Type Theory or Model Theory?

Răzvan Diaconescu

June 24, 2019

1 Type Theory and Model Theory in Math and Computer Science

In the minds of many there is a deep antagonism between Type Theory (TT) and Model Theory (MT). While TT is virtually completely ignored by mainstream mathematics, MT has displayed some striking elegant applications in algebra and algebraic geometry and not only. In spite of these applications and of the fact that model theoretic methods have been established in mathematics, MT remains quite marginal within mainstream mathematics. This has to do with how the mathematical education is pursued worldwide both at the college and university levels, but also with the somehow unfair association of MT with mainstream logic, the latter being demised by mathematicians because of its formality and of its largely syntactic character.

The root of the demise of TT by mathematicians has to do with its very peculiar view on mathematics, expressed very clearly by Girard as follows: "*the semantics of proofs would express the very essence of mathematics*"[1]. It is difficult to find a view that is more divorced from reality than the one expressed above. Any *working* mathematician knows well two things:

1. the essence of mathematics is not proofs but discovery, the role of "proofs" being mainly to communicate,[2] and

2. a proof in mathematics is a social event, of a rather informal nature, rather than a type (which is a formal entity).

Moreover this heavily distorted view on mathematics is also connected with *constructivism*,[3] which is also strongly rejected by mathematicians.

On the other hand in informatics the situation is different. In the case of TT there is significant theoretical research in informatics departments in important universities worldwide and a number of tools have been developed (Isabelle, Coq, etc.).

[1] Taken from his book "Proof and Types".

[2] The case of Euclidean geometry represents a very clear example, solving a problem there most often reduces to just drawing appropriate figures that would reveal relevant properties.

[3] Its ideology being illustrated by "proofs as algorithms" slogan.

These developments are linked to the strong relationship between TT and the proof theory of logical systems. Moreover TT has a clear footprint in functional languages such as Haskell (although in the s/w industry their usage is negligible). In the case of MT informatics-driven research has led to the development of a new truly abstract approach to model theory. Also MT-based languages and tools have been developed (Maude, CASL, Hets, etc.), some of them being successfully used in a limited way in industrial applications.

The antagonism between TT and MT can be understood also philosophically, while TT represents an *absolutist* perspective on logic (anything in logic can be expressed in higher order logic), MT represents the *relativist* perspective. The latter can be understood both locally (there are multiple interpretations for a system of axioms) and globally through abstract model theory (there is a myriad of logical systems). Moreover while TT is a merely syntactic theory, MT is a semantic one.

The aim of the dialogues at the 9th Congress of the Romanian Mathematicians is neither to destroy the image of any of these two areas, nor to establish the harmony between them. The aim is just to put forward and analyse some of the arguments that have been circulated in the computing science for decades. A general conclusion is not to be expected (although each of us may have our own opinions, sometimes strong opinions).

2 Two great G's expressing antagonistic positions

Often radical positions pro and contra TT or MT have been expressed by promoters of one of these two areas. We give only the examples of two prominent scientists.

Girard considers that semantics (aka MT, the Tarskian trend in logic) is a fundamental grave intelectual error. He devoted much of his scientific efforts to "liberate" logic from the plague of semantics. Moreover others say that TT has unequalled verification power which has no correspondent in the MT approaches.[4]

On the other side of things, Goguen expressed often the opinion that while TT is interesting within the context of foundations of mathematics, it should be irrelevant for informatics. For Goguen the idea of informatics would be centered around the industrial level of s/w engineering. Two of Goguen's main arguments against TT are:

- the mere syntactic nature of TT may lead easily to inconsistencies,[5] and
- higher order programming is a bad, confusing, and unnecessary style of programming.

The foes of TT further argue that:

- It lacks proper specification capabilities (it cannot properly specify real systems), hence its verification power become useless in the absence of a specification level. This aspect is linked to TT not having a proper denotational

[4] This I found difficult to buy given the MT-based formal verification tools available.

[5] Goguen always emphasised on semantics, his favorite slogan being "Semantics first!"

semantics. Formal systems with only operational semantics are considered in the MT oriented community as subject to a grave methodological error, as denotational semantics is intimately linked to “understanding”.

- TT being divorced from mathematics and its most important tool, namely algebra, which are semantic domains, becomes tedious to work with.

3 The chimera of formality

In both TT and MT based formal methods there is a strong belief in the power of formality, which sometimes blurs the line between incompetence and utter insanity. In this respect it is difficult to find a better example than the following “vision” shared by many world wide established scientists in informatics (this time not from the “proofs-as-types” community) and articulated as follows:

> ”...we believe and hope that within a few years (e.g., a decade), there will be a cognitively-inspired software, which we can call now the Universal Theorem Prover (UTP), that takes a solvable mathematical conjecture C as input, and after an amount of time less than that required by a professional mathematician, it solves C, namely, UTP gives an understandable mathematical proof or it offers a clear formal counter example.”[6]

An apparently less radical view is that of the *proof assistants*, which are non-automatic provers that work on the basis of inputs from the user. Their relative succes in formal verification of computer systems led some to believe that at some point proof assistants will be used as a better medium for editing, maintaining and communicating mathematics. However such hopes are largely rejected by the community of mathematicians.

Such views seem to ignore several important aspects of mathematics. That mathematics is an informal subject, that at the core of doing mathematics lies the “seeing” mentioned above, the non-verbality of mathematical ideas and that æstetics is intimately connected to mathematical truth.[7]

So what do you think of all these?

[6] At page 237 in *Concept Invention – Foundations, Implementation, Social Aspects and Applications*, Springer 2018.

[7] An informed discussion on these issues may be found in Penrose’s best-seller *The emperor’s new mind*.