Advanced Functional Programming

Software Engineering Chair and Programming Systems Lab

Small-group work

Questions for *Fun with Phantom Types* by Ralf Hinze. It appeared as a chapter in the Book *The Fun of Programming*, edited by Jeremy Gibbons and Oege de Moor, 2003 Palgrave MacMillan, pages 245–262.

1. What is a *phantom type*? Can you imagine other uses that are not in the paper?

2. What the paper really talks about are guarded algebraic datatypes (GADTs) and their application. What extensions relative to 'normal' Haskell do GADTs consist of?

3. Look at the `compress` funciton in Section 2. Try to implement the inverse `uncompress :: Type t -> [Bit] -> t`. You may use `fail` for error cases.

4. Typing of GADTs is more intricate than it might seem at first. For example, consider

   ```
   f (Dyn _ x) = x
   ```

   (with the definition of type `Dynamic` from Section 3). How does this declaration behave, and why?

5. The paper claims that type inference is not possible for GADTs. Can you think of a simple counterexample demonstrating this?

6. The last section gives a translation of GADTs into plain Haskell plus existential types. Apply it to the `Term` type and the `eval` function from the first section. (Note: You do not need existential types for this example.)

7. Are GADTs really a suitable substitute for type classes, as the author claims? Discuss commonalities and differences.

Homework

1. Read *The Influence of Browsers on Evaluators* by Christian Queinnec.

2. Summarize the paper *in your own words* on one page. Put your name and student ID on your summary and drop off a printout at office **326/45** until Monday, January 23th at noon (12 am). If the door is closed, slide your printout under the door. No Emails.