

Introduction to Dependent Types

Eagan Technology Unconference

Joseph Ching

September 22, 2015

Table of Contents

1 Introduction

Table of Contents

1 Introduction

2 Questions

Introduction

Syntax highlighting test reference, to be removed later.

```
-- Comment
data Maybe a = Nothing | Just a
               deriving (Show, Eq)

fmap :: Functor f => (a -> b) -> f a -> f b
map  _ []         = []
map  f (x:xs) = f x : map f xs

type family TF a :: *
type instance TF Int = Bool

{- block comment -}
foo :: Bool -> Int -> String
foo False 0 = "Bad"
foo True  0 = "Questionable"
foo False n = "Fake"
foo True  n = "Read"
```


Questions?