

Effectively Effecting Effects

Examples of Effects

I/O { ^{cache} ^{input} ^{output} }
 { ^{ops} }
 heap { read, write (with regions), alloc }
 exceptions
 synchronization

non-determinism { ^{events} }
 randomness
 ! and ?
 information flow

What is an effect?

classification of
 How a program ~~computes~~ ^{to value} executes
 classification of programs besides inputs and outputs
 inputs \rightarrow $P \leftarrow P + \Delta$ \leftarrow outputs
 effect ϵ

Effect System

A set of classifications of programs
 (ie a set of effects)
 and a description of interaction
 behavior of these classifications

Semantics for Effects: Monads (a starting point)

Monad is a functor M with ^{So what about effects?!}
 unit: $Id \Rightarrow M$
 join: $M M \Rightarrow M$

M unit $\rightarrow M M$ $\rightarrow M$ \rightarrow join
 M unit $\rightarrow M M$ $\rightarrow M$ \rightarrow join

Producer Effects

$P \rightarrow \Delta \xrightarrow{\text{semantics}} [P] \rightarrow P, \text{etc}$

