# Advanced Functional Programming

Software Engineering Chair and Programming Systems Lab

## Group Work

Questions for *Origami Programming* by Jeremy Gibbons; it appeared as a chapter in *The Fun of Programming*, Jeremy Gibbons and Oege De Moor (Editors), pages 41–60, Palgrave Macmillan, 2003.

1. The type of Haskell's `foldr` is $(\alpha \to \beta \to \beta) \to \beta \to [\alpha] \to \beta$. There is a connection between `foldr`'s arguments and the list constructors `(:)` and `[]`—which one?

2. Consider the following Haskell declaration for type `Tree`:

   ```
   data Tree = Leaf Int | Fwd Int Tree | Branch Tree Int Tree
   ```

   (a) The type of the list constructor (:) is $\alpha \to [\alpha] \to [\alpha]$. What are analogously the types of the `Tree` constructors?

   (b) Give the type of a function `fold` that traverses a `Tree` value.

   (c) Give an implementation of `fold`.

3. While folds over lists are frequently found in functional programs, they are rare for other types. Speculate why.

4. Implement the following functions over lists using `foldr`:

   - list concatenation `++` : $[\alpha] \to [\alpha] \to [\alpha]$
   - mapping `map` : $(\alpha \to \beta) \to [\alpha] \to [\beta]$
   - list flattening `concat` : $[[\alpha]] \to [\alpha]$
   - Solve exercise 3.19.

## Timeline for Presentations

Below is the timeline for your presentation. The dates given are the latest possible so you are free to present your results earlier.

- In the week starting March 6th: talk with your advisor about your paper and related papers.
- In the week starting March 13th: present a draft of your slides.
- March 22nd and 23rd: present your topic in a 25-minute talk.
- Your written presentation is due until Friday, April 7th, 12am.