

Best Lecture: Purely Functional Queues

Colin Runciman
University of York, UK

May 2017

Abstract

In an advanced undergraduate course on functional programming, after a rapid review of basic ingredients including polymorphic types, recursion equations, higher-order functions, laziness and modularity, what next?

One answer is to study some purely functional data structures — say for queues, sets and graphs. This lecture is the first of a trio, and its story centres around three alternative implementations of purely functional queues.

Beyond the interesting problem of representing queues, the lecture includes various themes of wider relevance for functional programmers, such as:

- Inverting conventional thinking: persistence is easy but efficient update is hard.
- Choosing abstraction boundaries: type-generic specification with type-specific implementations.
- Celebrating lists: a last hurrah then plug and play.
- Expressing properties beyond types: invariants as design choices and the limits of verification.
- Enjoying laziness: simplicity of expression and reasoning about efficiency.

The lecture leads on to a variety of programming exercises involving queues, or some combination of queues with sets or graphs.