Omer Sinan Agacan

- Haskell developer and GHC contributor
- GHC core developer since early 2015
- Second year PhD student at Indiana University

- Programming sucks
- Compile-time performance, runtime performance, safety, abstraction ... pick ~~none~~ one.
- Things are at odds
- I focus on making runtime of functional programs better (by making compile-time worse)

```
mapOfMap f g = map f . map g
```

```
h0 f g = map f . map g

h1 f g xs = map f (map g xs)

h2 f g xs = map f (case xs of
                     []     -> []
                     (h : t) -> g h : map g t)

h3 f g xs = case (case xs of
                     []     -> []
                     (h : t) -> g h : map g t) of
              []       -> []
              (h' : t') -> f h' : map f t'
```

```
h3 f g xs = case (case xs of
                    []      -> []
                    (h : t) -> g h : map g t) of
              []        -> []
              (h' : t') -> f h' : map f t'

h4 f g xs = case xs of
              []      -> case [] of
                           []        -> []
                           (h' : t') -> f h' : map f t'
              (h : t) -> case (g h : map g t) of
                           []        -> []
                           (h' : t') -> f h' : map f t'
```

```
h4 f g xs = case xs of
               []       -> case [] of
                             []       -> []
                             (h' : t') -> f h' : map f t'
               (h : t) -> case (g h : map g t) of
                             []       -> []
                             (h' : t') -> f h' : map f t'

h5 f g xs = case xs of
              []       -> []
              (h : t) -> f (g h) : map f (map g t)
                                   -- looks similar to h1

h6 f g xs = case xs of
              []       -> []
              (h : t) -> f (g h) : h6 f g t
```