Modern Type Theories for Natural Language Semantics

(Introductory Course in Language and Logic)

Zhaohui Luo*
Royal Holloway
University of London

July 6, 2016

Abstract

Modern Type Theories (MTTs) provide us with a new framework for formal semantics with attractive advantages as compared to Montague Grammar. First, MTTs have rich type structures that can be employed effectively to capture various linguistic features that have proved difficult in the Montagovian setting. Second, MTTs are proof-theoretically specified and can hence be usefully implemented in proof assistants such as Coq, where the MTT-semantics has been implemented for computer-assisted reasoning. These two respects may be characterised as saying that the MTT-semantics is both model-theoretic and proof-theoretic. They offer unique features unavailable in traditional logical systems that have proved very useful in formal semantics.

We shall introduce MTTs and how they can be used for formal semantics. The lectures will be informal and explanatory. They will be rigorous but contain a lot of examples, to illustrate the use of MTTs, on the one hand, and to compare the MTT-semantics with Montague Grammar, on the other. They should be suitable for any semantics-oriented linguists and the others who can gain from learning a new framework of formal semantics.

*Email: zhaohui.luo@hotmail.co.uk; home page: http://www.cs.rhul.ac.uk/home/zhaohui/; tel: +44-7962-100437; fax: +44-1784-439786.

1 Motivation and Description

Modern Type Theories (MTTs) are computational logical systems that include Martin-Löf's type theory and the type theory implemented in the Coq proof assistant.[1] Based on the proof technology associated with them, MTTs have useful applications in several areas including foundations/formalisation of mathematics, software verification in computer science, and formal semantics of natural languages.

Formal semantics based on MTTs (MTT-semantics for short) was first studied by Ranta [23] and has recently been intensively studied[2] and become an attractive alternative to Montague semantics. Studying MTT-semantics is in the direction of semantic research on rich type structures (see, for example, [2, 10, 24], among others.) First, MTTs have rich type structures that can be employed effectively to capture a wide range of linguistic features, some of which have proved difficult to be described adequately in the Montagovian setting.[3] Second, MTTs are proof-theoretically specified and can hence be usefully implemented in proof assistants such as Coq, where the MTT-semantics has been implemented for computer-assisted reasoning [8]. These two respects may be characterised as saying that the MTT-semantics is both model-theoretic and proof-theoretic [17] and this makes MTTs become a promising and attractive semantic framework. Therefore, MTT-semantics is rather unique: it was not available in traditional logical settings and offers new perspectives in NL semantics.

However, learning MTTs can be laborious. This is partly because MTTs incorporate several new concepts which even traditional logicians may find difficult to apprehend at the beginning, let alone the linguists who apply logics to formal semantics. This is one of the problems that this series of lectures aim to tackle. We shall introduce the basic logical mechanisms in MTTs, including judgements, signatures/contexts, inductive types, coercive

[1]More precisely, examples of MTTs include Martin-Löf's predicative type theory [21, 22] and the type theories pCIC [11] and UTT [12], as implemented in the proof assistants Agda [1], Coq [11] and Lego/Plastic [19, 4], respectively.

[2]The proposer and colleagues have in the recent years (since 2009 until now) worked on this, with publications on various venues including L&P, JoLLI, I&C, LACL, SALT, ESSLLI, S&B, MoL, TYPES and LENLS.

[3]An example of such difficulties, among several others, is the linguistic feature of copredication, which is a typical to show that a treatment with subtyping in the Montagovian setting does not work, while in MTT-semantics where CNs are interpreted as types, work as we expect. See, for example. [13, 16] for more details on this.

subtyping mechanisms, dependent types such as Π-types and Σ-types, type universes and dot-types. Although our explanations are based on solid theoretical basis (see, for example, [21, 12]), we shall use examples in linguistic semantics to explain these type constructors in an introductory way.

The other focus of the lectures will be to introduce how MTTs can be used for formal semantics. We shall explain with examples how various semantic issues are dealt with in the MTT-semantics, including the topics like CNs as types [15], adjectival and adverbial modifications [7], coordination [6], copredication [13, 5], linguistic coercions [3], among others. For each of the above, we shall also compare with its treatment(s) in Montague Grammar (if it can be dealt with in MG) and discuss the advantages and pitfalls along the way. It is hoped that, by means of this, we can also show why using MTTs may be advantageous as compared with the simple type theory (as in the Montagovian setting) or traditional logical systems.

We shall explain in what sense the MTT-semantics is both model-theoretic and proof-theoretic [17] and why these two respects allow MTTs to have wide semantic coverage of linguistic features, on the one hand, and to support practical reasoning based on the existing proof technology provided by MTT-based proof assistants [8], on the other. Finally, some advanced research topics will also be discussed, albeit briefly, including for example a discussion of a Lambek calculus with dependent types [18] which forms a basis to establish a uniform basis for NL analysis: from automated syntactical analysis based on a Lambek categorial grammar to logical reasoning in proof assistants based on MTT-semantics.

In summary, this course will focus on informal explanations of MTTs and the MTT-semantics with linguistic examples. It is our hope that the participants with traditional logical backgrounds can benefit from such informal introductions.

2 Course Outline

Tentatively, the course consists of the following lectures:

- *Monday*: Introduction to formal semantics, in the Montagovian style. We shall introduce Montague's semantics and informally mention some of the treatments in MTT-semantics, so that the students/audience will get some intuitive ideas about MTTs and MTT-semantics. At the same time, we shall introduce some examples to be used in the following lectures.

- *Tuesday*: Introduction to the basics of MTTs. The basic concepts in MTTs will be introduced, and their meanings and uses explained by means of examples in linguistic semantics. Some of these basic concepts are not familiar to traditional logicians or semanticists (e.g., judgements, signatures/contexts, inductive types, dependent types, type universes) and will be explained in more details.

- *Wednesday*: Introduction to the MTT-semantics. Based on Monday's lecture, we shall systematically introduce how MTTs are used to give formal semantics. These include interpretations of many basic linguistic features (e.g., modifications and coordination) and some more advanced ones such as co-predication, among others. Comparing with other semantic frameworks such as Montague Grammar, we shall explain the MTT-semantics more thoroughly and discuss its advantages and potential issues for further studies.

- *Thursday*: This lecture explains the idea that MTT-semantics is both model-theoretic and proof-theoretic. It will first introduce traditional notions of model-theoretic semantics (in both logics and NLs) and proof-theoretic semantics (in logic and philosophy) and, then, explains why the MTT-semantics can be seen as both and, more importantly, why this has substantial theoretical and practical implications that lead to wide coverage of linguistic features, on the one hand, and practical reasoning supported by the current proof technology, on the other.

- *Friday*: Computer-assisted reasoning based on MTT-semantics and advanced research issues. This lecture explains how the proof assistants like Coq can be used to implement MTT-semantics and to conduct computer-assisted reasoning in NLs accordingly. Examples are considered and discussed on the development of a reasoning engine based on MTT-semantics and the syntactic theory based on Lambek calculus (and its extensions with dependent types, as studied in [18, 20]).

Course material, including lecture notes, lecture slides and related papers, will be made available to the students.

3 Expected Level and Prerequisites

The proposed is an introductory course in the area of Language and Logic. A preliminary background in logic and the basic knowledge about Montague Grammar are useful.

4 Other Information

4.1 Proposer

The proposed lecturer, Zhaohui Luo, is full professor in Department of Computer Science, Royal Holloway, University of London. He is an expert in modern type theories [12] and the associated proof assistants [19, 4]. Luo is a member of the steering committee of the TYPES consortium which has a successful conference series on modern type theories and their applications. Luo has worked and published on MTT-semantics in the last several years [16, 13, 14, 15, 3, 8, 9] – see following URL for the web page of a research project on MTT-semantics for more references:

`http://www.cs.rhul.ac.uk/home/zhaohui/lexsem.html`.

4.2 Previous ESSLLI Courses

The proposer was a lecturer at ESSLLI 2011 in Ljubljana (joint course with Prof N. Asher) and at ESSLLI 2014 in Tubingen (joint course with Dr. S. Chatzikyriakidis).

References

[1] Agda proof assistant. `http://appserv.cs.chalmers.se/users/ulfn/wiki/agda.php`, 2008.

[2] N. Asher. *Lexical Meaning in Context: a Web of Words.* Cambridge University Press, 2012.

[3] N. Asher and Z. Luo. Formalisation of coercions in lexical semantics. *Sinn und Bedeutung 17, Paris*, 2012.

[4] P. Callaghan and Z. Luo. An implementation of LF with coercive subtyping and universes. *Journal of Automated Reasoning*, 27(1):3–27, 2001.

[5] S. Chatzikyriakidis and Z. Luo. Individuation criteria, dot-types and copredication: A view from modern type theories. *Proc of the 14th Inter. Conf. on Mathematics of Language, Chicago.*

[6] S. Chatzikyriakidis and Z. Luo. An account of natural language coordination in type theory with coercive subtyping. *Constraint Solving and Language Processing 2012, LNCS 8114*, 2013.

[7] S. Chatzikyriakidis and Z. Luo. Adjectives in a modern type-theoretical setting. *The 18th Conf. on Formal Grammar, Dusseldorf. LNCS 8036*, 2013.

[8] S. Chatzikyriakidis and Z. Luo. Natural language reasoning in Coq. *J. of Logic, Language and Information*, 23(4), 2014.

[9] S. Chatzikyriakidis and Z. Luo, editors. *Modern Perspectives in Type Theoretical Semantics.* Studies of Linguistics and Philosophy, Springer, 2015. (To appear).

[10] R. Cooper. Records and record types in semantic theory. *J. Logic and Compututation*, 15(2), 2005.

[11] The Coq Development Team. *The Coq Proof Assistant Reference Manual (Version 8.3), INRIA*, 2010.

[12] Z. Luo. *Computation and Reasoning: A Type Theory for Computer Science.* Oxford University Press, 1994.

[13] Z. Luo. Type-theoretical semantics with coercive subtyping. *Semantics and Linguistic Theory 20 (SALT20), Vancouver*, 2010.

[14] Z. Luo. Contextual analysis of word meanings in type-theoretical semantics. *Logical Aspects of Computational Linguistics (LACL'2011). LNAI 6736*, 2011.

[15] Z. Luo. Common nouns as types. In D. Bechet and A. Dikovsky, editors, *Logical Aspects of Computational Linguistics (LACL'2012). LNCS 7351*, 2012.

[16] Z. Luo. Formal semantics in modern type theories with coercive subtyping. *Linguistics and Philosophy*, 35(6):491–513, 2012.

[17] Z. Luo. Formal Semantics in Modern Type Theories: Is It Model-theoretic, Proof-theoretic, or Both? *Invited talk at Logical Aspects of Computational Linguistics 2014 (LACL 2014), Toulouse. LNCS 8535*, pages 177–188, 2014.

[18] Z. Luo. A Lambek calculus with dependent types. *Types for Proofs and Programs (TYPES 2015), Tallinn*, 2015.

[19] Z. Luo and R. Pollack. LEGO Proof Development System: User's Manual. LFCS Report ECS-LFCS-92-211, Dept of Computer Science, Univ of Edinburgh, 1992.

[20] Z. Luo and Y. Zhang. A Linear Dependent Type Theory. *Types for Proofs and Programs (TYPES 2016), Novi Sad*, 2016.

[21] P. Martin-Löf. *Intuitionistic Type Theory.* Bibliopolis, 1984.

[22] B. Nordström, K. Petersson, and J. Smith. *Programming in Martin-Löf's Type Theory: An Introduction.* Oxford University Press, 1990.

[23] A. Ranta. *Type-Theoretical Grammar.* Oxford University Press, 1994.

[24] C. Retoré. The Montagovian generative lexicon $\lambda\mathsf{Ty}_n$: an integrated type-theoretical framework for compositional semantics and lexical pragmatics. TYPE 2013, 2013.