Haskell

Why

Who even uses it?

Finance

Standard Chartered

» 2.5m lines of Haskell

» Used by engineers and traders alike to write code

Facebook

Sigma: Facebook's spam/malware/abuse detection system

» Fighting spam with Haskell[1]

» The Road to Running Haskell at Facebook Scale – Jon Coens[2]

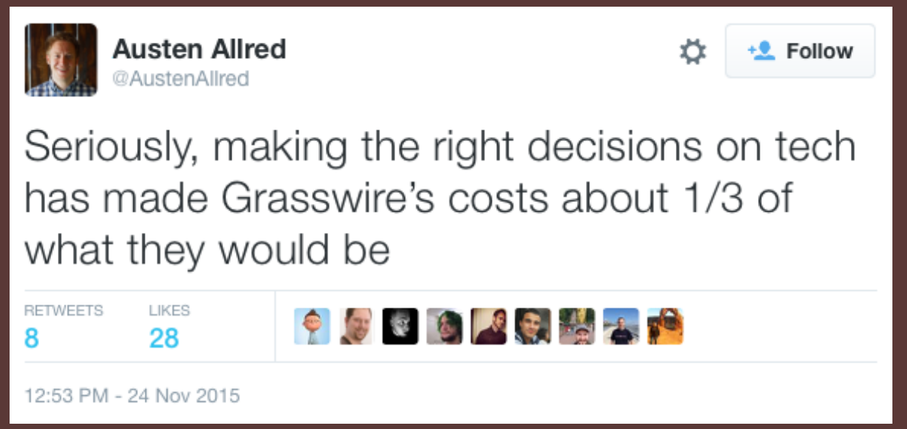
---

[1] https://code.facebook.com/posts/745068642270222/fighting-spam-with-haskell/

[2] https://www.youtube.com/watch?v=sl2zo7tzrO8

Grasswire


Austen Allred
@AustenAllred
Follow
Building in Haskell and being remote lets Grasswire play some serious moneyball. Work with really sharp people, save $, build faster.
RETWEETS
16
LIKES
33


Austen Allred
@AustenAllred
Follow
Seriously, making the right decisions on tech has made Grasswire's costs about 1/3 of what they would be
RETWEETS
8
LIKES
28
12:53 PM - 24 Nov 2015

What is Haskell?

Haskell is a standardized, general-purpose purely functional programming language, with non-strict semantics and strong static typing.

-- Wikipedia

What does it all mean??

Swift

```
func sayHello(personName: String) -> String
```

Swift

```
func sayHello(personName: String) -> String {
    let greeting = "Hello, " + personName
    return greeting
}
```

Swift

```
func sayHello(personName: String) -> String {
    if arc4random() != 0 {
        return greeting = "Hello, " + personName
    } else {
        return "Bye, " + personName
    }
    return greeting
}
```

Swift

```
func sayHello(personName: String) -> String
```

Anything can happen 🙃

Haskell

```
sayHello :: String -> String
```

Haskell

```
sayHello :: String -> String
sayHello = ("Hello, "++)
```

Haskell

```haskell
sayHello :: String -> IO String
sayHello name = do
  randomInt <- randomIO
  if randomInt == 0
    then return $ ("Bye, " ++ name)
    else return $ ("Hello, " ++ name)
```

Haskell

```haskell
sayHello :: Integer -> String -> String
sayHello i name =
  if i == 0
    then return $ ("Bye, " ++ name)
    else return $ ("Hello, " ++ name)

-- Some other code/function would deal with the IO
```

TDD?

Type Driven Development

There's a saying in Haskell: "If it compiles, it works"

Type Driven Development

There's a saying in Haskell: "If it compiles, it works"

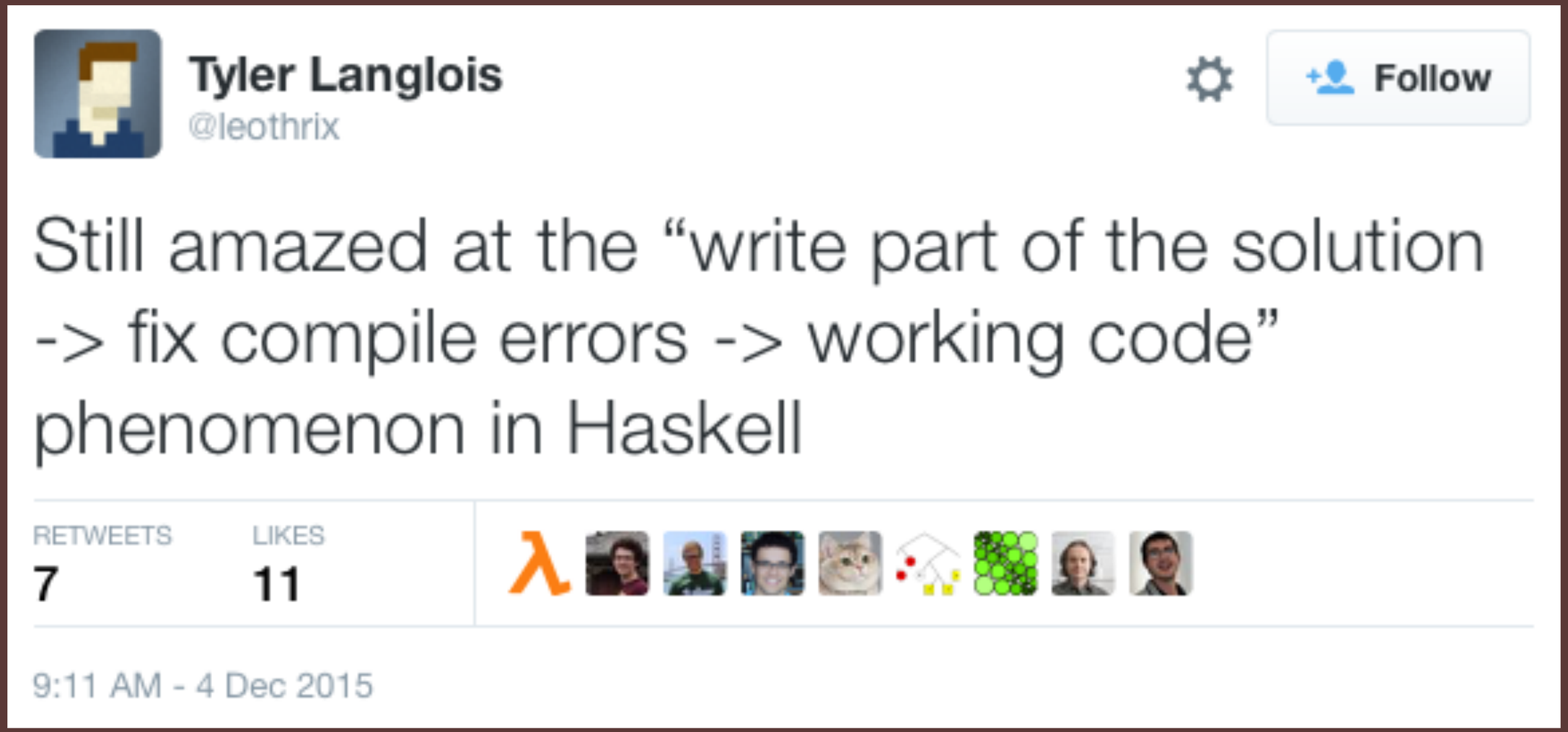
Development process ends up being:

» Write the types

» Write code until compiler stops complaining

» …

» Profit!



https://twitter.com/leothrix/status/672825545273565184

Refactoring

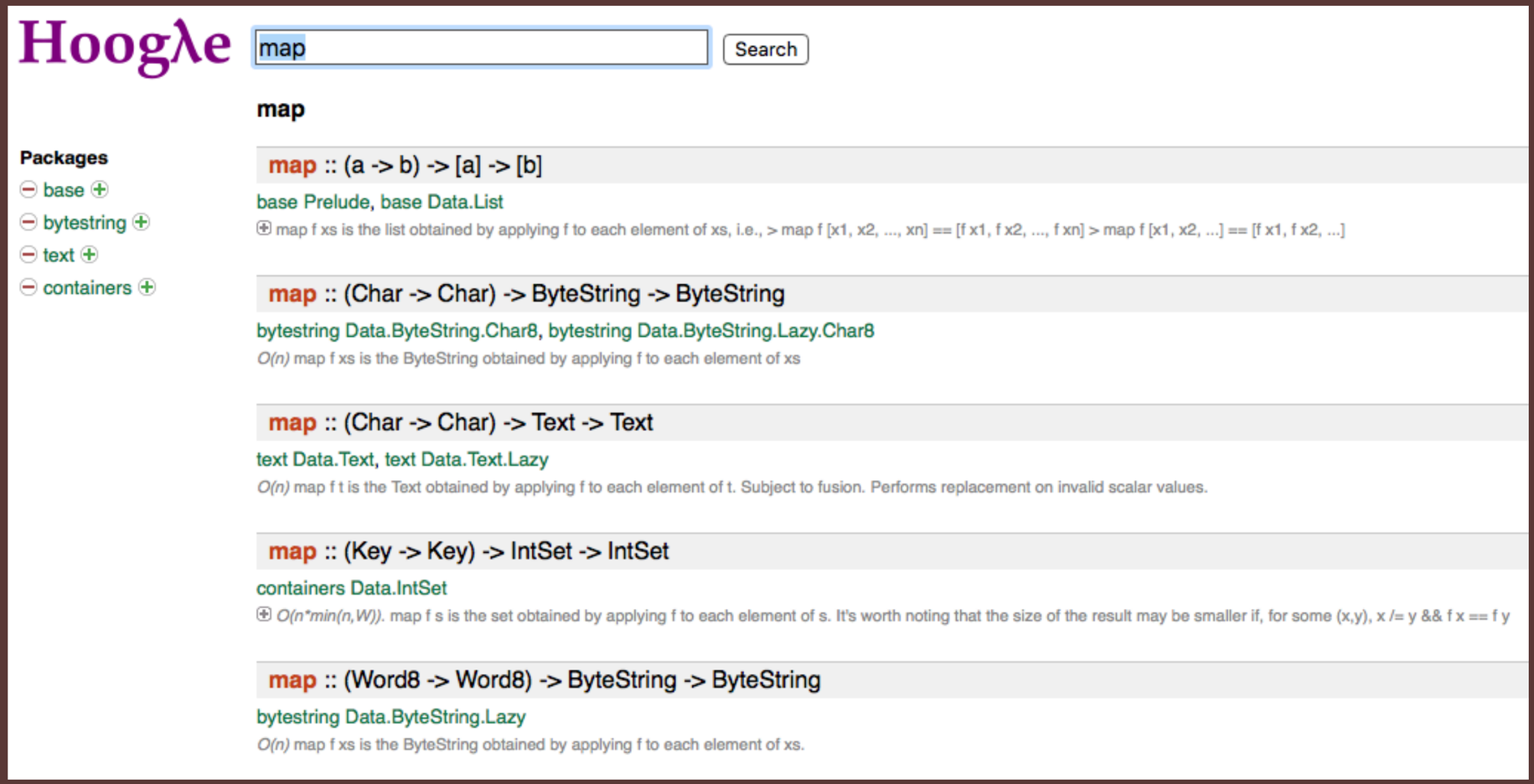
"If it compiles, it works"

Testing

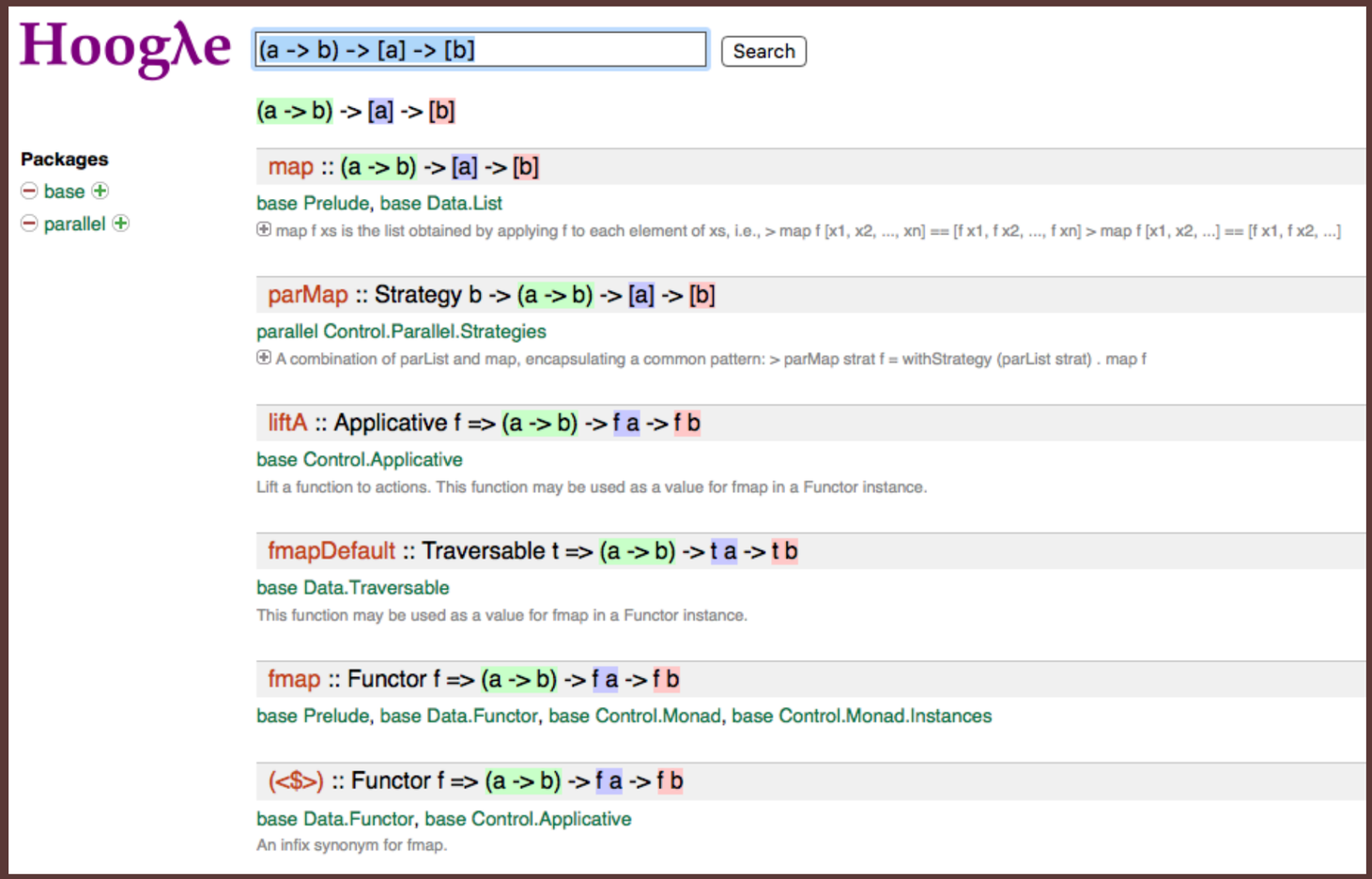
- More testable code
- Less things to test (especially vs. JS/Ruby/etc.)
- State of the art tools (QuickCheck)

Tooling

Hoogle/Hayoo/etc.



Hoogle/Hayoo/etc.



REPL

- Incremental development
- Type inspection
- Type holes (_)

And more…

- Code reuse
- Learning curve
- Speed
- Parallelism (STM)
- New abstractions

How do I start?

- haskellbook.com
- CIS194
- #haskell/#haskell-beginners on freenode
- Santa Monica Haskell meetup

Thanks!