

Discussion questions for *Why Functional Programming Matters*

COMP 150 - Applied Functional Programming

September 19, 2012

Why Functional Programming Matters can be viewed in two ways:

- As a *tutorial*, showing basic technique
- As an *advocacy* paper, arguing that real-world programmers should pay attention to functional programming because it helps solve real-world problems

We will explore both aspects.

Analysis

Expect to spend a lot more time on some questions than on others.

- (1) Most papers answer a question or advance a claim. Just to be sure we're all on the same page, what is Hughes's question or claim?
- (2) Can you break down Hughes's paper into details and relate it to your own experience?
 - Are there ideas whose value you can confirm from your own knowledge as a functional programmer?
 - Are there ideas whose value you can't confirm directly, but are consistent with your past experience as a functional programmer?
 - Are there ideas that sound good to you, but that you can't buttress with prior experience?
 - Are there ideas you are skeptical about? Based on experience, or not?

Laziness

- (3) Some functional programmers think the world would be better if function application were strict and only datatype constructors were lazy. What do you think?
- (4) Are Hughes's functions `anytrue` and `anyfalse` (page 4) as efficient as they can be? If not, how could they be improved?

Modularity

- (5) What is modularity? Is there more than one legitimate definition? If so, in what sense does Hughes use the term?
- (6) Hughes says that lazy evaluation is the "most powerful" modularization tool in the programmer's repertoire (page 7). What evidence or argument does he provide to substantiate that claim? Are you convinced?

Retrospective

- (7) According to Hughes, everything is just great. Is he right? If there is a weakness somewhere, what is it?