Computability Space

Daniel Nikpayuk

June 14, 2015

Motivation:

Having a background in the landscape of mathematics and some of its history, I don't think this period in time of science and math in general is as much one for "new discovery" than it is for cleaning up and consolidation of existing theories.

In particular, my plan here is to take all the existing ideas of computing science and rework them into a single narrative. What I've learned about computing science is that computing like the humanities tries to mitigate the complexity of stories and narratives and paradigms within their respective literatures—which is why it's much more connected to creative writing than it is math.

Engineering is the originator of "mitigating complexity" as a way of thinking and is certainly an inspiration to what I'm trying to do here (I have most definitely been inspired by SICP). As a mathematician, I can honestly say that no existing branch of pure mathematics tries to mitigate complexity; each branch of math exposes and expounds a narrative, a worldview, a perspective, a single recursive way of interacting with the world.

As the humanities have been around for a long while now, they have the intuitive framework to explicate the human experience, and so I have borrowed the central ideas from them: Context, Semiotics, Media. I have reoriented this intuitive language as a formal structure I call a technology space. A way to think about it, is a technology space consists of a context space, a semiotic space, and a media space; each being independent of each other in a formal sense; each you can think of as a branch of mathematics—each has its own axiomatic worldview, but none by themselves mitigate complexity. It's when you take the three together, within their interplay do you get an emergent approximation of the complexity you're trying to model. You start with the axioms of each, and they use each other to build themselves further, allowing for a co-evolution of all three.

In particular, I aim to change the narratives of what currently exists in the landscape, recentering the philosophical idea of "a computation" (not to be taken lightly, there are many years of research by brilliant scholars about the idea) around the coevolution of three axiomatic spaces: Reference, Application, Evaluation. A reference space being the context space, an application space being the semiotic space and an evaluation space being the media space. Such a technology space would be called a computability space.

Keep in mind, this idea already exists under a different package: The Lambda Calculus, which takes as its three primitives: Abstraction, Application, and Reduction. Here, an abstraction space is the context space, an application space is again the semiotic space, and a reduction space is the media space. The Lambda Calculus is a formal branch of applied mathematics, but focuses more on self-contained formal proofs than it does on mitigating complexity and expressivity of computation.

In terms of computability spaces, one actually starts out philosophically, asking what it means to "reference". We can play with that idea and its connotations for a while and figure out what are the minimal number of assumptions needed (before the concept loses its emergent meaning) and those will be our axioms for a reference space. We will play a similar game to determine the simplest natures of our application and evaluation spaces. Finally, we can then start using these minimal spaces to help build upon each other to form more complicated versions of each, creating new paradigms and modular forms along the way which will inevitably lead us to our mitigating of the complexity of computability.

summary

To reiterate: A computability space is a technology space with the following bindings of undefined terms:

$$\text{context space} \quad := \quad \text{reference} \tag{1}$$

$$\text{semiotic space} \quad := \quad \text{application} \tag{2}$$

$$\text{media space} \quad := \quad \text{evaluation} \tag{3}$$

To intuitively validate this against the *lambda calculus*, our context is *abstraction*, our semiotics is *application*, while our media is *reduction.*

Computability

What does it mean to "compute" something? Intuitively we start with an *input* and result in an *output.* The part in-between is the hidden and mysterious computation. Let's suspend our disbelief momentarily, and assume we do know what it means to compute something, then we could start looking at the broader patterns. From this perspective, two main patterns would stand out:

1. Once we were comfortable with the existence of our computations, we would start to see them as *substitutions* and we would thus start to *compose* them. It would then become a matter of becoming comfortable with the practice of repeated substitutions until we reduced our input to a point where we had nothing left to substitute, in which case we would have our output.

We start by looking at systems of reference and in particular the word "reference" and its connotations, as it is the central context of computability.

Systems of Reference

Computabiity suffers from complexity, which is to say there is no single right way to *compute.* Instead, we will look at different known ways to mitigate the complexity of computability, starting as simple as possible; to be fair, before we can even look at *what* it means to reference, we require motivation as to *why* we need to reference in the first place: Substitution.

Any way you look at it, a computation takes an *input* and produces an *output.* In mathematics this is called a mapping, but from the worldview of a computability narrative, it is better to say this is a substitution. The most basic form of substitution is to use a *lookup table*, which is a system of memory, a system of reference which holds *bindings* of the input to the output. How we navigate this lookup table is another matter entirely, but it is an axiom that such a space exists as our initial and simplest system of reference. Also, for those lovers of rules, we will go into detail and construct complicated systems reference as we delve into the matter at hand—you will find our three initial axiomatic spaces of reference, application and evaluation to coevolve and infact help build each other up in our journey of mitigating the complexity of computability.

fin

Pijariiqpunga.