Full Lisp Interpreter

Faraaz Ismail

Brett Smith

Paul Glass

December 4, 2012

Parser

* Parser.jj
  * Create an abstract syntax tree
  * Converts lets to lambdas applications

ASTNode

* ASTNode.java
  * The abstract syntax tree
  * Visitable
  * IdNode, NumNode, FunNode, AppNode, etc…

Visitor

* Visitor.java
  * subclass PrintVisitor prints out the AST
  * subclass Interpreter inteprets the AST

Interpreter

* Interpreter.java
  * Visits the AST
  * Constructs an Environment as it goes
  * Evaluates closures

Environment

* Environment.java
  * A LinkedHashMap, useful for printing
  * Map Strings to EnvValue objects

EnvironmentValue

* EnvironmentValue.java
  * Num class
  * List class
  * Closure class

Features

* Features
  * Recursion
    * “letrec”
    * Create a cyclic environment
  * Lists with car, cdr, cons
  * Static or dynamic scoping
  * Whole number combinators
    * Y combinator: another way to do recursion