Homotopy Sub-Types

by Sven Nilsen, 2017

A homotopy sub-type is used to reason about functions with a topological interpretation that corresponds to paths of atomic functions. For example:

$$q : ((a \rightarrow b)_c \rightarrow (c \rightarrow d)_c)_c$$

$$q : real \rightarrow real \rightarrow P$$
$$a, b, c, d : P$$

The sub-script `c` stands for `continuous`.

This means that given points `a, b, c, d` of type `P`, the family of all functions that give a surface parameterized by real numbers intersects these points at the “corners” of the surface by using unit intervals.

In path semantics it is common to reason about paths as functions connected by functions:

$$f[g_0 \rightarrow g_1] \Leftarrow\Rightarrow h$$

Assume that these are atomic functions, so one can write the following:

$$f(a) = b$$
$$h(c) = d$$
$$g_0(a) = c$$
$$g_1(b) = d$$

The non-trivial nature of paths arises from atomic functions, that are in some sense all equivalent. An atomic function takes some arguments of unique value and maps to a unique value. One can construct atomic paths for any atomic function, using atomic functions. Since this behavior is shared by all atomic functions, one can think of them as all expressing the same underlying idea. This nature gets non-trivial when introducing paths, which gives you normal functions and normal paths etc.

Instead of mapping between two discrete values, one can generalize the idea of these atomic functions to continuous transformations, meaning a function as following:

$$f : (a \rightarrow b)_c$$

$$f(0) = a \qquad f(1) = b$$
$$f : real \rightarrow P$$
$$\forall x : [0, 1] \{ \exists y \{ f(x) == y \} \}$$

The output does not need to be smoothly connected, but the function must be defined for every value between 0 and 1. In addition it must be “anchored” by `a` and `b` when the homotopy sub-type is defined.

One can construct a function that represents a "surface" between the contour curves defined by 4 continuous functions. A such surface is contractible to the edges and corners and therefore stores the information contained in these functions.

$q : \text{real} \to \text{real} \to P \wedge ((a \to b)_c \to (c \to d)_c)_c$

$\forall$ x : [0, 1] { q(0)(x) = f(x) }
$\forall$ x : [0, 1] { q(1)(x) = h(x) }
$\forall$ x : [0, 1] { q(x)(0) = g_0(x) }
$\forall$ x : [0, 1] { q(x)(1) = g_1(x) }

g(0, 0) = f(0) = a	g(0)(1) = f(1) = b
g(1, 0) = h(0) = c	g(1, 1) = h(1) = d
g(0, 0) = g_0(0) = a	g(1, 0) = g_0(1) = c
g(0, 1) = g_1(0) = b	g(1, 1) = g_1(1) = d

$f : \text{real} \to P \wedge (a \to b)_c$
$g_0 : \text{real} \to P \wedge (a \to c)_c$
$g_1 : \text{real} \to P \wedge (b \to d)_c$
$h : \text{real} \to P \wedge (c \to d)_c$

Homotopy sub-types makes it possible to talk about such surfaces and higher dimensional cubes without bothering about the implementation details of the curves and how they are combined.

A candidate for constructing a surface from contour functions is the `cquad` function:

```
cquad := \(ab : real → P, cd : real → P, ac : real → P, bd : real → P) =
        \(t0 : real) = \(t1 : real) = {
                w0 := 4 · (t0 – 0.5)^2
                w1 := 4 · (t1 – 0.5)^2
                a :=  ab(t1) + (cd(t1) - ab(t1)) · t0
                b :=  ac(t0) + (bd(t0) - ac(t0)) · t1
                if w0 == 1 {a}
                else if w1 == 1 {b}
                else if (w0 + w1) == 0 {0.5 · (a + b)}
                else {(a · w0 + b · w1) / (w0 + w1)}
        }
```

$\text{cquad} : ((a \to b)_c \to (c \to d)_c)_c$