C | A | U Kiel University
Christian-Albrechts-Universität zu Kiel
Faculty of Engineering

Haskell Code Generation

Can we make it faster?

KiCS2 — Kiel Curry System

A compiler that generates Haskell code

At our group, we are actively developing the Curry compiler KiCS2. The generated Haskell code is — of course — not comparable to code that is directly written in Haskell. Moreover, since Curry adds logic features like non-determinism, such language constructs have to be translated to more complicated code in Haskell. Because of the non-determinism, standard data types — like lists — cannot be mapped one-to-one. Up to now, most people trust the Haskell compiler GHC to perform optimisations on the generated code. However, it was never investigated how well the optimisations work and whether we can trigger more optimisations.

Can we trigger more optimisations?

The goal of this project is to evaluate which kind of Curry programs can benefit from GHC optimisations as it is. Your task will be to implement a testing environment, read up on possible options to fine-tune GHC's compiling process with respect to optimisations, evaluate best practices for the current state of the code generation, and document your results for further research.
If you are comfortable with KiCS2's code generation before the end of the project, we would love to hear your input on possible improvements in order to take advantage of GHC's optimisations.

Haskell
We expect you to have knowledge in Haskell and its toolchain, e.g., GHC and cabal.

Curry
With background in Haskell, we expect you to quickly get comfortable with Curry, a functional logic language that features built-in non-determinism.

ABOUT KIEL AND OUR GROUP

Kiel is the capital of Germany's northernmost federal state with a direct border to the Baltic Sea. Every summer, the "Kiel Week" gathers tourists and sailors from all over the world to celebrate the annual sailing festival. Ten percent of Kiel's inhabitants are students at the university.

Our group consists of five PhD students (in their first to fifth year) and our advisor Prof. Michael Hanus. We like declarative languages such as Prolog, Haskell, and Coq, and use Curry as a research language.

We look forward to a successful project and are glad to respond to any questions beforehand, just write us an e-mail!

– Finn & Sandra

M.Sc. Sandra Dylus & M.Sc. Finn Teegen

Programming Languages and Compiler Construction Group