The Reference Manual for ƎL

Version 3.14

The ƎL Development Team

March 5, 2016

Contents

1   Language Guide   1
    1.1   types   1
    1.2   constants   1
    1.3   list   1
    1.4   tuple   1
    1.5   function   1
    1.6   control flow   1

2   Usage   2
    2.1   declare constants   2
    2.2   declare function   2
    2.3   pattern match   2
    2.4   Input and Output   2
    2.5   sample code   2

Chapter 1

Language Guide

§1.1   types

Ǝ provides fundamental types, including `Int` for integers, `Double` for floating-point values, `String` for textual data, and `Bool` for Boolean value. Ǝ also provides Tuple and List as described in §1.3, §1.4.

§1.2   constants

You cannot use variable. Alternatively, Ǝ provides constant. The value of a constant cannot be changed once it is set in the future. This constraint ensures referential transparency.

§1.3   list

§1.4   tuple

§1.5   function

§1.6   control flow

Chapter 2

Usage

§2.1   declare constants

§2.2   declare function

Listing 2.1   function $fibo(n)$

```
1  fibo(n) :: Int -> Int
2   = 0 [n == 0]
3   = 1 [n == 1]
4   = fibo(n-1) + fibo(n-2)
```

§2.3   pattern match

§2.4   Input and Output

§2.5   sample code

Here is the sample code, which returns a fibonacci number.

Listing 2.2   sample code

```
1  fibo(n) :: Int -> Int
2   = 0 [n == 0]
3   = 1 [n == 1]
4   = fibo(n-1) + fibo(n-2)
5  where
6   who :: Int
7   =5
8   hoge :: Int
9   =6
10
11 m :: Double
12 =2
```