Thoughts on Martin-Löf's Meaning Explanations

Peter Dybjer

Chalmers tekniska högskola, Göteborg

to Peter Hancock
at his 60th birthday celebration
Glasgow, 19 December 2011

Martin-Löf on meaning explanations

Martin-Löf on meaning explanations

Some people feel that when I have presented my meaning explanations I have said nothing.

Martin-Löf on meaning explanations

Some people feel that when I have presented my meaning explanations I have said nothing.

And this is how it should be.

Martin-Löf on meaning explanations

Some people feel that when I have presented my meaning explanations I have said nothing.

And this is how it should be.

The standard semantics should be just that: standard. It should come as no surprise.

Martin-Löf on meaning explanations

> *Some people feel that when I have presented my meaning explanations I have said nothing.*
>
> *And this is how it should be.*
>
> *The standard semantics should be just that: standard. It should come as no surprise.*

Martin-Löf at the Types Summer School in Giens 2002 (as I recall it)

The meaning of hypothetical judgements (Martin-Löf 1979)

$$a \in A \ \ (x_1 \in A_1, \ldots, x_n \in A_n)$$

means that

The meaning of hypothetical judgements (Martin-Löf 1979)

$$a \in A \ \ (x_1 \in A_1, \ldots, x_n \in A_n)$$

means that

$$a(a_1, \ldots, a_n/x_1, \ldots, x_n) \in A(a_1, \ldots, a_n/x_1, \ldots, x_n)$$

provided

$$a_1 \in A_1,$$
$$\vdots$$
$$a_n \in A_n(a_1, \ldots, a_{n-1}/x_1, \ldots, x_{n-1}) \in A(a_1, \ldots, a_{n-1}),$$

and, moreover,

$$a(a_1, \ldots, a_n/x_1, \ldots, x_n) = a(b_1, \ldots, b_n/x_1, \ldots, x_n) \in A(a_1, \ldots, a_n/x_1, \ldots, x_n)$$

provided

$$a_1 = b_1 \in A_1,$$
$$\vdots$$

Program testing and Martin-Löf's meaning explanations

- Mathematical induction vs inductive inference.

Program testing and Martin-Löf's meaning explanations

- Mathematical induction vs inductive inference.
- Idealist vs realist; subjective vs objective; epistemic vs ontological? Constructive platonism??

Program testing and Martin-Löf's meaning explanations

- Mathematical induction vs inductive inference.
- Idealist vs realist; subjective vs objective; epistemic vs ontological? Constructive platonism??
- Testing, QuickCheck (Claessen and Hughes 2000), input generation. Test manual for Martin-Löf type theory. Cf Hayashi's proof animation for PX from the1980s.

Program testing and Martin-Löf's meaning explanations

- Mathematical induction vs inductive inference.
- Idealist vs realist; subjective vs objective; epistemic vs ontological? Constructive platonism??
- Testing, QuickCheck (Claessen and Hughes 2000), input generation. Test manual for Martin-Löf type theory. Cf Hayashi's proof animation for PX from the1980s.
- Meaning (testing) of hypothetical judgements, type equality, identity types?

Program testing and Martin-Löf's meaning explanations

- Mathematical induction vs inductive inference.
- Idealist vs realist; subjective vs objective; epistemic vs ontological? Constructive platonism??
- Testing, QuickCheck (Claessen and Hughes 2000), input generation. Test manual for Martin-Löf type theory. Cf Hayashi's proof animation for PX from the1980s.
- Meaning (testing) of hypothetical judgements, type equality, identity types?
- Meaning (testing) of functionals. Continuity, domains, games.

Program testing and Martin-Löf's meaning explanations

- Mathematical induction vs inductive inference.
- Idealist vs realist; subjective vs objective; epistemic vs ontological? Constructive platonism??
- Testing, QuickCheck (Claessen and Hughes 2000), input generation. Test manual for Martin-Löf type theory. Cf Hayashi's proof animation for PX from the1980s.
- Meaning (testing) of hypothetical judgements, type equality, identity types?
- Meaning (testing) of functionals. Continuity, domains, games.
- Meaning based on the evaluation of closed expressions or on the evaluation of open expressions? Weak head reduction?

Truth and proof: ontological or epistemic concepts?

Prawitz 2011:

> *That proof is an epistemic concept is of course normally not in doubt, whereas opinions differ concerning truth. Some hold that sentences are true in virtue of a reality given independently of us, while others hold that our linguistic expressions are about our experiences or possible experiences and that truth therefore should be understood in terms of what it is to experience or know something.*

Truth and proof: ontological or epistemic concepts?

Prawitz 2011:

> *According to the first standpoint, known as realism, truth may be called an ontological concept. The contrary standpoint, often labelled anti-realism, takes truth to be instead an epistemic notion. Since mathematical intuitionists explain the meaning of their sentences and what it is for them to be true in terms of what counts as proofs of them, intuitionism is commonly seen as a clear-cut example of an anti-realistic view.*

Truth and proof: ontological or epistemic concepts?

Prawitz 2011:

> *But not all intuitionists agree with that view. It appears from what Per Martin-Löf has written in the 90's and from what he said at the conference at which the contributions to this volume were presented that he does not. Martin-Löf explains the meaning of propositions in terms of proofs, and defines the truth of a proposition as the existence of a proof. Nevertheless, he takes truth to be an ontological concept, not explained in terms of any epistemic notions. If you ask how this is possible, the answer is that he takes even proof to be a non-epistemic concept.*

Truth and proof: ontological or epistemic concepts?

Prawitz 2011:

> *More precisely, Martin-Löf makes a distinction between two senses of proofs.*

ontological: "proof object" a in $a \in A$

epistemic: "demonstration"

$$\frac{\vdots}{\Gamma \vdash a \in A}$$

a tree of inferences where each inference is an instance of an inference rule of the theory

Truth and proof: ontological or epistemic concepts?

ontological: "proof object" a in $a \in A$. Judgements

$$\Gamma \vdash a \in A$$

are valid provided they pass all "tests"; tests are interactive as in games. Player computes output and type-checks. Opponent generates input. Meaning explanations provide a test manual.

epistemic: "demonstration"

$$\frac{\vdots}{\Gamma \vdash a \in A}$$

a tree of inferences where each inference is an instance of an inference rule of the theory. The system is sound if all demonstrations end in judgements which pass all tests following the test manual.

Truth and proof: ontological or epistemic concepts?

Prawitz 2011:

> *Martin-Löf still adheres to intuitionism or constructivism; the law of excluded middle does not come out as true according to his philosophy. But the traditional connection between intuitionism and an anti-realistic or epistemic understanding of meaning and truth has been abandoned. His new position is in this way exhibiting a quite original and interesting combination of ideas.*

Dummett's neo-verificationism

Prawitz 2011:

> *Since in mathematics proofs are what count as grounds for assertions, Dummett finds the intuitionistic explanation of the logical constants in terms of proofs to provide a prototype for a theory of meaning built on this idea. To generalize this to ordinary language, he suggests that we speak instead of verifications. Some statements in natural language are verified by making certain observations, while others require both observation and inference. In mathematics verification is by inference alone, which is thus a limiting case, opposite to that of observational parlance.*

Is mathematics an experimental science?

Miquel 2010: The experimental effectiveness of mathematical proof:

> *We can thus argue (against Popper) that mathematics fulfills the demarcation criterion that makes mathematics an empirical science. The only specificity of mathematics is that the universal empirical hypothesis underlying mathematics is (almost) never stated explicitly.*

How to test categorical judgements $a \in A$?

Compute the canonical form of A and a!

- If $A \Rightarrow N$, then
  - if $a \Rightarrow 0$, then the test is successful.
  - if $a \Rightarrow s(b)$, then test whether $b \in N$.
  - if $a \Rightarrow c(b_1, \ldots, b_n)$ for some other constructor c, then the test fails.
- If $A \Rightarrow \Pi(B, C)$, then
  - if $a \Rightarrow \lambda(c)$, then test $y \in B \vdash c(y) \in C(y)$
  - if $a \Rightarrow c(b_1, \ldots, b_n)$ for some other constructor c, then the test fails.
- If $A \Rightarrow U$
  - if $a \Rightarrow N$, then the test is successful.
  - if $a \Rightarrow \Pi(b, c)$, then test whether $b \in U$ and $y \in b \vdash c(y) \in U$.

    $\vdots$
  - if $a \Rightarrow c(b_1, \ldots, b_n)$ for some c which is not a constructor for small sets, then the test fails.

Testing hypothetical judgements - how to generate input?

To test

$$y \in B \vdash c(y) \in C(y)$$

where $B \Rightarrow N$, we generate a canonical natural number. We can do this either

strictly: generate $y := s^n(0)$ and then test

$$c(s^n(0)) \in C(s^n(0))$$

lazily: try to test

$$y \in B \vdash c(y) \in C(y)$$

by computing the *open* expressions $c(y)$ and $C(y)$ and see whether their canonical forms match, and only if the canonical forms of $c(y)$ or $C(y)$ are neutral (outermost form not a constructor), then generate y.

Testing hypothetical judgements - lazy generation of input

Try to test

$$y \in B \vdash c(y) \in C(y)$$

without knowing $y \in N$. If the computation is blocked by reaching a neutral term, lazily generate either

- $y := 0$ and test
$$c(0) \in C(0)$$
- $y := s(y_1)$, and try to test
$$y_1 \in N \vdash c(s(y_1)) \in C(s(y_1))$$

Computation of open vs closed expressions

- Martin-Löf's meaning explanations are based on the lazy evaluation of closed expressions to lazy canonical form, the lazy method is based on lazy evaluation of open expressions
- The computation of open expressions for the purpose of testing is reminiscent of the computation of open expressions in the type-checking algorithm for intensional type theory. The difference is what you do when you hit a variable (neutral term). Lookup of type vs generation of input of that type.

How to generate functional input?

- Easy to generate input of type N (or other algebraic data types). Cf the idea that "scientifically applicable mathematics" is restricted to Π_1 sentences.
- But Martin-Löf type theory gives meaning to sentences with hypotheses of arbitrary type.
- To test

  $$y \in B \vdash c(y) \in C(y)$$

  where $B \Rightarrow N \rightarrow N$. The strict method does not work, because we do not know how to generate an arbitrary $y \in N \rightarrow N$.

Meaning of hypothetical judgements revisited

How do we understand:

$$a(a_1, \ldots, a_n/x_1, \ldots, x_n) \in A(a_1, \ldots, a_n/x_1, \ldots, x_n)$$

provided

$$a_1 \in A_1,$$
$$\vdots$$

?

Meaning of hypothetical judgements revisited

How do we understand:

$$a(a_1, \ldots, a_n/x_1, \ldots, x_n) \in A(a_1, \ldots, a_n/x_1, \ldots, x_n)$$

provided

$$a_1 \in A_1,$$
$$\vdots$$

?

- classical platonism?

Meaning of hypothetical judgements revisited

How do we understand:

$$a(a_1, \ldots, a_n/x_1, \ldots, x_n) \in A(a_1, \ldots, a_n/x_1, \ldots, x_n)$$

provided

$$a_1 \in A_1,$$
$$\vdots$$

?

- classical platonism?
- constructive subjectivism?

Meaning of hypothetical judgements revisited

How do we understand:

$$a(a_1, \ldots, a_n/x_1, \ldots, x_n) \in A(a_1, \ldots, a_n/x_1, \ldots, x_n)$$

provided

$$a_1 \in A_1,$$
$$\vdots$$

?

- classical platonism?
- constructive subjectivism?
- constructive platonism?

Testing functionals: an example

How to test

$$f \in N \to N \vdash \mathit{if}\ f(0) = 0\ \mathit{then}\ f(1)\ \mathit{else}\ f(2) \in \mathit{Fin}(s(f(1) + f(2)))$$

where

$$\begin{aligned} \mathit{Fin}(0) &= \emptyset \\ \mathit{Fin}(s(n)) &= 1 + \mathit{Fin}(n) \end{aligned}$$

$$\begin{aligned} 0 + n &= n \\ s(m) + n &= s(m + n) \end{aligned}$$

Assume suitable encoding of numbers and

$$\emptyset = \mathit{Fin}(0) \subseteq \mathit{Fin}(1) \subseteq \mathit{Fin}(2) \subseteq \cdots \subseteq N$$

Game-theoretic testing

Test $f : N \to N \vdash \ \textit{if } f(0) = 0 \textit{ then } f(1) \textit{ else } f(2) \in \mathit{Fin}(s(f(1) + f(2))!$

- Evaluate both term and type to whnf:

  $$f : N \to N \vdash \ \textit{if } f(0) = 0 \textit{ then } f(1) \textit{ else } f(2) \in 1 + \mathit{Fin}(f(1) + f(2))$$

  Check whether canonical form of term matches canonical form of type. No, term is neutral. We need the value of $f(0)$.
- Generate $f(0) := 0$ – opponent move.
- Evaluate term:

  $$f : N \to N \vdash \ f(1) \in 1 + \mathit{Fin}(f(1) + f(2))$$

  No match yet. Term is still neutral. We need the value of $f(1)$.
- Generate $f(1) := 0$ – opponent move.
- Evaluate term:

  $$f : N \to N \vdash \ 0 \in 1 + \mathit{Fin}(f(1) + f(2))$$

  Test has succeeded!

Identity types. How to generate $x \in I(A, a, b)$?

To test

$$y \in B \vdash c(y) \in C(y)$$

where $B \Rightarrow I(A, a, b)$? It is clear how to generate input $y := r$,

- if $A \Rightarrow N$, or another algebraic data type, because then we can decide $a = b \in N$.
- But what if $A \Rightarrow N \to N$?

Remove identity types (as primitives),

We are back to Martin-Löf 1972 where there is no primitive identity type, but where identity types can be defined using other primitives:

- Define $I(N, a, b)$ by primitive recursion. The rule

$$\frac{\Gamma \vdash c \in I(N, a, b)}{\Gamma \vdash a = b \in N}$$

  is valid in the testing interpretation (but was not part of the 1972 theory)

- Define $I(N \to N, f, g) = \Pi x \in N.I(N, f(x), g(x))$. The rule

$$\frac{\Gamma \vdash c \in I(N \to N, f, g)}{\Gamma \vdash f = g \in N \to N}$$

  is valid in the testing interpretation (but was not part of the 1972 theory)

- Etc

Martin-Löf on meaning explanations

> *Some people feel that when I have presented my meaning explanations I have said nothing.*
>
> *And this is how it should be.*
>
> *The standard semantics should be just that: standard. It should come as no surprise.*

This is the case for "Π_1-types", but the meaning of other types gets us into games and interactive programming with dependent types.

Happy Birthday, Peter!