Monads and Staging

COMP 617 Proposal for Fall 2008

Jun Inoue

August 27, 2008

1 Research Plan

I will first wrap up the verification of the "memoize → CPS → stage" method of staging a program.

I will work on understanding arrows, monads, and monad transformers, formally, intuitively, and practically, with the goal of producing an accessible but technical introduction to their theoretical grounding and their use in producing a monadically/arrowly structured interpreter with staging. The intended audience is programming language experts and students who may not be familiar with monads or arrows in particular, but are knowledgeable in other aspects of programming language design and formalization. I will also produce an implementation if possible.[1] The (quality of the) interpreter will be assessed in terms of modularity and performance.

If time permits, I will (in parallel) tinker with coreboot and the XO's[2] hardware initializer to find any opportunities for staging or staging-related compilation techniques.

2 Papers To Present On

3 or 4 out of the following, listed in random order:

[1] I expect two obstacles for this. One is time constraints. The other is the fact that OCaml lacks type classes, which can be a show-stopper for monads as monads are useful to the extent that we have a generic interface for them.

[2] XO is the hardware produced by the One Latop Per Child (OLPC) project.

- "Monad transformers and modular interpreters." Liang, Hudak, and Jones (1995).
- "Computational lambda-calculus and monads." Moggi (1989). OR "Notions of computation and monads." Moggi.
- What I end up writing for verification of FFT staging. (Will present also in COMP 600.)
- What I might end up writing for the VPP $\rightarrow$ Uccello verification work that Gregory Malecha left and Cherif may or may not be working on at the moment.
- "Generalising monads to arrows." Hughes. Possibly augmented with "A new notation for arrows." Paterson.