VUB SCIENCES & BIOENGINEERING SCIENCES

The Research Group
Software Languages Lab

has the honor to invite you to the public defence of the PhD thesis of

Jens Van der Plas

to obtain the degree of Doctor of Sciences

Title of the PhD thesis:

Incremental Static Program Analysis through Reified Computational Dependencies

Promotors:
Prof. Dr. Coen De Roover (VUB)
Prof. Dr. Quentin Stiévenart (Université du Québec à Montréal, CA)

The defence will take place on
Monday, November 4, 2024 at 5 p.m. in Promotiezaal D.2.01

The defence can also be followed through a live stream. Please send an email to jevdplas@vub.be for more information.

Members of the jury

Prof. Dr. Dominique Maes (VUB, chair)
Prof. Dr. Jens Nicolay (VUB, secretary)
Prof. Dr. Wolfgang De Meuter (VUB)
Prof. Dr. Lynn Houthuys (VUB)
Prof. Dr. Görel Hedin (Lund University, SE)
Prof. Dr. Ben Hermann (Technische Universität Dortmund, DE)

Curriculum vitae

Jens Van der Plas started his studies at VUB in 2014, where he obtained his master’s degree in 2019. Afterwards, he started a PhD at the Software Languages Lab under the supervision of Prof. Dr. Coen De Roover and he was awarded an FWO grant. The research of Jens is situated in the domain of incremental static program analysis. He co-authored eight peer-reviewed publications at international conferences and in an international journal and was a first author three times. Jens supervised five bachelor theses and equally many master theses. He represents his colleagues in the faculty council and in the faculty board of his faculty.

Abstract of the PhD research

Over the last few decades, computers have become an indispensable part of modern society. As the programs running on these computers play an essential role in everyday life, for example in banking and communication, it is crucial that they are reliable. To this end, developers have come to rely on static analysis tools to verify a wide range of program properties without actually running the program. Static analyses are typically integrated into modern software development environments and continuous integration systems to safeguard software quality throughout the entire development process.

During the development process, software developers continuously apply small changes to the program. It is therefore not only important for a static analysis to deliver precise feedback, but also to efficiently update its feedback whenever the program is modified. To this end, an incremental static analysis avoids a full recomputation of the analysis result. Instead, upon a change in the program, it reuses and updates the previously computed result, saving valuable analysis time.

It is challenging to develop an incremental analysis that is sufficiently powerful to verify a wide range of program properties and that supports complex language features found in contemporary programming languages. In this work, we present a novel generic approach to construct such analyses by using reified computational dependencies. We show how an analysis that reifies the computational dependencies within the analysed program can be rendered incremental. By relying on these dependencies, our incremental analysis restricts the impact of a change to the affected parts of the analysis result. Moreover, we impose minimal requirements on the analysis itself, allowing a broad applicability of our approach to incrementalisation.

To preserve the precision of the resulting incremental analysis, we introduce three complementary result-invalidation strategies that limit the loss in precision. These strategies are built around the core idea of interleaving invalidation with computation.