Atomic Universal Existential Paths

by Sven Nilsen, 2017

For every function of finite types, there exists an associated function on sets. This function contains all information that is needed for the universal existential path of a function `f : A → B`:

∃f{} := \(g : A → bool) = \(x : B) = ∃ a : [g] true { f(a) == x }

One difference is that one normally uses a high level notation using functions to describe the sub-types, which is capable of describing infinite sets, such as:

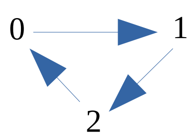
$\exists mul_{\mathbb{N}}$\{(> 1), (> 1)\} <=> composite

An atomic universal existential path encodes each value as a set directly and can compute the output set directly by applying the function to each value in the set and then take the union. This means the nice high level description of these sets disappear, and one loses the ability to describe infinite sets. On the other hand, an atomic universal existential path solves the problem of representing the universal existential path in code, but at low level only.

Consider the following function:

f := \(x : nat ∧ (< 3)) = (x + 1) % 3

Creating a space where each value is located and arrows tell how the function computes:



Represented as a table:

0 → 1
1 → 2
2 → 0

The atomic universal existential path can be written as a table on the power-set:

{} → {}
{0} → {1}
{1} → {2}
{2} → {0}
{0, 1} → {1, 2}
{1, 2} → {0, 2}
{0, 2} → {0, 1}
{0, 1, 2} → {0, 1, 2}