Symbol Clustering: Resolving ambiguous symbol references of large-scale C / C++ projects based on linkage information

Richárd Szalay[a], **Zoltán Porkoláb**[a], **Dániel Krupp**[b]

[a]Eötvös Loránd University, Faculty of Informatics,
Dept. of Programming Languages and Compilers
szalayrichard@inf.elte.hu, gsd@elte.hu

[b]Ericsson Ltd.
daniel.krupp@ericsson.com

Abstract

Resolving symbol references is an important part of many application areas from development environments to various static analyser tools. Different occurrences of the same program elements, like function definitions and their call sites, variable declarations and their usage, or type definitions and their applications should be connected. In case of the C++ programming language, the most current tools use mangled names to correlate symbols, e.g. when implementing actions like "go to definition" or "list all references". However, for large projects, where multiple binaries are created, symbol resolution based on mangled names can be, and usually is, ambiguous. This leads to inaccurate behaviour even in major development tools. In this paper we explore the reason of this ambiguity, and propose a method to improve the accuracy of symbol resolution for large C and C++ projects. We introduce our clustering algorithm based on essential build information of the project and discuss various implementation approaches to minimise run-time and storage overhead of the method. We implemented our method as part of the CodeCompass open source code comprehension tool and measured its efficiency on various large C++ projects. Although our method still leaves some ambiguity in case of the dynamically used symbols (e.g. via `dlopen`), in most of the practical cases it eliminates the uncertainty connected to symbol resolution and can significantly increase the quality of development environments.

Keywords: program comprehension, C++ programming language, symbol resolution

MSC: 68N15 Programming languages