MUNI
FACULTY
OF INFORMATICS

DISERTATION THESIS
READER'S REVIEW

Title:	*Static Analysis of C Programs*
Author:	*Ing. Viktor Malík*
Supervisor:	*prof. Ing. Tomáš Vojnar, Ph.D.*
Reader:	*prof. RNDr. Jan Strejček, Ph.D.*

I have carefully read the dissertation thesis of Viktor Malík. Before I comment on its content and evaluate its qualities, I have to declare that I know Viktor at least since 2018 when we met as researchers interested in software analysis and verification. However, we have never directly cooperated on any research project and I feel no bias or conflict of interests.

The thesis presents new results in the area of program analysis. More precisely, it focuses on the following two problems.

Verification of heap- and array-manipulating programs

The thesis describes extensions of the verification framework 2LS, which is briefly introduced in Chapter 2. Chapter 3 presents a new memory model and an encoding of basic memory-manipulating operations into quantifier-free first-order formulas. Further, it introduces an abstract template domain for shape analysis (i.e., an abstraction of possible memory contents) and its use for verification of memory safety. Chapter 4 then introduces an anstract template domain for reasoning about arrays. All the suggested extensions have been implemented to 2LS and dramatically improved its performance on relevant programs, as described in Chapter 5.

The suggested extensions are published in the proceedings of FMCAD 2018 (Core B). The implementation in 2LS is also presentented in three publications related to the participation of the tool in the *Competition on Software Verification (SV-COMP)* 2018, 2020, and 2023. Finally, the contribution is also covered by a chapter about 2LS that should soon appear in a book about current software verification tools.

Determining semantic equivalence of various versions of large C projects

The second part of the thesis presents a novel approach to determining semantic equivalence of the corresponding interface functions of two versions of C projects. The overall approach is introduced in Chapter 6, while Chapters 7 and 8 focus on built-in and custom change patterns, respectively. Chapter 9 then deals with the use of global variables. Chapter 10 presents the implementation in a tool DiffKemp and experimental results. Chapter 11 discusses related work.

The approach has been designed with the aim to scale to large C project and experimental evaluation clearly shows that this goal has been achieved and the tool indeed produces relevant results. The approach has been presented in two publications in conference proceedings of ICST 2021 (Core A) and NETYS 2022. Moreover, the core algorithm is a subject of an U.S. patent.

I consider the problems addresses in the thesis to be very relevant not only for the community of academic researchers, but also for industry. The achieved results are original and they significantly contributed to the state of the art. More precisely, the presented extensions of 2LS improved its applicability and performance, which helped the tool to became a successful and highly respected participant of the SV-COMP. The presented approach to semantic equivalence checking delivers a fresh, complex, and efficient solution of the problem. Moreover, the DiffKemp tool showed to be a valuable industrial-quality tool with a clear potential to save a lot of human effort in software development.

The text of the thesis is well structured, carefully typeset (with the exception of using T and $\mathcal{T}$ with different meaning in the same formula), readable, and illustrated by suitable running examples. I found only a few typos. Unfortunately, some parts of the text are hard to follow. The description of the kIkI algorithm suffers from the fact that some notation used in Figure 2.2 illustrating the algorithm is defined only 5 pages later. Some parts of the text would benefit from more precise formulation. These issues induced the some of the following questions.

- Can the approach presented in Chapter 3 handle arrays of the length dependent on the input?
- Figure 5.1 presents result of 2LS in heap-related categories in years 2017–2023. Were the benchmarks in these categories the same over all these years?
- How should the output of DiffKemp be exactly interpreted? The text says that a reported non-equivalence does not guarantee that the corresponding functions are indeed non-equivalent. In my opinion, the reported equivalence does not imply equivalence of the sources (for example, because some freedom in the semantics of C sources can be lost by their translation to LLVM IR). Should the reported equivalence be interpreted as the equivalence of binaries obtained by Clang? Or is it only a heuristics that do not provide any guarantees?
- The slicing algorithm presented in Section 9.2 reconstructs control flow of the slice in phase 3 such that the successor of each instruction in the slice is identified. Does it also identifies some predecessors of each instruction in the slice? How is it done?
- Why Table 10.4 does not contain the comparison of versions 8.0/8.1?

Regardless of these questions, the thesis proves the author's solid knowledge of the studied field, his scientific erudition, and ability to conduct original research. The list of Viktor's publications contains papers on relevant, respected, and competitive international conferences, in particular FMCAD 2018 and ICST 2021. I appreciate the fact that all research results have been also properly implemented and intesively evaluated on relevant data.

To sum up, I recommend to accept the dissertation. In my opinion, the submitted thesis and the author's achievement meet the generally accepted requirements for the award of a doctoral degree (in accordance with Section 47 of Act No. 111/1998 Coll. on higher education institution).

Brno, November 23, 2023 Jan Strejček