About Publishing Path Semantics

by Sven Nilsen, 2019

If you are interested in helping to get parts of path semantics published, please contact me.

My email is: bvssvni@gmail.com

I am willing to co-author articles and papers about path semantics if somebody are interested.

If you are critic about the workflow of path semantics, please read the last section about "Workflow".

Where I would like to start publishing

It is easier to publish something if the idea is self-contained, so this is where I would like to start.

I written so much about path semantics that it is unreasonable to expect that all of it get published. Maybe once some major ideas are established and accepted, the rest might be published as a book.

One might think of this as a downside, but one can also think this as a luxury problem: There is a lot of material and one gets to pick out the piece that is mostly self-contained and introduces some new idea.

An obvious candidate is **Probabilistic Paths**, which might benefit from being published as an open problem to explain why it works and what the limits are. Currently, I know no person yet that has fully wrapped their head around this topic to a degree that they can confidently say that they understand it.

Probabilistic Paths might be a new contribution, or it might be already understood and used under another name. It is hard to find out, because there are few people working on this exact area. This problem is not made easier by the vast body of knowledge published about related topics in Probability Theory. I probably need assistance from somebody that are already familiar with the literature.

What is interesting, but I do not believe are new contributions

I believe that the fundamentals of path semantics is nothing new, just stringed together in a peculiar way. The axiom of path semantics is a restricted version of Leibniz' law to collections of symbols with some kind of order. This axiom has an element of vagueness, but the amount of vagueness is designed to permit just enough rigor to start the informal bootstrapping procedure into theories which are already established in mathematics.

I could be that I am wrong, though. The idea of combining rigor and vagueness in axioms could be new, and worth discussion. It is surely interesting, seen from the perspective of the debate between pragmatism and platonism in philosophy, for which positions are about this very subject.

From a logical point of view, Leibniz' law is already well studied. The introduction of vagueness into an axiom is not unnatural from an intuitive perspective, if one thinks about it as some unspecified informal procedure that eventually results in propositions with truth values. I believe this should be well understood in the philosophy of languages, if not, then I am interesting in publishing it.

Another issue is that people might find it interesting to look at all of mathematics through the lens of the axiom of path semantics, and this perspective might give other insights than e.g. viewing mathematics through the lens of Category Theory. However, one must be careful to distinguish between formal and informal arguments of this kind. Viewing mathematics in a particular way might be useful to build intuition about how it is used and how it works, but viewing mathematics as a formal process is another topic entirely. Formally, I suspect there is a lot of isomorphisms between theories.

Why I think the axiom of path semantics is interesting:

On one side, you have the rigorous framework of reasoning where symbols must be associated in a specific way, on the other side, you have an informal and vast space of creativity. This creates the balance that makes path semantics suitable both as a tool for formalizing things and viewing parts of mathematics through a somewhat humanly comprehensive perspective.

By following the procedure of informal bootstrapping, one ends up with established theories of mathematics. This claim is consistent with the belief that the axiom of path semantics “describes” mathematics somehow (I use quotations marks because the semantics is different from standard axiomatic theories). However, it is not necessary to test formalized concepts as new claims on their own, since the established techniques one ends up bootstrapping into, are already powerful as theorem proving tools. If one accepts Leibniz’ law from a logical point of view, then applying existing techniques are sufficient for soundness. Or, at least I believe so, and would love to be proven wrong.

Informal bootstrapping means that you end up using the same language as in existing techniques. From that point on, the claim of soundness is just as strong as the claim of existing techniques.

For this reason, is very hard to argue why the basic building blocks of path semantics requires publishing to be reviewed for correctness, since they are already widely used in mathematics. Equations and commuting squares/diagrams are not new. Some new syntax with an equivalent equational form does not introduce unsoundness, although it could be interesting to read about for somebody. My argument here is that this might be published because the audience finds it *interesting*, not because they expect to find errors, but this depends on the audience. If you think that the new syntax is important enough to be published, please contact me.

Path semantics does not build on Type Theory, but derives it through creative use of symbols. Along the way some new rules are introduced as definitions, e.g. type membership, which means that the axiom of path semantics only implies Type Theory formally given these new rules. Formalizing this process is very hard and most of the gain is to learn the informal bootstrapping procedure, so I do not expect this process to be formalized anytime soon, due to high costs and low gains.

I believe it should be enough to accept the informal bootstrapping procedure at face value, because if not, you run into a philosophical problem: What determines soundness if you do not have the ability to talk about it through any means, since the context of the language under bootstrapping does not include the very tools you use to talk about things? The theory can not prove its own consistency.

Path Semantics vs Category Theory

Path semantics defines a notation to be used on its own, but several people asked me how this notation fits with overlapping semantics, such as Category Theory.

A key to understand the difference, is that formal path semantics is more like programming.

Category Theory assumes that there exists some abstract collection of objects called “categories” with the central property of composition of morphisms, or arrows between objects, which are fitted onto various patterns in mathematics and the real world.

Concepts modeled in path semantics do not build on Category Theory, because you can not really talk about anything in path semantics without defining it to some degree first. However, you can define Category Theory in path semantics.

If you use Category Theory, in combination with Homotopy Type Theory, then it is possible to talk about path semantics. Category Theory is very useful in path semantics as a meta-language.

I do not use Category Theory as much as path semantics, because I like the “programming approach”:

There is no requirement that objects in a category have a type, making it difficulty to check some proof for consistency without being an expert. This is just my opinion from experience. I can model certain proofs of Category Theory using e.g. a generic monotonic solver, but I consider Type Theory to be a stronger foundation for computer checked proofs.

However, Category Theory works pretty well for doing mathematics and is invaluable to some people.

Personally I have not used Category Theory for anything specific, but I have used it as inspiration to create an algorithm that mines equations from data. I also think that dualities such as co-categories are useful insights, so I consider myself a person that benefits a lot from Category Theory, even if I find it difficult to contribute to it. Maybe this will change over time.

Category Theory is a mathematical language that often assume implicitly that you are dealing with abstract objects, while path semantics uses varying degrees of abstractness. In path semantics, this abstractness is often formalized in the language of functions.

Other people might use Category Theory in a different way and disagree with these view points. I am just trying to explain how I use those theories differently. There are many ways you can combine them that might make it easier to just consider the whole as a mathematical activity, instead of trying to come up with some clear-cut distinction between the two.

Workflow

The way I work when developing path semantics, is by writing down definitions, giving them names, and use them to construct or prove new ideas.

Usually, I do not consult literature during this creative process, because it slows down the workflow. I like to spend this time thinking, not reading. By using my own deep intuition of mathematics, I create a stream of ideas, which I then assemble and write down.

After I am finished with an idea and written a paper about it, I might check whether other people have worked on similar ideas. A lot of ideas have been investigated in some way or another. Sometimes there is a lot of overlap. Very often, there are various special cases that people have studied that I do not find interesting and distracts from the idea I was working on. Other times, there are related ideas that I find inspiring. Sometimes the same idea can be directly translated into another, which I call a "bridge". A bridge is more vague than isomorphism.

I do not much theorem proving myself by hand, except as a method of explaining things.
I prefer to use an automated theorem prover when checking for correctness is needed.
I do not follow mathematical journals or rely on a review process for correctness.

My biggest problem is to define ideas formally in an expressive way, which is why I use path semantics. I started working on path semantics because I wanted a better and more efficient way to understand what mathematics is about.

I do not work on mathematics to discover something that no one else have discovered, but in order to make progress myself in directions that I care about. This is sufficient for me, and if other people benefit from my method, then that is a bonus.

When I find that other people have been working on something similar, I assume that the theory is pretty well established unless there are major open problems. This is the common case.

If I was not connecting new ideas to my own framework, it would not be possible for other people to see how you get from one level of reasoning to another. I try to use established terminology when I am certain that it is the same thing, but if not, I am forced to invent a name on my own.

So, in a way my approach has a lot in common with programming language design. You start with some definitions and write down ideas, similar to comments in code, which are connected to other ideas. It is like assembling bits and pieces which becomes a larger "program".

Personally, I have little gain from publishing, because I do not learn that much this way. If I were to publish parts of path semantics, it would be to benefit other people, not myself. I do not expect to become famous, neither do I want to become famous. Since I can check things using automated theorem provers, I do not need a review process to check for correctness.

This does not mean I am against changes to the workflow. I am open for a combination of publishing, reviewing and the current workflow.

People are of course welcome to review and comment on papers in the path semantics repository. I published it in a free and open document format so people can suggest modifications through PRs.