Unit Interval Focus

by Sven Nilsen, 2018

In this paper I represent a function that maps the unit interval to itself, focused on an attractor value:

uif(a : prob, b : (>= 0), x : prob) = if x < a {a*(1-(1-x/a)^b)} else {(1-a)*((x-a)/(1-a))^b+a}

`a` controls the attractor value and `b` controls focus:

b = 0	Negatively focused
b = 1	Unfocused
b = ∞	Focused

This function has many desirable mathematical properties.

One such property is for all values of `a` when `b` is non-zero, the codomain covers the unit interval:

∀ a : prob, b : (> 0) { ∃uif(a, b) <=> prob }

Basically, it says that when it's not negatively focused, it always maps back to itself. This property is very nice for constructing global optimization algorithms that behave asymptotically similar to local optimization algorithms. This was the primary motivation behind the design of the `uif` function.

Here, `∃uif(a, b)` means the existential path of `uif(a, b)`. It returns `true` for all values that the function returns, which is the same as the sub-type of the codomain. It is called a "higher order existential path" because the expression contains an existential path quantified over some parameters.

Since the existential path is quantified over `a` that has the same sub-type as the input, one can use parameter elimination of higher order existential paths to write:

∀ b : (> 0) { ∃(uif b) <=> prob }

It turns out that when `b = 0`, the `uif` function has another desirable mathematical property:

∃(uif 0) <=> (= 0) ∨ (= 1)	It returns `0` or `1` when negatively focused

$(\exists_p(\text{uif } 0))(0) = a$	<=>	$0 : [\exists_p(\text{uif } 0)]\ a$
$(\exists_p(\text{uif } 0))(1) = 1 - a$	<=>	$1 : [\exists_p(\text{uif } 0)]\ (1 - a)$

When negatively focused and choosing some random input from the unit interval, the `uif` function returns `0` a fraction of the time corresponding to `a` and `1` all other times.

Another desirable mathematical property is that the derivative at `a` is the reciprocal of `b`:

∀ b : (> 0) { f'(a) = 1 / b }