

Higher Order Operator Overloading Lifting

by Sven Nilsen, 2019

Assume the following equation using Higher Order Operator Overloading (HOOO):

$$f = a + g$$

$$\begin{aligned} f &: X \rightarrow \text{real} \\ g &: X \rightarrow \text{real} \\ a &: \text{real} \end{aligned}$$

Expanded to full closure/lambda form:

$$\lambda(x : X) = f(x) == a + g(x)$$

Higher order operator overloading lifting is done by replace `a` with a function `h`:

$$h := \lambda(x : X) = a$$

Such that the expanded version can be written in an equivalent form:

$$\lambda(x : X) = f(x) == h(x) + g(x)$$

Since this is an equation with only functions, it can be written in pure HOOO form:

$$f = h + g$$

Now, by swapping all occurrences of `a` with `h`, one can think of `a` being transformed into `a'`:

$$f = a' + g$$

$$a' := \lambda(x : X) = a$$

Or, simply:

$$f = a + g$$

This makes any mixture of function and scalar types possible, as an extension to pure HOOO syntax.