

From monad transformers to the Eff monad Are we there yet?

Walter Chang

Agenda

- Referential Transparency
- Effects
- Monads
- Monad transformers
- MTL style
- The Eff monad
- Future

Referential Transparency

An **expression** is said to be **referentially transparent** if it can be replaced with its **corresponding value** without changing the program's behavior.

As a result, evaluating a **referentially transparent function** gives the same value for same arguments.

Such functions are called **pure functions**.

Effects

- Pure effects
 - Computation
- Side effects
 - Mutation of global state
 - IO
 - Exception

Monads

- Sequential computation
- All monads (that I know of) work with a SINGLE unknown type parameter (hole)
 - `Option[?]`
 - `Either[Exception, ?]`
- Namespaces
 - `Type`
 - `Value`
- Reader
- Writer

Readers

- Example 1
- Example 2
- Example 3
- Example 4

Writers

- Example 1
- Example 2
- Example 3

Monad Transformers

- Compose a specific inner monad with any outer monad
- Combine 2 monads together into just one monad
- OptionT

Readers

- Example 5
- Example 6

Writers

- Example 4
- Example 5

Readers

- Example 7
- Example 8

Writers

- Example 6
- Example 7

Monad Transformers

<i>Monad Transformer</i>	<i>Underlying</i>
<code>IdT[F[_], A]</code>	<code>F[A]</code>
<code>ReaderT[F[_], R, A]</code>	<code>R => F[A]</code>
<code>StateT[F[_], S, A]</code>	<code>S => F[(S, A)]</code>
<code>OptionT[F[_], A]</code>	<code>F[Option[A]]</code>
<code>EitherT[F[_], A, B]</code>	<code>F[Either[A, B]]</code>

Guess The Number

- Take 1

MTL Style

- <https://www.youtube.com/watch?v=knK70T4X7YE&t=19s&list=WL&index=10>
- <https://github.com/typelevel/cats/issues/1210>
- We are not there yet

Guess The Number

- Take 2

The Eff Monad

- A monad to rule all monads
- Based on the Free monads
- Primitive effects
 - Reader
 - Writer
 - State
 - ...
- Write your own effects
- Translate high-level stack to low-level stack

Guess The Number

- Take 3

Bonus

- Hangman

Yet Another Bonus

- Pong

Future

- Final tagless
- Freestyle
- Dotty
 - Coherent trait

Example Code

- <https://github.com/weihsiu/scalataiwan16>
- <https://github.com/weihsiu/pong>
-