

Clang Matchers for Verified Usage of the C++ Standard Template Library

Gábor Horváth, Norbert Pataki

Dept. of Programming Languages and Compilers,
Fac. of Informatics, Eötvös Loránd University, Budapest
`xazax.hun@gmail.com, patakino@elte.hu`

Abstract

The *C++ Standard Template Library (STL)* is the exemplar of generic libraries. Professional C++ programs cannot miss the usage of this standard library because it increases quality, maintainability, understandability and efficacy of the code. However, the usage of C++ STL does not guarantee bugfree or error-free code. Contrarily, incorrect application of the library may introduce new types of problems. Unfortunately, there is still a large number of properties are tested neither at compilation-time nor at run-time. It is not suprising that in implementation of C++ programs so many STL-related bugs occurred.

In this paper we argue for a new approach that is based on static analysis. We match patterns on *abstract syntax trees* (AST) with the help of predicates. The predicates can be combined and define an embedded language. We have developed a tool which finds the potential misusages of the STL as a validation of our approach. The software takes advantage of the Clang ASTMatcher technology. The tool is in-use in the Ericsson. We advised new matchers that have get into the Clang codebase.

Keywords: C++ STL, Clang, AST, static analysis, code validation

MSC: 68N19 Other programming techniques