Equivalence Paths

by Sven Nilsen, 2018

Assuming semantics of constrained functions, an equivalence path is an "auto-constraint" operator:

$$\sim\ := \backslash(f : A \to B) \to (A \to B) = f\{\backslash(a : A) = |[f]\ f(a)| == 1\}$$

This unary operator `~` is called "universal equivalence path".

All structure preserving functions have themselves as equivalence paths:

$$\sim f \iff f \qquad \text{if `f` is structure perserving (`f` got an inverse)}$$

Similarly, applying `~` twice has no effect:

$$\sim\sim f \iff \sim(\sim f) \iff \sim f$$

There is a short version for accessing the two functions using `~` as a binary operator:

$$\begin{array}{lllll} \forall\sim f & \iff & f \sim 0 & \iff & \backslash(a : A) = |[f]\ f(a)| == 1 \\ \exists\sim f & \iff & f \sim 1 & \iff & \backslash(b : B) = |[f]\ b| == 1 \end{array}$$

When a function `f` is constrained to `f ~ 0`, its existential path is `f ~ 1`:

$$\exists f\{f \sim 0\} \iff \exists\sim f \iff f \sim 1$$

If one knows `f ~ 1`, then one can construct `f ~ 0` by composing `f` with `f ~ 1`:

$$f \sim 1 \cdot f \iff \forall\sim f \iff f \sim 0$$

One can think of `~f` as crossing out all input-output pairs that intersect.
Using constrained functions, it is possible to fine-tune any desired equivalence path:

~f	**~f{(¬= 3)}**	**~f{(¬= 2)}**	**~f{(¬= 2) ∧ (¬= 5)}**
0 → a	0 → a	0 → a	0 → a
1 → b	1 → b	1 → b	1 → b
~~2 → c~~	2 → c	~~2 → c~~	~~2 → c~~
~~3 → c~~	~~3 → c~~	3 → c	3 → c
~~4 → d~~	~~4 → d~~	~~4 → d~~	4 → d
~~5 → d~~	~~5 → d~~	~~5 → d~~	~~5 → d~~