

Symbolic & Declarative Computing / Artificial Intelligence (COS5012-B)

November 02, 2016

Week 6 Lab2

LAB MANUAL: HIGHER ORDER FUNCTIONS

In this tutorial you will learn about

- Higher Order Functions (fold and its variants)

Fold

- Write a program to calculate the sum of the elements of an Integer list.
- Write a program to calculate the product of the elements of an Integer list.
- Write a program to find the maximum element of an Integer list.
- Write a program to find the minimum element of an Integer list.

Now write the following commands in Hugs

```
Hugs> foldl1 (+) [1,2,3,4,5]
Hugs> foldl1 (*) [1,2,3,4,5]
Hugs> foldl1 max [1,2,3,4,5]
Hugs> foldl1 min [1,2,3,4,5]
```

Fold takes in a function and folds it in between the elements of a list

Two flavours:

Right-wise fold: `foldr op b [x1,x2,x3] => x1 op (x2 op (x3 op b))`

Left-wise fold: `foldl op b [x1,x2,x3] => ((b op x1) op x2) op x3`

Where 'b' is the base element and 'op' is the binary operator

Other variants

Both `foldr` and `foldl` comes with other variants `foldl1` and `foldr1`.

Exercise

- Write down your own implementation of all variants of fold
- Define `length`, which returns the number of elements in a list, using `foldr`.
- Redefine it using `foldl`.