*Robert Reitmeier*

Confidence in R2

Version: 1.0

Date: 2022-02-02

State: **in progress** / presented / released

Classification: Confidential

File: confidence\_in\_r2.docx

# History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Change |
| 0.1 | 2022-01-28 | R. Reitmeier | Draft |
|  |  |  |  |

# Scope

This document gives a rationale why one can have high confidence using the dynamic analysis tool R2 for closing coverage gaps in C source code.

# Description of the Core Functionality of R2

R2 is a dynamic analysis tool for C source code, implemented in the functional language Haskell. Based on the abstract syntax tree of C code, it performs a depth-first search over all control flow paths through a C function, including called subfunctions. All decision conditions (like “if”, “while”, “for”, conditional expressions etc.) on the way through the code are collected and a corresponding SMTLIB2 model is constructed. This model is passed on to a SMT solver (currently Z3) in order to find solutions, which are then collected by R2. Finally, a report is created giving all stimuli for maximum code coverage, as well as all decisions that are considered dead code.

In addition, there are the following verification facilities provided by R2:

* Each stimulus that is found is verified by compiling and executing the source code, comparing the function’s predicted output value to the execution output value.
* By inserting a so called “solver\_find” pragma at specific location in the source code, it can be verified if the specific location was actually reached during execution of the code.
* By inserting a so called “solver\_debug” pragma at specific location in the source code, the theoretical prediction of an expression’s value at this location can be compared to the “real” execution runtime value. However, currently this has to be done manually.

# Summary