JOHANNES KEPLER UNIVERSITÄT LINZ

Bachelor's Thesis

Kotlin Compiler IR Visualizer

Dipl.-Ing. Dr. Markus Weninger, BSc
Institute for System Software
T +43-732-2468-4361
markus.weninger@jku.at

Student: David Weinstabl
Advisor: Dipl.-Ing. Dr. Markus Weninger, BSc
Start date: March 2025

In the realm of software development, compilers play a crucial role in transforming human-readable code into machine-executable instructions. Kotlin, a statically typed programming language targeting the JVM, Android, and JavaScript, is renowned for its conciseness and expressiveness. The Kotlin compiler utilizes an Intermediate Representation (IR), which is an abstract syntax tree that represents the program structure during the compilation process. Understanding the Kotlin IR is essential for developers, especially when creating compiler plugins that modify or extend the compiler's functionality.

Developing Kotlin compiler plugins is a complex task, primarily due to the initial learning curve associated with understanding the Kotlin IR. New developers often struggle with the internal data structures of the Kotlin compiler, such as the different node types within the IR, the information they contain, and the methods to manipulate them. This complexity can hinder the development of plugins that, for example, record method entry and exit times to identify performance bottlenecks.

The primary objective of this bachelor's thesis is to develop an IR Visualizer for the Kotlin compiler. This visualization tool will enable plugin developers to inspect the internal data structure of the compiler during the compilation process. By providing a graphical representation of the compiled classes, methods, properties, and their relationships within the IR tree, the tool will facilitate a deeper understanding of the Kotlin IR.

Sub-goals and deliverables include:

- A web-based visualization tool that graphically represents the Kotlin IR.
- Interactive features allowing developers to inspect different IR nodes and their connections.
- Documentation on how to use the tool effectively.

The beneficiaries of this tool include Kotlin compiler plugin developers, educators teaching compiler construction, and students learning about Kotlin's internal mechanisms. While the immediate goal of this bachelor's thesis is to create a visualization tool for the Kotlin compiler's Intermediate Representation (IR), there is significant potential for extending this tool into a full-fledged graphical programming environment in future work. Such an environment would not only serve as a visual aid but could also become an interactive platform for code development and manipulation directly through visual IR.

The student undertaking this thesis should have:

- A basic understanding of compiler theory and Kotlin programming.
- Proficiency in web development technologies for creating the visualization tool.
- An interest in developing educational tools and user interfaces.

Skills to be developed during the thesis include advanced knowledge of Kotlin's IR, experience in designing interactive visualization tools, and the ability to synthesize complex technical information into user-friendly formats.

<u>Modalities:</u>

The progress of the project should be discussed at least every four weeks with the advisor(s). A time schedule and a milestone plan must be set up within the first 2 weeks and discussed with the advisor(s). It should be continuously refined and monitored to make sure that the thesis will be completed in time. The final version of the thesis must be submitted not later than 31.08.2025.