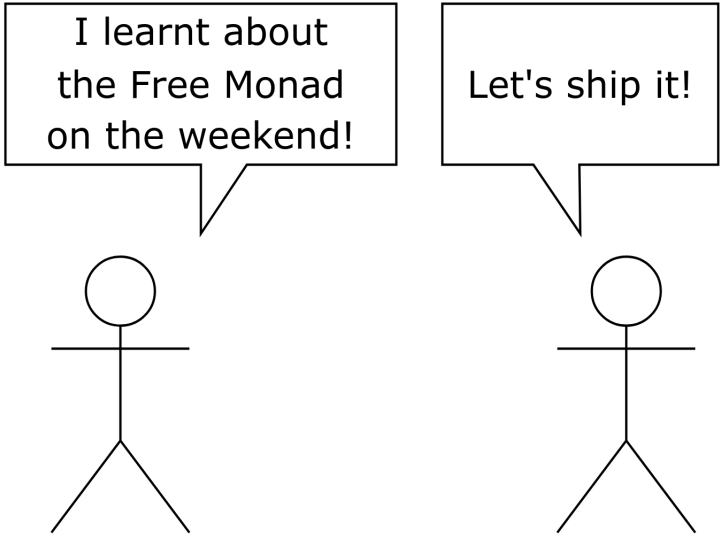
Cultivating an Engineering Dialect

George Wilson

PaidRight

george.wilson@paidright.io

6th May 2021


I learnt about
the Free Monad
on the weekend!
Let's ship it!

Project 1
- transformers
- functions, types, instances
- cabal

Project 1
- transformers
- functions, types, instances
- cabal

Project 2
- Finally tagless
- Use type classes for everything
- stack

Project 1	Project 2	Project 3
- transformers - functions, types, instances - cabal	- Finally tagless - Use type classes for everything - stack	- MTL + classy optics - Use lens for everything - nix

```
{-# language RankNTypes #-}
{-# language ExistentialQuantification #-}
{-# language ExplicitForAll #-}
{-# language TypeSynonymInstances #-}
{-# language FlexibleInstances #-}
{-# language FlexibleContexts #-}
{-# language UndecideableInstances #-}
{-# language TypeInType #-}
{-# language DataKinds #-}
{-# language ConstraintKinds #-}
{-# language GADTs #-}
{-# language PatternSignatures #-}
{-# language RecordWildcards #-}
{-# language DuplicateRecordFields #-}
```

Project 1
- transformers
- functions, types, instances
- cabal

Project 2
- Finally tagless
- Use type classes for everything
- stack

Project 3
- MTL + classy optics
- Use lens for everything
- nix

Project 4
- extensible effects
- profunctor optics
- type families, GADTs
- nix (haskell.nix)

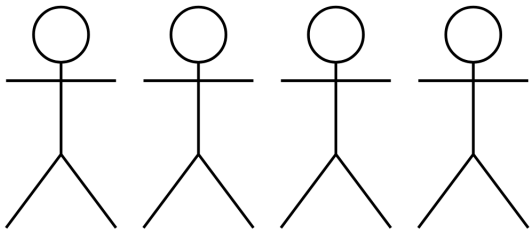
Project 1
- transformers
- functions, types, instances
- cabal

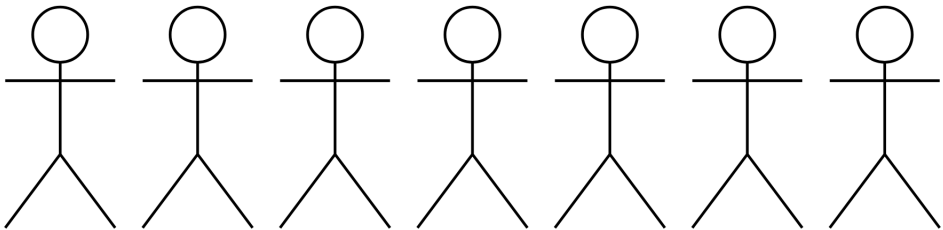
Project 2
- Finally tagless
- Use type classes for everything
- stack

Project 3
- MTL + classy optics
- Use lens for everything
- nix

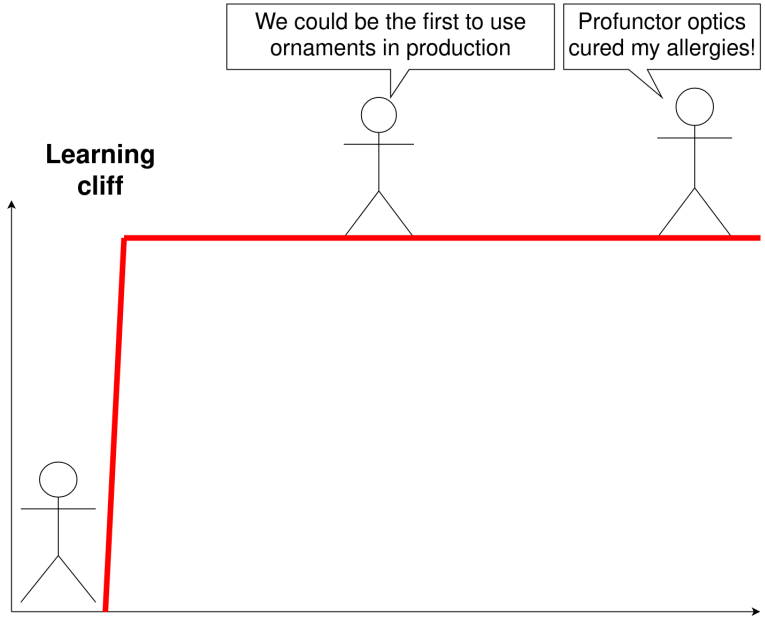
Project 4
- extensible effects
- profunctor optics
- type families, GADTs
- nix (haskell.nix)

Project 5
- write entire project at the type level
- Switch to Agda
- Transcend reality





Your code base has a learning curve


We could be the first to use ornaments in production
Profunctor optics cured my allergies!
Learning cliff


GAME OVER

How can we do better?


Simple Haskell
Pure functions and strong types
are the key to reliably delivering quality software.

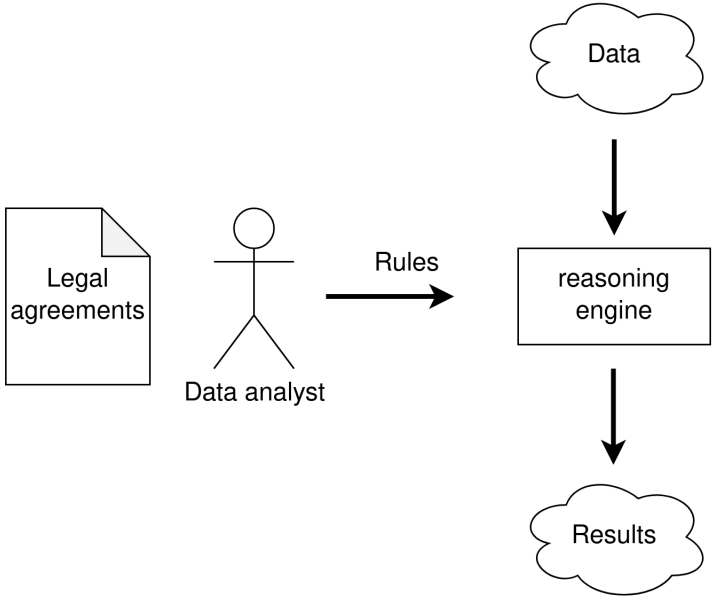

SNOYBERG
Boring Haskell Manifesto
PUBLISHED NOVEMBER 21, 2019

Power-to-weight

Make intentional decisions
informed by your problem domain
as a team

Build a learning curve

PaidRight


Data
Legal agreements
Rules
reasoning engine
Data analyst
Results

Example: GADTs

Example: GADTs

```
data Expr a where
  EInt :: Int -> Expr Int
  EBool :: Bool -> Expr Bool
  EAdd :: Expr Int -> Expr Int -> Expr Int
  EEq :: Expr Int -> Expr Int -> Expr Bool


eval :: Expr a -> a
```

Appendix B: Haskell Tools and Techniques

Generalised Algebraic Data Types (GADTs)

Recommended resource: The GHC manual's GADT section

For help, ask Isaac Elliott or Dave Laing.

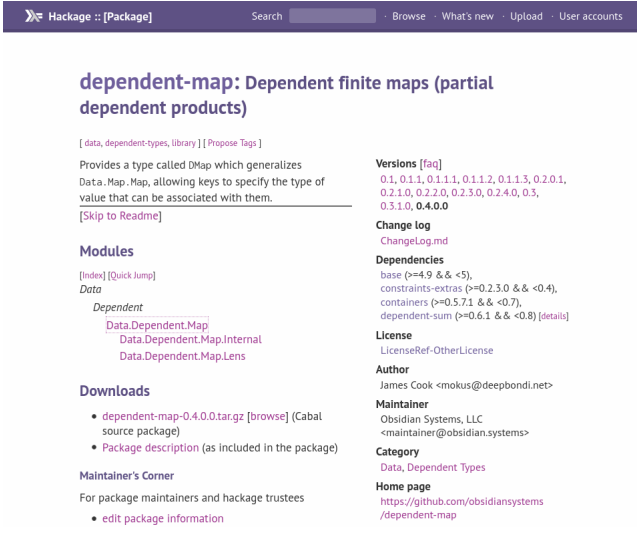
Example: Plated

```haskell
import Control.Lens.Plated

data Rule =
  ... -- 45 constructors


instance Plated Rule where
  plate = uniplate
```

Example: Database access

```haskell
import qualified Traction

getEmployeesByDepartment pool department =
  Traction.runDb pool (
    Traction.query [sql|
      SELECT e.id, e.name, e.salary
      FROM employees e
      INNER JOIN department d
        ON e.department_id = d.id
      WHERE d.id = ?
    |] (Traction.Only department)
  )
```

Example: dependent-map



Hackage :: [Package] Search · Browse · What's new · Upload · User accounts

dependent-map: Dependent finite maps (partial dependent products)

[data, dependent-types, library] [Propose Tags]

Provides a type called `DMap` which generalizes `Data.Map.Map`, allowing keys to specify the type of value that can be associated with them.

[Skip to Readme]

Modules

[Index] [Quick Jump]

Data

Dependent

Data.Dependent.Map

Data.Dependent.Map.Internal

Data.Dependent.Map.Lens

Downloads

- dependent-map-0.4.0.0.tar.gz [browse] (Cabal source package)
- Package description (as included in the package)

Maintainer's Corner

For package maintainers and hackage trustees

- edit package information

Versions [faq]
0.1, 0.1.1, 0.1.1.1, 0.1.1.2, 0.1.1.3, 0.2.0.1, 0.2.1.0, 0.2.2.0, 0.2.3.0, 0.2.4.0, 0.3, 0.3.1.0, **0.4.0.0**

Change log
ChangeLog.md

Dependencies
base (>=4.9 && <5), constraints-extras (>=0.2.3.0 && <0.4), containers (>=0.5.7.1 && <0.7), dependent-sum (>=0.6.1 && <0.8) [details]

License
LicenseRef-OtherLicense

Author
James Cook <mokus@deepbondi.net>

Maintainer
Obsidian Systems, LLC <maintainer@obsidian.systems>

Category
Data, Dependent Types

Home page
https://github.com/obsidiansystems/dependent-map

Make intentional decisions
Build a learning curve

Thanks for listening!