CLABUREDB: Classified Bug-Reports Database
Tool for Developers of Program Analysis Tools

Jiri Slaby, Jan Strejček, and Marek Trtík

Faculty of Informatics, Masaryk University
Botanická 68a, 60200 Brno, Czech Republic
{slaby,strejcek,trtik}@fi.muni.cz

Abstract. We present a database that can serve as a tool for tuning and evaluation of miscellaneous program analysis tools. The database contains bug-reports produced by various tools applied to various source codes. The bug-reports are classified as either real errors or false positives. The database currently contains more than 800 bug-reports detected in the Linux kernel 2.6.28. Support of other software projects written in various programming languages is planned. The database can be downloaded and manipulated by SQL queries, or accessed via a web frontend.

1 Introduction

Many successful bug-finding tools based on various program analysis methods appeared during the last ten years. None of them is perfect. Each tool either reports both real errors and false positives, or it discovers only a part of real errors. To improve or evaluate such a tool, one needs to run the tool on some source codes and then analyze the obtained bug-reports, i.e. classify them as false positives or real errors, and find errors in the sources that were missed by the tool. This work is usually tedious and time consuming, especially when one tunes or studies performance of a tool for real software projects. The tedious work can be avoided if suitable benchmarks, i.e. programs with information about their errors, are available.

There exist benchmark suites consisting of small synthetic programs [1,2,4,6,7] and those consisting of real-world programs [2,3,5]. In benchmark suites [1,4,6], relevant program locations in small synthetic programs are explicitly marked as either erroneous or safe. The benchmark suite [2] marks only known real errors. The situation is different for benchmarks with real-world programs: [5] contains marked test cases triggering/non-triggering errors, while [3] provides bug-reports (both real errors and false positives mixed together) produced by FINDBUGS and sorted according to priority levels assigned to the reports by the tool. The benchmark suites discussed so far consider different error types and

---

1 We deliberately use term “bug” in the paper. It stems from the need of the database to comprehend for example coding style violations. In those cases, commonly used terms like “failure” or “fault” fail to apply.
provide their own error taxonomies with exception of suites \cite{12}, where errors types are linked to Common Weakness Enumeration (CWE) \cite{10}.

As far as we know, there is no benchmark suite containing big real-life projects with a significant list of uniformly described bug-reports classified as real errors or false positives. Our benchmark suite ClabureDB should fill this gap.

ClabureDB currently contains only a single project, namely the Linux kernel 2.6.28. We have collected about 850 bug-reports of 11 error types produced by several bug-detection tools run on the kernel. The reports have been manually classified as either real errors or false positives by skilled programmers with a help of Linux kernel developers. In fact, it would be sufficient to store only real errors assuming that we know all of them (and thus we can assume that all other bug-reports are false positives). As this assumption is completely unrealistic for a real-life project, we store both real errors and false positives.

The database is still developing in several directions: we plan to add more bug-reports for the Linux kernel, to support other software projects, and to augment the web interface to allow other users to add and maintain the database content.

The database can be downloaded in the SQLite 3 format for local use under the Open Database License v1.0 \cite{11} or accessed via a web interface at:

\url{http://claburedb.fi.muni.cz/}

The paper is organized as follows. The following section introduces the basic structure of the database and our web interface. Section 3 describes the current content of the database including considered kinds of errors and an overview of collected bug-reports and their sources. Section 4 suggests possible use of the database for evaluation of a program analysis tool. Finally, the last section presents our future plans with ClabureDB.

2 Database Structure

The database is designed to accommodate various kinds of errors from diverse projects and project versions. As projects can be written in arbitrary programming languages, can contain very specific kinds of errors, can be maintained by different teams, and can be interesting for distinct user groups, we decided to have each project in a separate sub-database. A sub-database comprises information about considered error types, bug-reports, users, tools, and relations between them. There are two main tables in each sub-database:

**error_type.** This table keeps a specification of considered kinds of errors. We will outline some of possible types later in Section 3.2. The table contains a name of the type, short description and a reference CWE number if exists (see later).

**error.** Each line in this table corresponds to one bug-report. It is specified by the error type, location (usually file and line), URL for reference (to provide more information), classification (false positive, real error, unclassified), confirmation (an argument supporting the validity of the bug-report classification, e.g. a commit ID of the corresponding bug-fix), user who inserted the entry, and timestamp of the moment of insertion.