Inverse Zero Order Existential Paths

by Sven Nilsen, 2017

The first order existential path of `f` is written `∃f`. Assume that you know `∃f`, what should you call `f` without referring to it by name? I call this the *zero order existential path*. This terminology is useful when you want to express relationships between functions through their existential paths.

An *inverse zero order existential path* is a function `h` that outputs only values that are not output by the zero order existential path `f`:

$$∃h <=> ¬∃f$$

The negation symbol `¬` is interpreted literally, by flipping `true` to `false` and `false` to `true`. If `h` is the inverse zero order existential path of `f`, then `f` is the inverse zero order existential path of `h`:

$$¬∃h <=> ∃f$$

The second order existential path of both `f` and `h` must be `$true_1$`, since `f` halts and it must have some values it does not output, such that `h` can output them instead.

$$∃∃f <=> true_1$$
$$∃∃h <=> true_1$$

Otherwise, `h` would never halt, since there are values that are not returned, but none that are returned.

For every halting function `f`, one can prove the existence of the inverse zero existential path by showing that `∃∃f <=> $true_1$`. To do this it is sufficient to find two objects `A` and `B` for which:

$$∃f(A) = true$$
$$∃f(B) = false$$

Therefore, because the requirement of two such objects existing, the smallest output type of `f` is a boolean type:

$$f : A → bool \qquad \text{(smallest type)}$$

Notice that `∃h` is mentioned in the existential path relationship, but never `h` itself. This means that one is allowed to change the input type of `h` arbitrary, as long the returned values are those not returned by `f`:

$$f : A → C$$
$$h : B → C$$

If both `f` and `h` have an inverse, their domains are smaller than their output type. This is because there must be a one-to-one correspondence between input and output, and the presence of an inverse zero order existential path means the output type contains a value that is not returned by either function.