Incompleteness of Path Semantics

by Sven Nilsen, 2019

This is an informal article about why path semantics is incomplete and never can become complete.

Path semantics is incomplete: No amount of axioms or methods to produce axioms can exhaust the space of axioms that makes theorem proving possible or more efficient. This is why path semantics, instead of providing a complete list of axioms, starts with a core axiom. The purpose is to serve as a starting point for reasoning about mathematics, enough for “bootstrapping” into more powerful theorem techniques.

On the other hand, path semantics gives you lists of functions to use as a starting point[1].

One can think about this as writing some code for a programming language. There is no exhaustive list of functions one can define, because the programming language permits an infinite set of functions. As functions become more complex, there are more and more things one can say about them. So, the things you can say about functions grow even faster, the more you add more code.

Path semantics permits you to express a lot of things you can say about code. Therefore, there exists no level of abstraction that is the “final solution” to what you can express.

Forget about Gödel’s incompleteness theorems for a moment[2], and just think about the practical aspect of expressing mathematical ideas: Where does it end?

It does not end! So, why should you limit yourself to some language that “ends”?

You can check for correctness by using an automated theorem prover or a theorem prover assistant. A lot of smart people are working on building these programs. And: Most programs are free! However, personally I prefer to express my own ideas more clearly, instead of formalizing existing mathematics in code. Formalizing mathematics is not what I want to do with my time. I want to express my own ideas in a syntax that I can later translate into code, equations, or pictures.

While path semantics is designed for informal theorem proving, this does not mean that path semantics lacks rigor. There are lot of tools in path semantics which works, every time.

There are many ways of doing mathematics. You have many choices. There are many fields that are interesting. A lot to explore. New syntax you can experiment with.

Expressing ideas is the hardest problem I have. It is much harder than theorem proving. Once I have a clear definition of what I am thinking about, it becomes much easier to reason using mathematics.

All you can, and the best you can, is viewing mathematics from different perspectives. The “mathematical objects” that you are talking about in a highly abstract way are not actually seen as they are, in their full complexity. We use symbols or techniques like diagrams to help us understand what we are reasoning about. This corresponds to a language that “focuses on” different aspects of the same mathematical object.

References:

[1] “Reference material for path semantics”
AdvancedResearch, Sven Nilsen
https://github.com/advancedresearch/path_semantics/blob/master/sequences.md#reference-material

[2] “Gödel’s incompleteness theorems”
Wikipedia
https://en.wikipedia.org/wiki/G%C3%B6del%27s_incompleteness_theorems