Constrained Existential Path Implies Normal Path

by Sven Nilsen, 2018

In this paper I show that the constrained existential path of a function implies the normal path. This means that the normal path can be thought of as a way to construct a partial constrained existential path. With other words, finding the normal path is not strictly necessary when having access to the constrained existential path, because the constrained existential path predicts the normal path.

Expressed in Nilsen-Cartesian product notation, for every normal path `$f[g_{i \to n}]$`, there exists a constrained existential path `$\exists f\{[g_i]\ x_i\}$` such that:

$$\exists f\{[g_i]\ x_i\} \Rightarrow [g_n]\ f[g_{i \to n}](x_i)$$

In expanded notation:

$$\exists f\{[g_0]\ x_0,\ [g_1]\ x_1,\ \ldots,\ [g_{n-1}]\ x_{n-1}\} \Rightarrow [g_n]\ f[g_0 \times g_1 \times \ldots \times g_{n-1} \to g_n](x_0,\ x_1,\ \ldots,\ x_{n-1})$$

For example:

add[even] <=> eq
∃add{[even] x, [even] y} <=> [even] (x == y)

mul[even] <=> or
∃mul{[even] x, [even] y} <=> [even] (x ∨ y)

concat[len] <=> add
∃concat{[len] x, [len] y} <=> [len] x + y

Here, the relation `<=>` holds because for every output with the sub-type on the right side, there exists some input with the sub-types on the left side.

For example, every even number can be constructed by adding two even numbers:

∃add{[even] true, [even] true} <=> [even] true
∃add{even, even} <=> even   Short version

Likewise, every odd number can be constructed by adding an even number with an odd number:

∃add{even, ¬even} <=> ¬even

This also means that the constrained existential path is defined by the normal path in these cases.

One can easily create a constrained existential path that only implies the normal path:

add{(> 0), (> 0)}[even] => eq 			Path set contains `eq`
∃add{(> 0) ∧ [even] x, (> 0) ∧ [even] y} => [even] (x == y)

The number `0` is even, but it can not be constructed by adding two even numbers greater than zero:

∃add{(> 0) ∧ even, (> 0) ∧ even} => even

To define the constrained existential path in this case, one must add `(¬= 0)` sub-type:

∃add{(> 0) ∧ even, (> 0) ∧ even} <=> (¬= 0) ∧ even

A more interesting example is Goldenbach's conjecture:

∃add{(> 2) ∧ prime, (> 2) ∧ prime} <=> (> 4) ∧ even
∃add{(= 2), (= 2)} <=> (= 4)

It is obvious that since every prime greater than 2 is odd, then the sum of any two such numbers is even. The number 4 is even and greater than 2, but that can not be written as a sum of two such numbers. Therefore, 4 is a special case.

It is known that:

∃add{(> 2) ∧ [even] x, (> 2) ∧ [even] x} <=> (> 4) ∧ even

Therefore, Goldenbach's conjecture implies that:

∃add{(> 2) ∧ prime, (> 2) ∧ prime} <=> ∃add{(> 2) ∧ [even] x, (> 2) ∧ [even] x}

However, this idea can not be expressed as a normal path, because:

- Goldenbach's conjecture only predicts for primes, not for non-primes
- The normal path does not imply the constrained existential path

One way to think about normal paths is as a language that is less powerful than the language of constrained existential paths. All the things that can be said in the language of normal paths can be said in the language of the constrained existential paths. So, the language of normal paths is a subset of the language of constrained existential paths.

However, the sub-type given by a normal path is a superset of the sub-type given by the constrained existential path. This is because the normal path is less specific, so it gives a larger sub-type.