Catuṣkoṭi Existential Lift

by Sven Nilsen, 2021

In this paper I introduce a proof technique that exploits conditional branches of functions depending on some sub-type of the input resulting in indeterminate results encoded in many-value logic.

The result of this paper is the following:

f[g](x) = both   =>   g . f{(= x) . g} => both   Catuṣkoṭi Existential Lift

The Collatz function[1] is defined as following:

collatz(x : nat ∧ (> 0)) = if x % 2 = 0 { x / 2 } else { 3 * x + 1 }

It is known that for odd numbers, the result is even (in Path Semantical notation[2]):

collatz[even](false) = true

However, for even numbers, the result is indeterminate:

∃collatz[even]{(= true)} <=> \true

Using an existential path equation, this can be simplified.

collatz[even]{(= true)} => \\true
collatz[even]{(= true)} => both

For more information, see the paper "Catuṣkoṭi Existential Path Equations"[3].
Since the domain constraint `(= true)` is concrete, one can write this as:

collatz[even](true) = both

In general:

f[g](x) = both   <=>   f[g]{(= x)} => both
f[g](x) = neither   <=>   f[g]{(= x)} => neither

However, since there is a symmetric path `f[g]` and the existential path equation is indeterminate with respect to the function identity `f[g]{(= x)}` under `both`, one can do the following trick:

f[g](x) = both   =>   g . f{(= x) . g} => both

∵   f[g]{(= x)} => both
∴   g . f[g → id]{(= x)} => both
∴   g . f[id → id]{(= x) . g} => both
∴   g . f{(= x) . g} => both

References:

[1] "Collatz conjecture"
Wikipedia
https://en.wikipedia.org/wiki/Collatz_conjecture

[2] "Path Semantics"
AdvancedResearch
https://github.com/advancedresearch/path_semantics

[3] "Catuṣkoṭi Existential Path Equations"
Sven Nilsen, 2021
https://github.com/advancedresearch/path_semantics/blob/master/papers-wip2/catuskoti-existential-path-equations.pdf