Formal Definition of Normal Paths

by Sven Nilsen, 2020

In this paper I give the formal definition of normal paths in path semantics.

A homotopy path[1] `H` is a map between two continuous functions `f` and `g` such that:

$$\forall x \{ H(x, 0) = f(x) \} \quad \wedge \quad \forall x \{ H(x, 1) = g(x) \}$$

$$H : X \times [0, 1] \to Y$$
$$f : X \to Y$$
$$g : X \to Y$$

It follows from this definition that there can be paths between paths, e.g. `X <=> [0, 1]`.

When only the end points are considered, one can use a `𝔹` (boolean) instead of `[0, 1]`:

$$(H\ \text{false}) \Leftrightarrow f \quad \wedge \quad (H\ \text{true}) \Leftrightarrow g$$

$$H : X \times \mathbb{B} \to Y$$

This means that a homotopy path represents a binary relation[2] of an ordered pair:

$$(f, g)$$

Where the truth values of of these binary relations defines the space of all paths.
When the relation is `true`, there exists a path, but when the relation is `false`, there exists no path.
In constructive mathematics[3] one can not prove that the relation is `true` from `¬false` and vice versa.
This means that there exists some construction that represents the path: The proof of the path.

In Homotopy Theory[4], the proof of the path is constructed by the homotopy path `H`.
However, this is looking at the path from the outside.
From within the path, there is only `X → Y`, which can be proved as following.

By using function currying[5], one can transform this type into:

$H : X \to (\mathbb{B} \to Y)$ the choice of some function `X → 𝔹` determines `X → Y`

The type `X → 𝔹` is a set over `X`. The complexity[6] in number of bits of this set is `$\log_2(|X|) + 1$`.
From within the path, there is only `x`, which has type `X` and maps into some value of `Y`.
Therefore, from within the path, there is only `X → Y`, which is an ordinary function type.

The change in linear input complexity from inside to outside is `$2^{\log2(|X|) + 1} - 2^{\log2(|X|)} = |X|$`.
From using this, one can derive that the change in linear output complexity is `$|Y|^{|X|} = |X \to Y|$`.

Therefore, there should exist a space similar to “computation” that relates functions to each other.

Where is this space?

The insight of Homotopy Type Theory[7] is: Dependent types[8] can model proofs of Homotopy Theory.

This means that the direction of this space, seen from inside these paths, is analogous to dependent types. However, dependent types only describe a single step of “computation” in this direction.

What is beyond dependent types?

Using the analogy of complex numbers[9]:

1	the base of normal computation
i	the base of path-space computation

For all complex numbers in base `**e**`, the following property holds:

$$A\mathbf{e} + B\mathbf{e} = (A + B)\mathbf{e}$$

The analogy law for functions in base `**e**` is the following:

$$f\mathbf{e} \cdot g\mathbf{e} = (f \cdot g)\mathbf{e}$$

Once the base of path-space computation is found,
one can use normal function composition rules in the new dimension.

Now, if one looks at commutative diagrams[10]:

$$\begin{tikzcd} \circ \arrow[r, "g_0"] \arrow[d, "f"'] & \circ \arrow[d, "h"] \\ \circ \arrow[r, "g_1"] & \circ \end{tikzcd} \qquad g_1 \cdot f \iff h \cdot g_0$$

For every point, or value, they behave as normal computation, which satisfies `$f\mathbf{e} \cdot g\mathbf{e} = (f \cdot g)\mathbf{e}$`.
Yet, the point-wise computation is not the space one is looking for, since this is covered by `$X \to Y$`.

From within the paths, the total complexity can be thought of as a product:

$$|X \times \mathbb{B} \to Y| = |Y|^{|X \times \mathbb{B}|} = |Y|^{2|X|} = |Y|^{|X|} \cdot |Y|^{|X|} = |X \to Y|^2$$

This means that the “missing parameter” one is looking for is the “function surface” `$g_0 \to g_1$`:

$$H := f[g_0 \to g_1]$$

The constructive proof is `h` to represent `H` from within homotopy paths (called “normal paths”):

$$f[g_0 \to g_1] \iff h \qquad\qquad \text{Q.E.D.}$$

References:

[1] "Homotopy"
Wikipedia
https://en.wikipedia.org/wiki/Homotopy

[2] "Binary relation"
Wikipedia
https://en.wikipedia.org/wiki/Binary_relation

[3] "Constructive Mathematics"
Stanford Encyclopedia of Philosophy
https://plato.stanford.edu/entries/mathematics-constructive/

[4] "Homotopy Theory"
Wikipedia
https://en.wikipedia.org/wiki/Homotopy_theory

[5] "Currying"
Wikipedia
https://en.wikipedia.org/wiki/Currying

[6] "Computational complexity theory"
Wikipedia
https://en.wikipedia.org/wiki/Computational_complexity_theory

[7] "Homotopy Type Theory"
https://homotopytypetheory.org/

[8] "Dependent type"
Wikipedia
https://en.wikipedia.org/wiki/Dependent_type

[9] "Complex number"
Wikipedia
https://en.wikipedia.org/wiki/Complex_number

[10] "Commutative diagram"
Wikipedia
https://en.wikipedia.org/wiki/Commutative_diagram