Advanced Functional Programming

Software Engineering Chair and Programming Systems Lab

Small-group work

Divide into groups and discuss the following questions. Please try to answer the questions in the given order but go on to the next question if you can't agree on an answer within the allotted time.

1. Hutton uses the term *combinator parsing*—what is a combinator? Give some examples for combinators not from this paper. (5 minutes)
2. Hutton stresses the use of lazy languages; is laziness for parser combinators essential? (5 minutes)
3. What is the role of `$using`? Could you do without? (10 minutes)
4. Is combinator parsing efficient? Consider `$alt` and how often a character in an input stream is processed. (10 minutes)
5. What is the contribution of this paper? We will try to answer this question in a plenary discussion. (30 minutes)

How to read a research paper

After we have read some papers let's reflect for a moment and ask some questions about the process of reading papers:

1. How do you read a paper? That is, what do you look for, what is important for you?
2. Is there a universal yardstick by which all papers can be measured?
3. What papers to read?

Homework Assignment

1. Read *The essence of functional programming* by Philip Wadler, Proceedings of the 19th ACM SIGPLAN Symposium on Principles of Programming Languages, pages 1–14, 1992. Monads are tough to understand even for advanced functional programmers. So don't expect to understand everything. On the other hand, monads have been found a very useful abstraction—your effort will pay off.
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, November 21st at noon (12am). If the door is closed, slide your printout under the door. No Emails.