Implication Theorem

by Sven Nilsen, 2020

In this paper I present an implication theorem found in Path Semantical Logic.

The Implication Theorem is a proof in Path Semantical Logic[1]:

(a, b) (T, U):
a=>b, a(T), b(U) => T=>U

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

Naturally, one can think about this as `a` and `b` being two sub-types:

a <=> [f] x	f : T → X
b <=> [g] y	g : U → Y

The types `T` and `U` are generic (type variables), meaning they can be sub-types of each other.

The implication `a=>b` can be thought of as `a` being a sub-type of `b`:

a=>b   ∀ c { (c : [f] x) => (c : [g] y) }

As a consequence, the type `T` is a sub-type of `U`.

For example, `(< 5)` is a sub-type of `(< 10) | (= “hello”)`.
Therefore, the input type of `(< 5)` must be a sub-type of the input type of `(< 10) | (= “hello”)`.
In `(< 5)`, there are no strings. It can a sub-type of `(< 10) | (= “hello”)`, but not vice versa.

The Implication Theorem says that implications carries over to lower levels, just like equalities.

Like the core axiom of Path Semantics[2], the Implication Theorem is an almost-tautology[3] in normal propositional logic. The only case where it fails is the following:

a = 0   b = 0   T = 1   U = 0

This might be interpreted as when `a` and `b` are undefined symbols, their meaning must be the same. When `T` is a defined meaning of `a`, it can not imply an undefined symbol `U`. It is impossible mean something undefined from a defined meaning.

References:

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

[2] "Path Semantics"
Sven Nilsen, 2016-2019
https://github.com/advancedresearch/path_semantics/blob/master/papers-wip/path-semantics.pdf

[3] "Tautology (logic)"
Wikipedia
https://en.wikipedia.org/wiki/Tautology_(logic)