MTT-semantics in Martin-Löf's Type Theory with HoTT's Logic*

Zhaohui Luo**

Dept of Computer Science, Royal Holloway, Univ of London
zhaohui.luo@hotmail.co.uk

Formal semantics in modern type theories (MTT-semantics for short) [7, 1] is a semantic framework for natural language, in the tradition of Montague's semantics [10]. While Montague's semantics is based on Church's simple type theory [2, 4] (and its models in set theory), MTT-semantics is based on dependent type theories, which we call modern type theories (MTTs), to distinguish them from the simple type theory.

Usually, we say that MTTs include predicative type theories such as Martin-Löf's (intensional) type theory (MLTT) [11] and impredicative type theories such as UTT [5] and pCIC [3]. However, so far, we have mainly developed MTT-semantics in the impredicative type theory UTT in which there is a totality *Prop* of all logical propositions. In contrast, Martin-Löf's MLTT, as employed in the work by Sundholm [13], Ranta [12] and others, is predicative and in it there is no such a type of all propositions. In fact, Martin-Löf has identified types with propositions [8, 9] and this gives rise to a logic based on the principle of propositions as types – the usual logic in MLTT – let's call it the PaT logic.

Unfortunately, unlike UTT, MLTT with PaT logic is *inadequate* to be used for MTT-semantics (this has been pointed out and discussed in [6]). This paper, besides describing the problem briefly, proposes the idea that MLTT, when extended with the h-logic developed in the HoTT project [14], can be used adequately as a foundational language for MTT-semantics.[1] This also justifies the inclusion of MLTT as one of the MTTs for MTT-semantics, as we have always done in previous writings.[2]

---

* This short paper accompanies the author's invited talk at LACompLing18: it gives a concise description of a part of the talk that describes unpublished work.

** Partially supported by EU COST Action CA15123 and CAS/SAFEA International Partnership Program.

[1] I should emphasise that further study is needed to demonstrate whether MLTT extended with HoTT's logic can adequately deal with all the semantic matters as studied based on UTT, although intuitively I do not see any serious problems. To mention a potential issue: in a predicative type theory, formally there is no totality of all propositions (and hence no totality of predicates) – one can only have relative totalities of propositions or predicates using predicative universes (cf., PROP in §2). This is not ideal but it is to be seen whether it causes any serious problems.

[2] Although the current work has not been published, its idea, i.e., using HoTT's logic instead of the PaT logic, has been in the author's mind for a long time. This has

1 Background: Problem and Proposal

As I explained in [6], Martin-Löf's type theory with PaT logic is inadequate for MTT-semantics. The reason is that, in order to employ types to represent collections such as those for CNs, some principle of proof irrelevance is needed and such a principle is incompatible with the PaT logic where types and propositions are identified.

For example, one may use Σ-types to represent CNs modified by intersective adjectives [12]: *handsome man* can be interpreted as $\Sigma(Man, handsome)$ where Man is a type and $handsome : Man \to U$ with U being a predicative universe. Then, one can ask: what is the identity criterion for *handsome man*? An obvious answer should be that it is the same as that for *man*: two handsome men are the same if, and only if, they are the same man. This implies that, for any man m, any two proofs of $handsome(m)$ should be the same – proof irrelevance comes into play here.

A principle of proof irrelevance stipulates that any two proofs of the same logical proposition be the same. However, in order to state this principle, there must be a clear distinction between logical propositions and other types so that proof irrelevance can be imposed for the former (and not for the latter). In Martin-Löf's type theory with PaT logic, however, propositions and types are identified and, therefore, proof irrelevance would have implied the collapse of all types into singleton or empty types: this is obviously absurd and unacceptable. In contrast, in an impredicative type theory such as UTT, the distinction between propositions and types is clear – one has a type $Prop$ of propositions and, therefore, a principle of proof irrelevance can be stated and imposed in a straightforward way. For instance, proof irrelevance for computational equality can be imposed in UTT by means of the following rule [6]:

$$\frac{\Gamma \vdash P : Prop \quad \Gamma \vdash p : P \quad \Gamma \vdash q : P}{\Gamma \vdash p = q : P}$$

But, such rules would not be possible for MLTT with PaT logic.

Recently, based on Martin-Löf's type theory, researchers have developed Homotopy Type Theory (HoTT) [14] for formalisation of mathematics. One of the developments in the HoTT project is its logic (sometimes called h-logic) based on the idea that a logical proposition is a type that is either a singleton or empty. This, among other things, has given rise to a logic with a type of all (small) propositions. Our proposal is to use MLTT with HoTT's logic (or, more precisely, MLTT extended with h-logic) for MTT-semantics – let's call this type theory MLTT_h. We believe that, like UTT, MLTT_h serves as an adequate foundational semantic language as well.

partly contributed to the decision of including MLTT as one of the MTTs for MTT-semantics.

2 Martin-Löf's Type Theory with H-logic and Its Use for MTT-Semantics

We describe MLTT$_h$, MLTT with HoTT's logic, sometimes called h-logic. We shall assume the knowledge of MLTT (see Part III of [11] for its formal description) and describe, albeit concisely, the h-logic developed in the HoTT project [14].

Remark 1. MLTT$_h$ only extends MLTT with the h-logic. It does not include the other extensions of MLTT in the HoTT project: in particular, we do not use the univalence axiom or any other higher inductive types except those in h-logic.

2.1 H-logic

In HoTT, a proposition is a type whose objects are all propositionally equal to each other. Formally, let U be the smallest universe in MLTT and $A : U$. Then A is a proposition in h-logic if the following is true/inhabited:

$$\mathsf{isProp}(A) = \Pi x, y{:}A.\ Id_A(x, y),$$

where Id is the propositional equality (called Id-type) in MLTT. We can then define the type of propositions in U to be the following Σ-type:

$$Prop_U = \Sigma X{:}U.\ \mathsf{isProp}(X).$$

In the following, we shall omit U and write PROP for $Prop_U$. Note that PROP is different from *Prop* in an impredicative type theory like UTT, which is impredicative and contains all logical propositions. PROP does not – it only contains the propositions in the predicative universe U; sometimes, we say that PROP is the type of *small* propositions. Another thing to note is that an object of PROP is not just a proposition – it is a pair (A, p) such that A is a proposition in U and p is a proof of $\mathsf{isProp}(A)$.

The traditional logical operators can be defined and some of these definitions (e.g., disjunction and existential quantifier) use the following truncation operation that turns a type into a proposition.

– *Propositional Truncation.* Let A be a type. Then, there is a higher inductive type $\|A\|$ with the following rules:

$$\frac{\Gamma \vdash a : A}{\Gamma \vdash |a| : \|A\|} \qquad \frac{\Gamma\ valid}{\Gamma \vdash \mathsf{isProp}(\|A\|)\ true} \qquad \frac{\Gamma \vdash \mathsf{isProp}(B) \quad \Gamma \vdash f : A \to B}{\Gamma \vdash \kappa_A(f) : \|A\| \to B}$$

such that the elimination operator κ_A satisfies the definitional equality $\kappa_A(f, |a|) = f(a)$.

Note that $\|A\|$ is a higher inductive type and, in particular, in turning a non-propositional type A into a proposition $\|A\|$, one imposes that there is a proof of $\mathsf{isProp}(\|A\|)$, i.e., $\|A\|$ is a proposition – in other words, every two proofs of $\|A\|$ are equal (propositionally).[3]

The traditional logical operators can be defined as follows.

- $true = \mathbf{1}$ (the unit type) and $false = \emptyset$ (the empty type).
- $P \wedge Q = P \times Q$, $P \supset Q = P \to Q$, $\neg P = P \to \emptyset$ and $\forall x{:}A.P(x) = \Pi x{:}A.P(x)$.
- $P \vee Q = \|P + Q\|$ and $\exists x{:}A.P(x) = \|\Sigma x{:}A.P(x)\|$.

2.2 MTT-semantics in MLTT$_h$

MTT-semantics can be done in MLTT$_h$,[4] including the following examples.

Predicates. We can approximate the notion of predicate by means of the relative totality Prop of small propositions – i.e., a predicate over type A is a function of type $A \to$ Prop. Therefore, we can interpret linguistic entities such as verb phrases, modifications by intersective adjectives, etc. as we have done before based on UTT.

Proof Irrelevance. In h-logic as described above, every two proofs of a proposition in Prop are equal (by definition, for the propositional equality Id) and, in particular, this is imposed for $\|A\|$ when a non-propositional type A is turned into a proposition $\|A\|$. Therefore, the problem described in §1 is resolved satisfactorily in MLTT$_h$.

References

1. Chatzikyriakidis, S., Luo, Z.: Formal Semantics in Modern Type Theories. Wiley & ISTE Science Publishing Ltd. (2018), (to appear)
2. Church, A.: A formulation of the simple theory of types. J. Symbolic Logic 5(1) (1940)
3. The Coq Development Team: The Coq Proof Assistant Reference Manual (Version 8.3), INRIA (2010)
4. Gallin, D.: Intensional and higher-order modal logic: with applications to Montague semantics (1975)
5. Luo, Z.: Computation and Reasoning: A Type Theory for Computer Science. Oxford University Press (1994)
6. Luo, Z.: Common nouns as types. In: Bechet, D., Dikovsky, A. (eds.) Logical Aspects of Computational Linguistics (LACL'2012). LNCS 7351 (2012)
7. Luo, Z.: Formal semantics in modern type theories with coercive subtyping. Linguistics and Philosophy 35(6), 491–513 (2012)
8. Martin-Löf, P.: An intuitionistic theory of types: predicative part. In: H.Rose, J.C.Shepherdson (eds.) Logic Colloquium'73 (1975)

[3] For people who are familiar with type theory, this implies that canonicity fails to hold for the resulting type theory.

[4] See Footnote 1.

9. Martin-Löf, P.: Intuitionistic Type Theory. Bibliopolis (1984)
10. Montague, R.: Formal Philosophy. Yale University Press (1974), collected papers edited by R. Thomason
11. Nordström, B., Petersson, K., Smith, J.: Programming in Martin-Löf's Type Theory: An Introduction. Oxford University Press (1990)
12. Ranta, A.: Type-Theoretical Grammar. Oxford University Press (1994)
13. Sundholm, G.: Constructive generalized quantifiers. Synthese 79(1), 1–12 (1989)
14. The Univalent Foundations Program: Homotopy type theory: Univalent foundations of mathematics. Tech. rep., Institute for Advanced Study (2013)