Higher Order Operator Overloading and Self Reference

by Sven Nilsen, 2019

Higher Order Operator Overloading[1] states that when you have functions of the same input, you can overload any operator on the return values of the functions:

$$g(f_0, f_1) := \backslash(a : A) = g(f_0(a), f_1(a))$$

This makes it possible to treat functions as values you can compute with, constructing new programs.

Assume that you have the following function (the set of points on a unit circle[2]):

$$\text{circle} := \backslash(x : \text{real}, y : \text{real}) = x^2 + y^2 <= 1$$

Here, there are two arguments, \`x\` and \`y\`.

To reference \`x\` and \`y\` one can do the following:

$$x := \backslash(x : \text{real}, y : \text{real}) = x$$

$$y := \backslash(x : \text{real}, y : \text{real}) = y$$

With Higher Order Operator Overloading[1], the following expression re-constructs the unit circle:

$$x^2 + y^2 <= 1 \qquad <=> \qquad \backslash(x : \text{real}, y : \text{real}) = x^2 + y^2 <= 1$$

What happens here is that \`x\` and \`y\` are defined as functions that refers to themselves in the context of the function argument tuple \`(x : real, y : real)\` that is used to construct new functions.

This technique is called “self reference” and states that:

- If we have a function for every input ...
- … and overload all operators with Higher Order Operator Overloading …
- … then the original function can be re-constructed by re-interpreting the original function

By passing \`x\` and \`y\` as self referential functions, the following is true:

$$\text{circle}(x, y) <=> \text{circle}$$

One can use this technique to prove the commutative property[3] of the \`circle\` function in a new way:

$$\text{circle}(y, x) <=> \text{circle} \qquad <=> \qquad \forall\ x, y \{ \text{circle}(x, y) = \text{circle}(y, x) \}$$

References:

[1] “Higher Order Operator Overloading”
Sven Nilsen, 2018
https://github.com/advancedresearch/path_semantics/blob/master/papers-wip/higher-order-operator-overloading.pdf

[2] “Unit circle”
Wikipedia
https://en.wikipedia.org/wiki/Unit_circle

[3] “Commutative property”
Wikipedia
https://en.wikipedia.org/wiki/Commutative_property