Non-Composition of XOR Trick

by Sven Nilsen, 2020

In this paper I show that the Abstract Transport XOR Trick can not be composed.

The following is a proof in Path Semantical Logic[1], using Abstract Transport XOR Trick[2]:

(a, b, c) (A, B, C)
a ⩴ b, a(A), b(B), c(C) => C

Where the tuple `(a, b, c)` has level 1 and the tuple `(A, B, C)` has level 0.
The notation `a(A)` means `a=>A` where `A` is at a lower level.

The problem is that `C` is provable without providing any evidence of `c`.
Any proposition in level 0 that is associated with some proposition in level 1,
that is neither `a` or `b`, will collapse to `true`.

One can include `c` and `C` into the non-collapsing region by using OR:

(a ⩴ b) ∨ (b ⩴ c)

However, if one attempts to use AND:

(a ⩴ b) ∧ (b ⩴ c)

One can prove:

A ∧ C

Therefore, AND is too strong to avoid collapse, while OR is too weak to use as composition.

References:

[1] “Path Semantical Logic”
AdvancedResearch, reading sequence on Path Semantical Logic
https://github.com/advancedresearch/path_semantics/blob/master/sequences.md#path-semantical-logic

[2] “Abstract Transport XOR Trick”
Sven Nilsen, 2020
https://github.com/advancedresearch/path_semantics/blob/master/papers-wip/abstract-transport-xor-trick.pdf