

SIMULATING TAIL CALLS: LUA SEMANTICS WITHIN THE TRUFFLE FRAMEWORK

Univ.-Prof. Dr. Stefan Marr
Institute for System Software
P: +43-732-2468-4340
M: stefan.marr@jku.at

Student: Bernhard Lehner, BSc
Advisor: Univ.-Prof. Dr. Stefan Marr
Co-Advisor: Dipl.-Ing. Christoph Aigner
Begin: 1.10.2026

Secretary:
ARⁱⁿ Karin Gusenbauer
P: +43-732-2468-4342
M: karin.gusenbauer@jku.at

Lua¹ is a simple lightweight programming language, that is used as an embedded scripting language in a whole host of applications ranging from Apache HTTP Servers all the way to Roblox. The Truffle language implementation framework² is a library, that aims to make writing a fast language implementation as easy as possible. It provides a domain specific language for implementing interpreters, which are then used to compile the guest language just in time, to achieve good peak performance. The Truffle framework is part of GraalVM³, a high performance polyglot Java virtual machine (JVM) developed by Oracle.

This master thesis will build upon a currently semantically incomplete implementation of the Lua language on the Truffle framework called Truffle-Lua. One prominent missing feature is support for tail calls. The Lua specification mandates that any chain of calls in tail position must not build up the stack. Implementing this in Truffle is non-trivial, as the framework is constrained to Java semantics, which do not support tail calls natively. Therefore, the central goal of this thesis is to simulate the semantics of Lua tail calls within the confines of Truffle, while keeping the performance impact to a minimum.

Since tail calls are a common feature in functional languages, a main goal of this work is to investigate integration of a common tail call solution into the Truffle framework itself. An additional goal of this thesis is, to identify and fix further semantic incompleteness of Truffle-Lua along the way.

Goals of this thesis are:

- Enable infinite tail call recursion without stack overflow in Truffle-Lua.
- Investigate, compare, and contrast different tail call recursion approaches in environments without true tail calls.
- Build an efficient implementation of tail calls in Truffle-Lua.
- Investigate generalization of the chosen approach to the Truffle framework.
- Furthering semantic completeness for Truffle-Lua's implementation.

¹<https://www.lua.org/>

²<https://www.graalvm.org/latest/graalvm-as-a-platform/language-implementation-framework/>

³<https://www.graalvm.org/>

Modalities

The project progress is usually discussed every two weeks. As part of the Master's Seminar, three presentations are given to discuss the project progress with the institute. As opportunities arise, progress is discussed with interested external parties, also to gain feedback on relevant experiments to validate performance on real-world use cases. The final version of the thesis must be submitted within one year after start.