Properties of non-constructive objects in path semantics

by Sven Nilsen, 2016

Normally it makes no sense to talk of properties, when the object in question can not exist. Yet, in path semantics it is fully possible, because there is no restriction to only work with constructive objects. A such object is the sum \`S\`, such that:

$s_0 = 1$
$s_1 = 2 \cdot s_0$
$s_2 = 2 \cdot s_1$
…
$s_n = 2 \cdot s_{n-1}$

$S = \sum i\ [0, \infty)\ \{ s_i \}$

Obviously there is no way to write down the whole number \`S\`, since it is infinitely large. Still, we recognize the sequence as all 1s in binary form. The number can be written from the least significant digit to the most significant, by filling out an infinite space with 1s.

S = …111111111 (binary)

We know everything there is to know about this number, even though the number itself can not be constructed. It is possible to describe properties of something that can not exist physically.

Notice that the least significant binary digit is 1, which means \`S\` is an odd number. What does it mean to have an odd non-existing number?

odd(number) → bool
: (0) → false
: (1) → true
: (x) → !odd(x-1)

Somehow, if we can repeat the procedure above an infinite times, it ends at the \`1\` case and returns \`true\`. Imagine that there is a magical computer sitting at our desktop, which can perform such impossible operations.

Or, we can just use path semantics.

We know that if we add two odd numbers, we get an even number. If we add one odd number with an even number, we get an odd number:

add(false: odd, false: odd) → odd { false }
add(false: odd, true: odd) → odd { true }
add(true: odd, false: odd) → odd { true }
add(true: odd, true: odd) → odd { false }

In general, this can be written as:

add(a: odd, b: odd) → odd { a != b }

Because `S` is computed as a sum, it uses `add` for each step. It starts with 1, then adds 2, then 4 etc. At each step we can determine that since we add an odd with an even number, we get an odd number. Even we repeat the same procedure an infinite times, we get the same answer. Therefore, from path semantics alone it can be determined that `S` will be an odd number, even though the number itself can not exist in the same way 4 or 5 exists.

Usually one thinks of isolated properties described through path semantics as connected to unknown objects. Yet, here we have a situation where the object is known to not exist, but still we *know for certain* that it has a specific property of oddness.

This raises the question: What does path semantics mean to us when the objects we talk about can not be constructed? Is not the whole point to look for objects and functions that have specific properties?

The answer is: No, path semantics does not necessarily deal with existing objects alone, but about ideas that can be reflected over in properties. Since `S` can be represented, it is just as real to relate to as any other number, given a powerful enough mind to process it.

For example, if Alice and Bob use path semantics to talk about `S` to each other, they will agree about everything about `S` if they talk about the very same object, whether the object exists or not. If they can not read each other's minds, there is no way to prove for them that they both think about the same object, other than trying to come up with ways to falsify the hypothesis "we are thinking about the same object". When there is no way for them to tell the difference, then it just means that the hypothesis is not false so far.

When Alice and Bob agrees, it does not matter whether what they agree about is nearby or not. Path semantics is not about describing reality around you, but about sharing logic. Of course it can be used to describe things, but it is not limited to any specific domain. All ideas are somewhere in the space constructed naturally by path semantics, even descriptions of Alice and Bob themselves.

This is nice… but… what if reality contain objects that are non-constructible? If Alice can think of a non-constructive object, it might be conceivable to think of reality as a person that communicate properties of it to us. When we are making observations of reality, we do not perceive the object, but information traveling from the object to our sensor systems. The word "information" is not the information that is processed. We are forever separated by the things we think about. What is thought can not be re-experienced, only remembered. Reflection has distance in space-time.

There is no rule that says that reality must be constructible in path semantics. Constructible objects means constructible things by path semantics. Even we can construct all information perceivable about reality in path semantics, does not mean that reality itself is constructible. Alice and Bob might agree about all knowledge about reality, without being able to create a simulated reality in their minds. Path semantics can predict what you do when observing a color, without experiencing the color. It is more like a shadow than a real body.

So, does it mean that `S` can exists? In order to know whether something exists, there must be some part of reality that connects us to it. The object `S` has no way to connect to anything else in itself. It can only be experienced like we do in a universe where somebody can reflect on its properties. It is impossible for it to exist in a finite universe. Since all computers are finite, we can not construct `S`. Path semantics can not tell us whether `S` exists, so we will *never know*, only agree what it is.