Formal Semantics in Modern Type Theories: Theory and Implementation

(Proposal for Advanced Course in Language & Logic)

S. Chatzikyriakidis* and Z. Luo†
Department of Computer Science
Royal Holloway, University of London
Egham, Surrey, TW20 0EX, U.K.

June 2013

Abstract

Formal semantics based on Modern Type Theories (MTTs) provides us with not only a viable alternative to Montague Grammar, but potentially an attractive full-blown semantic tool with advantages in many respects. We shall introduce the MTT-based semantics and then study several issues such as adjectival modification, co-predication and coordination. Key comparisons to Montague Grammar are done all along, discussing the advantages of MTTs over simple type theory. For example, subtyping is crucially needed for proper semantic treatments of some linguistic features but has proven difficult in a Montagovian setting; coercive subtyping is adequate for MTTs and has become a key foundation for the MTT-based semantics. MTTs have been implemented in proof assistants, which can be used to implement MTT-based semantics and hence conduct computer-assisted reasoning in natural language. We shall consider NL inference implemented in Coq to show that such activities can be supported effectively.

1 Motivation and Description

Formal semantics as currently practised in linguistics is based on Church's simple type theory [6], as employed in Montague Grammar and the semantics exercised within the Montagovian tradition. Modern Type Theories (MTTs) provide us with not only a viable alternative, but in fact an attractive one with many methodological advantages[1]. In this course, we shall introduce the MTT-based formal semantics, compare it with the Montagovian methodology, and investigate how it can be used as a basis for computer-assisted reasoning in NLs.

Modern Type Theories, like Martin-Löf's type theory [19] and UTT [9], were developed for the foundations of constructive mathematics and their applications in computer science to such areas as program verification. Their application to formal semantics started with Ranta's work [21], where he uses Martin-Löf's type theory to discuss many issues in linguistic semantics. It

*Dr Stergios Chatzikyriakidis; email: stergios.chatzikyriakidis@cs.rhul.ac.uk
†Prof Zhaohui Luo; 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] One may argue that there are philosophical reasons to employ MTTs as well, but we shall not detail such considerations here, except that we shall touch this issue in the last lecture.

has been recognised that the MTT-based semantics would not be viable without an adequate notion of subtyping. Coercive subtyping [10, 17], which was developed initially for type theory's application to large proof development, provides such a framework of subtyping and it turns out that it has become one of the key foundations in MTT-based semantics [11, 15].

There are many advantages in using an MTT for formal semantics, as compared with the simple type theory employed in the Montagovian setting. The following two respects can be emphasised:

- *Compatibility of subtyping.* As mentioned above, subtyping is crucial for an MTT to be a viable language for formal semantics[2] and coercive subtyping provides a satisfactory solution. Furthermore, also very importantly, subtyping is needed when considering many linguistic features such as copredication [1, 20]. However, introducing subtyping into a Montagovian semantics has proven difficult, if not impossible. Coercive subtyping in MTTs solves this problem; for example, copredication can be dealt with in a satisfactory way [11, 23].

- *Powerful typing mechanisms.* MTTs provide us with powerful type structures that are not available in the simple type theory but very useful in formal semantics to deal with various linguistic features such as co-predication, adjectival modification, coordination and collective predication. These type structures include, for example,

  - Dependent Σ-types. An example for their use is to model adjectival modification [21] and this is the main (but not all) mechanism to support the doctrine 'CNs as types' (rather than predicates) [21, 14].
  - Dependent Π-types. This is another typical form of dependent types. Their uses include the support of polymorphism (eg, when typing adverbs in the context of multiple typing [12]; cf, [22]) and linguistic coercions in sophisticated situations [2].
  - Universes. Type universes (or types of types) turn out to be very useful in studying various respects in formal semantics. Examples include CN, the universe of (the interpretations of) common nouns, which allows useful quantification.
  - Many other type structures are also useful, examples of which include the disjoint union types which are used in formalising the semantics of CNs modified by privative adjectives [4].

Besides providing a subtyping framework, coercive subtyping also brings in useful mechanisms that can help model linguistic coercions [2], a subject we shall only touch, but do not discuss in detail, in the course.

MTTs have been implemented in *proof assistants* such as Coq [8] and Lego [16], computer systems that support interactive proof development. Such systems provide us ideal platforms to implement the MTT-based semantics and, based on such an implementation, to conduct computer-assisted reasoning in NLs. In order to show how such an implementation can be applied in practice, we take Natural Language Inference (NLI) in Coq as a case study.

- *Natural Language Inference in Coq.* We have implemented the MTT-based semantics in Coq [5, 12] and used it as a basis for NLI [18]. Particularly, we have formalised various examples, including those in the FraCas test suite [7], and conducted experiments which

[2] See §3.3 of [21] for the discussion of the problem of 'multiple categorisation of verbs'.

turn out to be very promising.[3] In this course, we shall introduce this implementation and show by examples how NLI can be supported by the computer, including a number of cases that are traditionally regarded as problematic. The issue of designing automated strategies for use with Coq is also discussed. The intention is to show that the use of an interactive theorem prover can help automation, in the sense of building better automated proof strategies using our experience with interactive proof engines.

2 Tentative Outline

Tentatively, the course consists of the following lectures:

- *Monday*: Introduction to MTTs and the MTT-based semantics. The basic concepts in MTTs such as consistent internal logic, many-sortedness, subtyping, dependent types and universes will be introduced intuitively in the context of their uses in linguistic semantics. The introduction of MTT-based semantics consists of two parts: the first part being the semantics of basic language features such as CNs, adjectives, adverbs, etc.

- *Tuesday*: Introduction to MTT-based semantics (continued). Continuing Monday's lecture, we shall discuss some more advanced linguistic features such as co-predication, adjectival modification and coordination, among others. We shall compare the MTT semantics with Montague Grammar and show that the former provides valuable constructions that are not available in the simple type theory.

- *Wednesday*: Introduction to the proof assistant Coq and the Coq implementation of MTT-based semantics. We shall introduce Coq as a natural toolkit to implement MTT-based semantics and explain how one can employ such an implementation in various applications, including the NL inference. NL inference will then be introduced (and further discussed in the following lecture).

- *Thursday*: Natural Language Inference (NLI) in MTTs and Coq. We shall explain how MTTs can be used to deal with NLI, including the use of subtyping and dependent types in such endeavours as those involving quantifiers, adjectival and adverbial modifications and temporal inferences, among others. These inferences have all been implemented in Coq. The system is tested by using example cases from the FraCas test suite.

- *Friday*: Lecture on more advanced and research issues in MTT-based semantics. We shall focus on three topics: (1) Comparative study between MTT-based semantics and Montague Grammar; (2) Why can we use an interactive theorem prover like Coq in this application? (3) In what sense the MTT-based semantics is a proof-theoretic semantics? These questions/problems will be discussed and tentative research issues raised.

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

[3] An encouraging fact is that we can use Coq's tactics to automate all of the reasoning examples. This leads to a promising future research: systematically using an interactive theorem prover like Coq to conduct automated reasoning tasks as well as sophisticated ones which have to be interactive. We shall briefly discuss this in the last lecture.

3 Expected Level and Prerequisites

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

4 Other Information

4.1 Proposed Lecturers

Both of the proposed lecturers are doing a research project on MTT-based formal semantics at Royal Holloway, University of London, funded by the Leverhulme Trust in U.K.

- *Prof Zhaohui Luo* is full professor, the grant holder of the above project. He has done research work in modern type theory [9], proof assistants [16], coercive subtyping [10, 17], and more recently, MTT-based semantics [15, 11, 12, 2, 14].

- *Dr Stergios Chatzikyriakidis* did his PhD in KCL and is currently a researcher in the above project. Together with Luo, he has worked on various aspects of MTT-based semantics, including NL inference [5], adjectival modifications [4] and coordination [3]. He has used Coq to implement the MTT-based semantics and the associated NLI engine [5].

More information on the above research project can be found at the following web page: `http://www.cs.rhul.ac.uk/home/zhaohui/lexsem.html`.

4.2 Previous ESSLLI Course

At ESSLLI 2011 in Ljubljana, one of the proposers (Z. Luo) has given a joint course with Prof N. Asher on Lexical Semantics [13].

References

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

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

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

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

[5] S. Chatzikyriakidis and Z. Luo. Natural Language Inference in Coq. Manuscript to be submitted to JoLLI, 2013.

[6] A. Church. A formulation of the simple theory of types. *J. Symbolic Logic*, 5(1), 1940.

[7] R. Cooper, D. Crouch, J. van Eijck, C. Fox, J. van Genabith, J. Jaspars, H. Kamp, D. Milward, M. Pinkal, M. Poesio, and S. Pulman. Using the framework. *Technical Report LRE 62-051r*, 1996. http://www.cogsci.ed.ac.uk/ fracas/.

[8] The Coq Development Team. *The Coq Proof Assistant Reference Manual (Version 8.1), INRIA*, 2007.

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

[10] Z. Luo. Coercive subtyping. *Journal of Logic and Computation*, 9(1):105–130, 1999.

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

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

[13] Z. Luo. Type-theoretical semantics with coercive subtyping. Lecture notes at ESSLLI 2011 (for Lexical Semantics, a course taught together with N. Asher), Ljubljana, 2011.

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

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

[16] 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.

[17] Z. Luo, S. Soloviev, and T. Xue. Coercive subtyping: theory and implementation. *Information and Computation*, 223, 2012.

[18] B. MacCartney. *Natural Language Inference.* PhD thesis, Stanford Universisty, 2009.

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

[20] J. Pustejovsky. *The Generative Lexicon.* MIT, 1995.

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

[22] C. Retore. The Montagovian generative lexicon ΛTy_n: an integrated type-theoretical framework for compositional semantics and lexical pragmatics. Manuscript, 2013.

[23] T. Xue and Z. Luo. Dot-types and their implementation. *Logical Aspects of Computational Linguistics (LACL 2012). LNCS 7351*, 2012.