Constructing functions

Tom Rochette <tom.rochette@coreteks.org>

November 2, 2024 — 36c8eb68

0.1 Context

0.2 Learned in this study

0.3 Things to explore

1 Overview

Let's assume that all that is available to us to write a new function is a list of existing functions.

First we want look at parameterless functions.

A parameterless function can do three of the following things:

- Return data (an intrinsic value or some structure/object)
- Call other parameterless functions
- Call other functions with parameters by instantiating the required arguments internally through a call to other parameterless functions

1.1 Return data (Instantiator)

- It can return consumable data
- It can instantiate static data

1.2 Call other parameterless functions (Delegator)

Overall, the "functions" of such a function are:

- Encapsulating functions ($f() = g()$)
- Encapsulating sequence of functions ($f() = g(), h(), i()$)
- Function composition/Chaining functions calls ($f = g \circ h \circ i = g(h(i()))$)
- Recursive function calls ($f = f^n$, $f = \circ^n f$,$f = \underbrace{f \circ f \circ \cdots \circ f}_{n}$)

1.3 Call other functions with parameters

Here we observe that calling functions with parameters only amounts to calling the appropriate instantiator and passing the result to the function expecting an argument.

We can consider the following program

to be convertible into

2 See also

3 References