

Assignment of bachelor's thesis

Title: Haskell Dynamic Tracing
Student: Ondřej Kvapil
Supervisor: Ing. Filip Křikava, Ph.D.
Study program: Informatics
Branch / specialization: Computer Science
Department: Department of Theoretical Computer Science
Validity: until the end of summer semester 2020/2021

Instructions

Lazy evaluation is a strategy that delays expression evaluation until its value is needed. This allows one to avoid unnecessary computation and use of infinite data structures. Recently, Goel and Vitek looked into the use of laziness in R [1], which is one of the most widely used lazy programming languages. They found little evidence supporting that programmers use laziness to save on computation or use infinite data structures. It would be interesting to compare this to the use of laziness in Haskell. For this, we need a way to trace the execution of real-world Haskell programs. The goal of this thesis is, therefore, to design and implement a dynamic tracing framework for Haskell. It shall be scalable in order to allow us to analyze a large corpus of Haskell code available on GitHub. The dynamic tracer should capture all interesting events such function call and argument order evaluation and present them in an easy to be queried form.

–

[1] DOI: 10.1145/3360579