

# Recovering Erlang conditionals from BEAM bytecode\*

Dániel Lukács, Melinda Tóth

Eötvös Lóránd Tudományegyetem, Budapest, Hungary  
{dlukacs,tothmelinda}@caesar.elte.hu

## Abstract

Erlang is a general-purpose functional programming language with wide range of applications mostly in the field of distributed systems. Erlang code is first compiled to BEAM bytecode, which consists of instructions targeting the Erlang Virtual Machine. Most Erlang libraries are shipped as BEAM files, and so Erlang sources commonly depend on information residing in bytecode format. Our goal is to extend the RefactorErlstatic analysis framework for Erlang, with a BEAM analysis capabilities, and thus to enable the users of the framework to also analyse and gain useful semantical information about source dependencies that are only available in BEAM representation.

Analysing BEAM code to recover information about the original sources presents several difficulties. While the Erlang source and the corresponding BEAM bytecode is semantically equivalent, the compilation process is, in general, not reversible: the Erlang compiler performs code optimization, and discards unnecessary syntactical information. Furthermore, Erlang is a high-level functional language, while BEAM is a low-level instruction-based imperative language, which results in non-trivial differences in the structure of the two representation. In this work, we introduce a procedure with the aim of resolving these difficulties, utilizing various static analysis techniques and higher level intermediate representations. Our main focus in this paper is the recovery of functional conditional expressions from BEAM conditional control structures. We will discuss how conditional patterns may be identified, present a technique for structuring such patterns into valid Erlang expressions, and propose a way to minimize the logical expressions recovered by analyzing bytecode conditional instructions.

*Keywords:* Erlang, BEAM, decompilation, structuring conditionals, bytecode  
*MSC:* 68N18, 68N20

---

\*SUPPORTED BY THE ÚNKP-16-2 NEW NATIONAL EXCELLENCE PROGRAM OF THE MINISTRY OF HUMAN CAPACITIES