The Id-Unit Function

by Sven Nilsen, 2019

In this paper I define a function that constructs a function tuple erasing a single argument.

The `id_unit` function is defined as following[1]:

id_unit := \(n : nat) = \(i : nat) = ∏ j [0, i) { id } × unit × ∏ j [i+1, n) { id }

Here, the `∏` loop means taking the Cartesian product[2] of the loop body.

The motivation for this function is to erase a single argument in normal paths[3].

For example, a function `f` taking 4 arguments can erase the 3rd argument with `id_unit(4)(2) → id`:

f[id_unit(4)(2) → id] : T × T × () × T → U

$f : T^4 \to U$

This normal path does not exist if `f` depends on the 3rd argument for some input.

When a subscript index is used with the id unit function, the size of the tuple is inferred:

$f[id\_unit_2 \to id] \iff f[id\_unit(4)(2) \to id]$

When multiple parameters are erased, a subscript interval can be used as a shorthand:

$id\_unit_{[0, n)} \iff id\_unit_0 \cdot id\_unit_1 \cdot \ldots \cdot id\_unit_{n-1} \iff unit^n$

One can also use the interval notation with the last argument:

id_unit(4)[0, 3) = (unit, unit, unit, id)

References:

[1] “Alphabetic List of Functions – Standard Dictionary for Path Semantics”
Sven Nilsen, 2017
https://github.com/advancedresearch/path_semantics/blob/master/papers-wip/alphabetic-list-of-functions.pdf

[2] “Cartesian product”
Wikipedia
https://en.wikipedia.org/wiki/Cartesian_product

[3] “Normal Paths”
Sven Nilsen, 2019
https://github.com/advancedresearch/path_semantics/blob/master/papers-wip/normal-paths.pdf