C | A | U Kiel University
Christian-Albrechts-Universität zu Kiel
Department of Computer Science

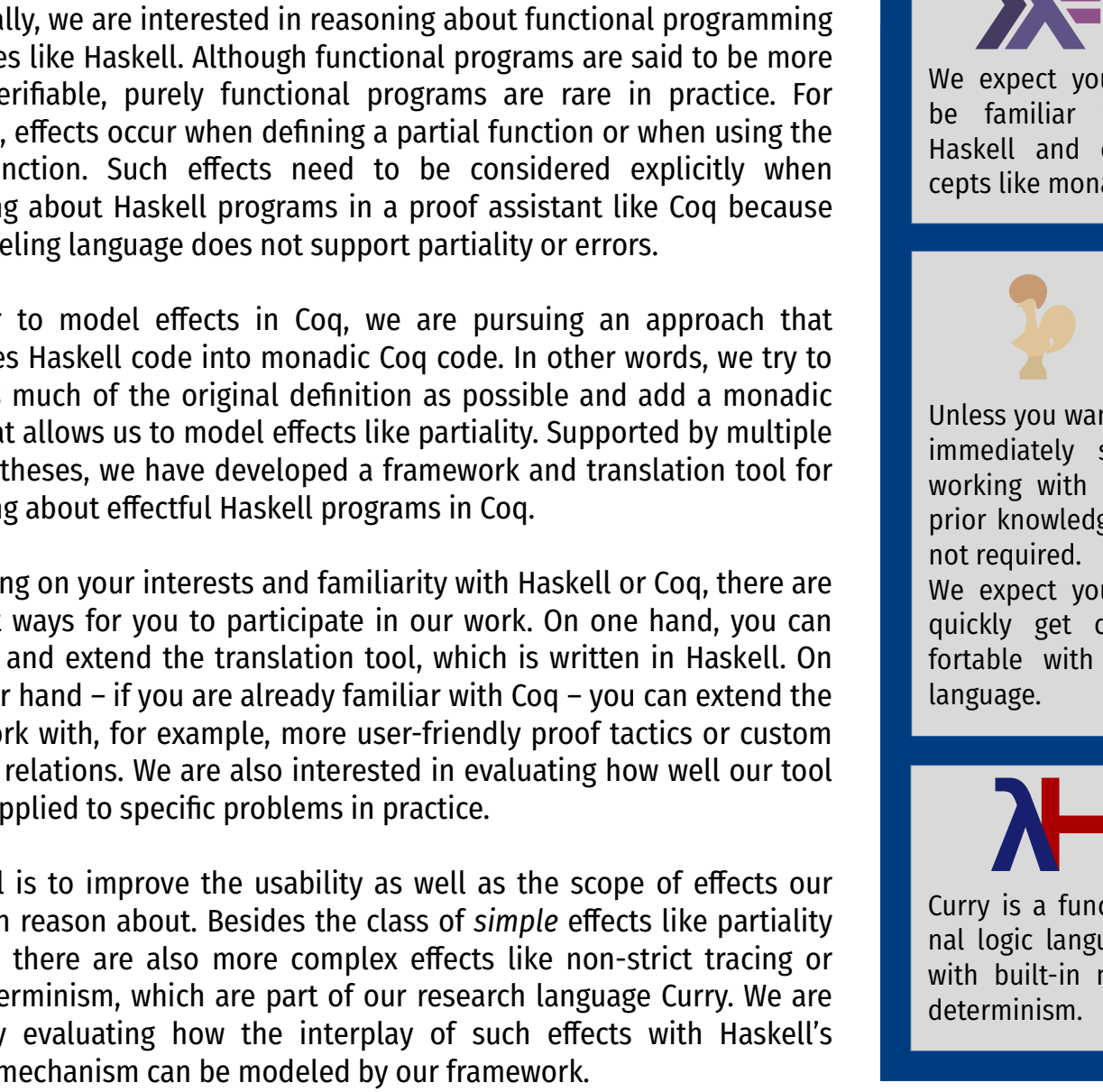
Verifying Effectful Haskell Programs in Coq

Program verification is one area of research our group engages in. Specifically, we are interested in reasoning about functional programming languages like Haskell. Although functional programs are said to be more easily verifiable, purely functional programs are rare in practice. For example, effects occur when defining a partial function or when using the error function. Such effects need to be considered explicitly when reasoning about Haskell programs in a proof assistant like Coq because the modeling language does not support partiality or errors.

In order to model effects in Coq, we are pursuing an approach that translates Haskell code into monadic Coq code. In other words, we try to reuse as much of the original definition as possible and add a monadic layer that allows us to model effects like partiality. Supported by multiple student theses, we have developed a framework and translation tool for reasoning about effectful Haskell programs in Coq.

Depending on your interests and familiarity with Haskell or Coq, there are different ways for you to participate in our work. On one hand, you can improve and extend the translation tool, which is written in Haskell. On the other hand – if you are already familiar with Coq – you can extend the framework with, for example, more user-friendly proof tactics or custom equality relations. We are also interested in evaluating how well our tool can be applied to specific problems in practice.

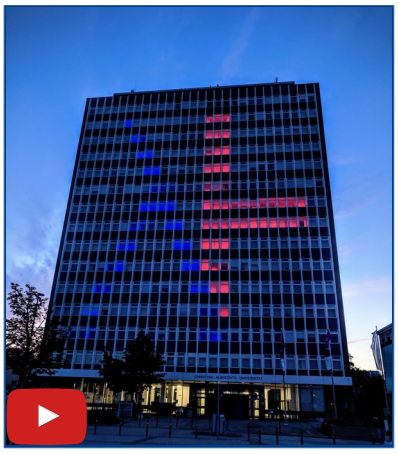
Our goal is to improve the usability as well as the scope of effects our tools can reason about. Besides the class of *simple* effects like partiality or error, there are also more complex effects like non-strict tracing or non-determinism, which are part of our research language Curry. We are currently evaluating how the interplay of such effects with Haskell's sharing mechanism can be modeled by our framework.



We expect you to be familiar with Haskell and concepts like monads.

Unless you want to immediately start working with Coq, prior knowledge is not required.
We expect you to quickly get comfortable with the language.

Curry is a functional logic language with built-in nondeterminism.



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 three PhD students (from their first to fifth year) and our advisor Prof. Michael Hanus. We like declarative languages such as 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 email!

M.Sc. Sandra Dylus & M.Sc. Niels Bunkenburg
Programming Languages and Compiler Construction Group