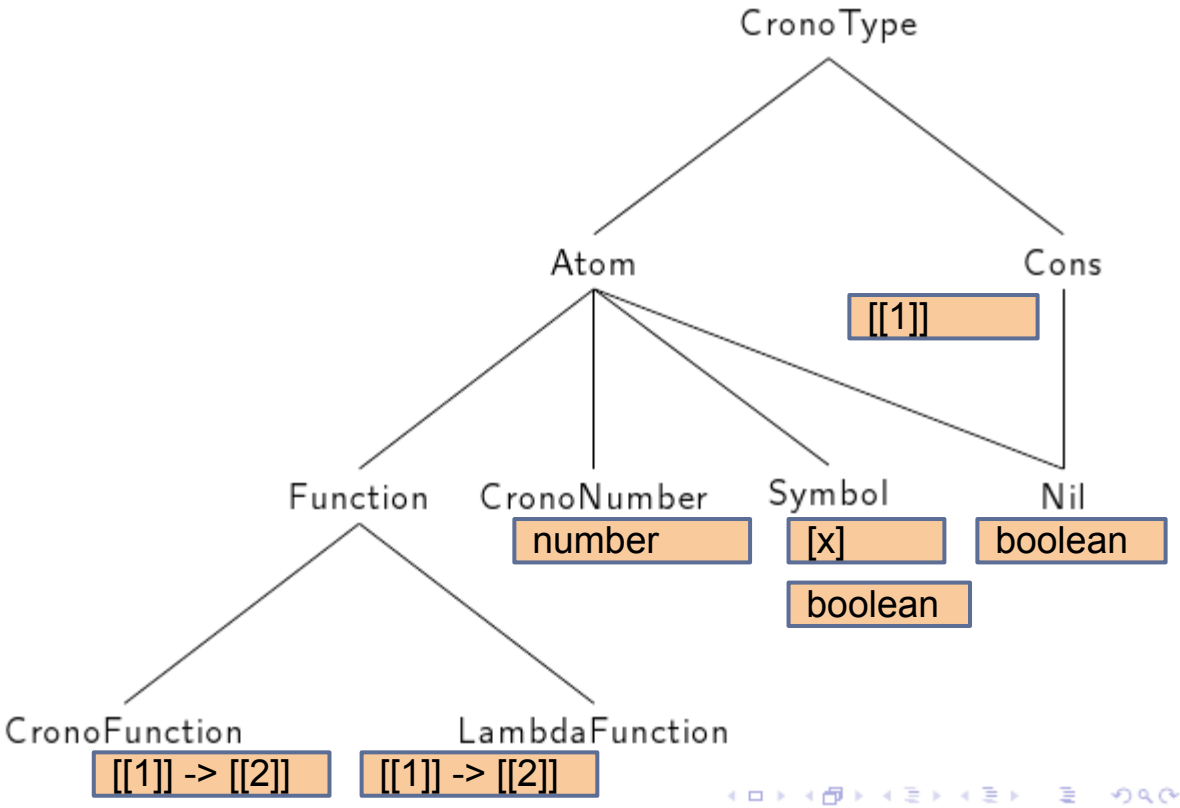
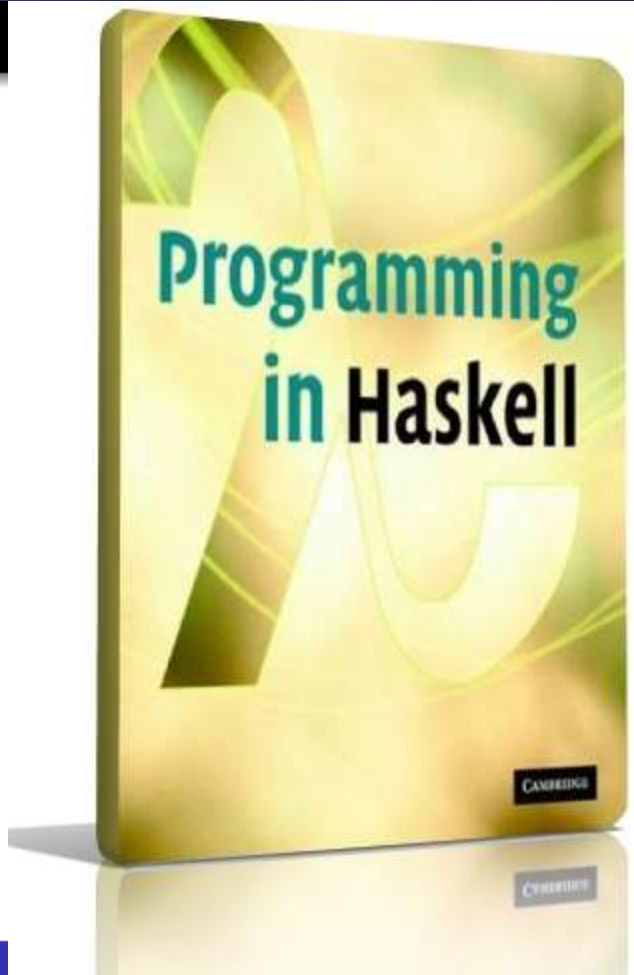
GETCrono

By Graham Benevelli, Tri Nguyen, and Edwin Edge

Types





Quick Introduction to Crono & Type Inferencing:

- Programming language based on Lisp
- Haskell can determine type of functions given no other information whatsoever

Steps of Type Inferencing

- CronoTypeConstraint
- CronoTypeVar
- CronoConstraintCreator

- Blindy assign each item (Cons, number, etc.) an identifier that is unique and ambiguous
- Build CronoTypeConstraints off of different CronoTypes
  - a number, symbol, cons, or a known function
- Get back a series of constraints
  - i.e. [[1]] = [y] -> [z]
  - i.e. [[2]] = number
- feed to unification process

The Unification Algorithm

- Two stacks
  - TODO stack, finished stack
- Pull constraints off TODO
  - Substitute constraint into both stacks
- Place constraint onto finished stack
- Rinse and repeat
- Example (define double (x) (* x 2))

Concepts

- Type Inferencing
- Haskell
- Unification
- Abstract Syntax Tree
- Lisp

If Time Other Examples

Examples

- (+ 5 4)
- (= 1 2)
- (if (= 1 1) 9 8)
- (define even (x) (= x (* 2 (/ x 2))))
- (define fact (n) (if (= n 1) 1 (* n (fact (- n 1))))))