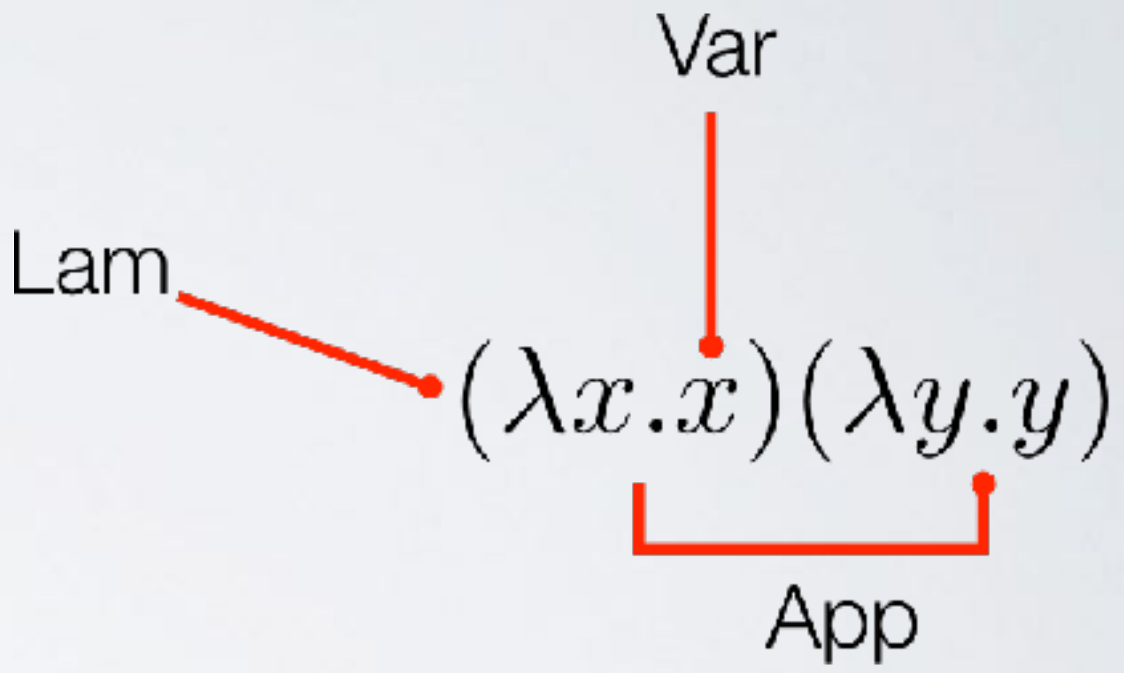
HASKELL简介

以及在滴滴的经验交流分享

— 韩冬

LONG LONG AGO…

- Alan Turing => 图灵机
- Alonzo Church => λ演算
- Church => Alan Turing’s PhD supervisor @ Princeton University. (WTF ???)



编程语言大爆发

- 1958年John McCarthy发表了LISP(List Processor)。之后LISP家族生生不息，现存的实现包括SBCL(Common Lisp), Racket(Scheme), Clojure….
- 1970s 函数式编程语言ML (MetaLanguage)诞生，在接下来的几十年衍生出Standard ML，Caml，Ocaml…
- 1969 ～ 1973@贝尔实验室：C语言诞生, 1978年Brian Kernighan 和 Dennis Ritchie出版巨著《The C Programming Language》， 史称K&R C。
- 至今C系语言已成编程主流，来随口说一个?

但是，牛逼的计院教授们觉得这些语言还是弱爆了，
于是在1987 年聚在一起，要设计一个吊到爆的编程语言

HASKELL

- 1990年发表了第一版《The Haskell Language Report》
- 学术界日益开始开始使用haskell来取代ML作为主流的函数式编程语言。
- 1999～2002发表《The Haskell 98 report 》
- 2010发表《The Haskell 2010 report 》
- Haskell的语言规范仍然在高速进化。

HASKELL现状

主要实现：

- GHC -100+ active commiter，2+ minor version per year
- Compiler -139955 loc / RTS - 48450 loc C/C— (2011)
- Shepherds - Simon Peyton Jones(MS) & Simon Marlow(FB)

业界应用：

- facebook - sigma
- standard chart - Mu
- microsoft - bond/ghc/…

社区建设：

- hackage -1w+ package, 10w+ download per day
- stackage -10+ nightly build per month

COMPILER介绍

- Full type inference with system-FC
- High rank type and GADT support
- Kind equality, unlifted type support.
- Cross-module inlining and optimisation
- Full LISP alike template
- Generics
- Multi-pass simplifier, float in/out optimisation
- LLVM codegen & Native codegen & C codegen
- Interactive REPL(GHCI)

RTS介绍

- GMP
- PrimOp
- Light weight thread
- FFI
- IO Manager
- Heap / Thread Profiler
- STM
- Parallel GC
- Compact Region(Comming in 8.2+)

代数类型

```
data Bool = False | True

data Maybe a = Just a | Nothing
case ma of Just a -> …
                Nothing -> …

data (,) a b = (,) a b
let (x, y) = …

data [] a = [] | (:) a [a]

map :: (a -> b) -> [a] -> [b]
map f [] = []
map f (x:xs) = f x : map f xs
```

类型类

```haskell
class Eq a where
    (==), (/=) :: a -> a -> Bool

    x /= y  = not (x == y)
    x == y  = not (x /= y)

instance Eq A where
    (==) :: A -> A -> Bool
    x == y  = …

-- 判断列表中是否存在某个元素
elem :: Eq a => a -> [a] -> Bool
elem _ [] = False
elem y (x:xs) = if x == y then True
                                    else elem y xs
```

高阶函数

```
(.) :: (b -> c) -> (a -> b) -> a -> c
f . g = \ x -> f (g x)
infixr 9  .

(unlines . map reverse . lines) “hello\nworld”
— "olleh\ndlrow\n"
```

MONAD

```
getLine :: IO String
putStrLn :: String -> IO ()

newtype IO a  =
    IO (State# RealWorld -> (#State#
RealWorld, a#))

getLine :: RealWorld -> (RealWorld, String)
putStrLn :: String -> RealWorld ->
(RealWorld, ())
```

MONAD

```
class  Monad m  where
    -- | Sequentially compose two actions, passing any
value produced
    -- by the first as an argument to the second.
    (>>=)       :: forall a b. m a -> (a -> m b) -> m b

instance  Monad IO  where
    (>>=)     = bindIO

bindIO :: IO a -> (a -> IO b) -> IO b
bindIO (IO m) k =
     IO (\ s -> case m s of (# new_s, a #) -> unIO (k a)
new_s)
```

结构控制函数

```
map :: (a -> b) -> [a] -> [b]
mapM :: (a -> m b) -> [a] -> m [b]

forM / filterM / replicateM / foldM /
zipWithM / forever …
when / unless / many / some / msum …
async / wait / forkMapM / mapConcurrently …
```

在滴滴HASKELL作了些什么？

- 网站后端
- 数据库中间件

举几个栗子

- 后端API
- SSO接入
- HTML模版
- 二进制解析

谢谢！

《Magic Haskell》