Advanced Functional Programming

Software Engineering Chair and Programming Systems Lab

Assignment

1. Read *Fun with binary heap trees* by Chris Okasaki, in *The Fun of Programming*, Jeremy Gibbos and Oege de Moor (Eds.), Palgrave – Macmillan, pages 1–16, 2003.

2. Discuss the paper with other students; here are some good questions you could try to answer:

   - What problem is Okasaki trying to solve or demonstrate?
   - What is the contribution of the paper?
   - Where could you imagine using this technique?
   - What do you like about the paper, what not?
   - Could you imagine implementing this in ML or C? Why or why not?

3. Summarize the paper *in your own words* on one page. Most of your summary should identify the problem being solved, the solution, advantages, and limitations. After you have done that you can offer also your opinion about the paper, best backed up by some arguments. You can write your summary either in German or English.

4. Put your name and student ID on your summary and drop off a printout at office 326/45 until Monday noon, 12am. If the door is closed, slide your printout under the door. Emails won't be accepted.

5. It is totally acceptable to not understand everything in a paper and to have questions. Write them down; we will try to answer them in the next session.

Background Material

Many papers about functional programming present code in Haskell. It is therefore a good idea to acquire a general understanding of Haskell.

1. Read *A Gentle Introduction to Haskell*, available from `www.haskell.org/tutorial`.

2. Install Hugs, a Haskell interpreter available from `www.haskell.org/hugs`. Alternatively you could install the GHC Haskell compiler and use GHCi. It is instructive to try the code from Okasaki's paper.