Introduction: Modern Perspectives in Type Theoretical Semantics

Stergios Chatzikyriakidis and Zhaohui Luo

Abstract Type theories, from the early days of Montague Semantics (Montague 1974) to the recent work of using rich or modern type theories, have a long history of being employed as foundational languages of natural language semantics. In this introductory chapter, we will describe and discuss the development of type theories as foundational languages of mathematics, as well as their applications as foundational languages for formal semantics. In the end, a brief description of each chapter in the volume will follow.

1 Type Theories: Historical Development

Type theory has a long and fruitful tradition spanning across multiple theoretical domains including logic, mathematics, computer science, philosophy and linguistics. The main, or at least original, motivation behind the development of type theory was to study the foundations of mathematics. For example, going back to the beginning of the 20th century, Russell's motivation for developing his Ramified Theory of Types (White and Russell 1925; Russell 1992) was to solve a foundational problem in Cantor's naive set theory exposed as a number of well-known contradictions relating to self-reference, including Russell's paradox. Some researchers, including Russell

---

S. Chatzikyriakidis—Supported by the Centre of Linguistic Theory and Studies in Probability in Gothenburg.
Z. Luo—Partially supported by the research grants from Royal Academy of Engineering, EU COST Action CA15123, and the CAS/SAFEA International Partnership Program for Creative Research Teams.

S. Chatzikyriakidis (✉)
University of Gothenburg, Dicksonsgatan 4, 41256 Göteborg, Sweden
e-mail: stergios.chatzikyriakidis@gu.se

S. Chatzikyriakidis
Open University of Cyprus, Latsia B1 33, Nicosia, Cyprus

Z. Luo
Royal Holloway, University of London, Egham, Surrey TW20 0EX, UK
e-mail: zhaohui.luo@hotmail.co.uk

S. Chatzikyriakidis and Z. Luo (eds.), *Modern Perspectives in Type-Theoretical Semantics*, Studies in Linguistics and Philosophy 98, DOI 10.1007/978-3-319-50422-3_1

himself, attributed such paradoxes to 'vicious circles' in the formations of logical formulae ('impredicativity', in a technical jargon), which is what Russell's theory of ramified types was designed to circumvent.

However, Ramsey (1925) pointed out that it was the logical paradoxes (e.g., Russell's paradox), not the semantic ones (e.g., the Liar's paradox), that can (and should) be avoided in formulations of logical calculi and that Russell had mixed up these two kinds of paradoxes, leading to complications and problems in his theory of ramified types. As Ramsey argued, although impredicativity in formula formation is circular, it is not vicious. Based on this, Ramsey suggested that the theory of ramified types can be 'simplified' into Simple Type Theory (STT), which was later formally formulated in 1940 by Church using λ-notations (Church 1940) and used by Montague in his Intensional Logic (IL) (Montague 1974) (and Gallin's TY_2 Gallin 1975 – see below) to represent formal semantics of natural language.

The above development of type theories has been driven by the search for foundational languages for classical mathematics. In the 1970s, various researchers studied foundational languages for constructive rather than classical mathematics. Besides other systems, Martin-Löf's type theory (Martin-Löf 1975, 1984), especially its intensional version as described in Part III of Nordström et al. (1990), has been widely studied and applied to the foundations of mathematics, computer science and linguistic semantics. It contains powerful typing mechanisms such as dependent typing, inductive typing and type universes. Its study, together with that of simple type theory, has led to the development of a family of (intensional) type theories called Modern Type Theories (MTTs), including the predicative type theories such as Martin-Löf's intensional type theory (Nordström et al. 1990) and the impredicative type theories such as the calculus of constructions (Coquand and Huet 1988) and the Unifying Theory of dependent Types (UTT) (Luo 1994). In computer science, MTTs have been implemented in proof assistants such as Agda (The Agda proof assistant 2008), Coq (The Coq Team 2007) and Lego/Plastic (Luo and Pollack 1992; Callaghan and Luo 2001), and used in applications such as the formalisation of mathematics and verification of programs. It is worth remarking that, although formalising constructive mathematics was the main motivation of the early development of Martin-Löf's type theory, it is not the case that modern type theories can only be employed constructively. Put in another way, powerful typing is not monopolised by constructive mathematics or constructive reasoning; instead, it can be used in much wider applications such as linguistic semantics to be studied in this book.

2 Type Theories as Foundational Languages of Formal Semantics

The application of type theory to formal semantics has been initiated by Montague's pioneering work (Montague 1974). Montague employed Church's simple type theory STT (Church 1940) (and Henkin's model theory of STT Henkin 1950) as the foundational language for formal semantics. This has since become the dominant approach in this field. An enormous amount of work based on Montague's original system IL, or its extensions/variations/simplifications, have been produced since then. One such work is Gallin's study of TY_2, a reformulation of Church's STT with an extra base type (concerning intensions), and his translation of Montague's IL into TY_2 which establishes a solid foundation for Montague semantics (Gallin 1975). For example, Gallin's work shows that everything expressible in IL can be expressed in STT/TY_2 and explains away some meta-theoretic deficiencies of IL as discussed in, for example, Muskens (1996). A number of researchers have been using Gallin's formulation ever since. There are many other related research on NL semantics in type theory. For instance, research related to frameworks such as dynamic logic (Groenendijk and Stokhof 1991) and Discourse Representation Theory (Kamp and Reyle 1993) includes Groenendijk and Stokhof's work (Groenendijk and Stokhof 1990) on extending Dynamic Predicate Logic with simply typed lambda calculus, in effect a Dynamic Montague Grammar, and Muskens' work of combining DRT with Montague Grammar (Muskens 1996).

In the last two decades or so, researchers have worked on employing rich type theories for formal semantics. In his seminal work, Ranta (1994) proposed to study various aspects of NL semantics using Martin-Löf's intensional type theory (a typical Modern Type Theory). Although Ranta had a more modest goal in his mind (and might not be thinking that he was developing a logical semantics per se), his work has laid down the foundations of type theories with rich type structures as foundational languages for formal semantics. Many other researchers have also recognised the potential advantages of rich type structures for formal semantics including, for example, Sundholm (1989), Luo and Callaghan (1998), Boldini (2000), Cooper (2005), Dapoigny and Barlatier (2009), Bekki (2014), Retoré (2013). More recently, there has been a move to develop Modern Type Theories as a full-blown setting for formal semantics (sometimes called MTT-semantics – see Luo 2012; Chatzikyriakidis and Luo 2014 among other papers). One of the notable developments is the application of subtyping (in particular, coercive subtyping Luo et al. 2012) in MTT-semantics, a crucial feature that allows the CNs-as-types paradigm to work in a proper way. The MTT-semantics has also been studied from many different angles and aspects including the studies of selectional restrictions, various classes of adjectives and adverbials, coordination and event semantics, among other things. Furthermore, it has been argued in Luo (2014) that the MTT-semantics has advantages of both proof-theoretic semantics (philosophically as discussed in Kahle and Schroeded-Heister 2006 and practically in its direct support of computer-assisted reasoning in proof assistants) and model-theoretic semantics (the rich type structures in MTTs deliver a wide semantic coverage of linguistic features).

At this point, it may be worth pointing out that types in type theories as foundational languages of formal semantics are different from sets in set theory, although both represent collections of objects/elements. In a nutshell, the difference may be summarised by saying (very informally, of course) that such types are only manageable sets in the sense that some sets and set operations (e.g., intersection and union), are not available in the world of types for, otherwise, some of the salient and important properties of such type theories would be lost.[1] For example, in type theories for formal semantics (either the simple type theory STT as used in Montague's semantics or MTTs in MTT-semantics), type checking is decidable; in layman's terms, it is mechanically checkable whether any object a is of type A.[2] In STT, this means that one can check mechanically whether an object is of type e of entities, or of type t of propositions, or of a function type $A \to B$. For STT, obviously this must be decidable for otherwise the internal higher-order logic would not work properly (e.g., the applications of its rules would become infeasible). This is similarly the case for MTTs for otherwise one would not have a working internal logic that is necessary for formal semantics.

3 Contents of the Volume

The current volume consists of original contributions to type theoretical semantics and related topics. They are divided into the following three parts.

Part I. Foundational Issues

This part consists of four chapters. In this chapter, Bekki and Mineshima study anaphoric expressions and presuppositions in the framework of Dependent Type Semantics (DTS), as considered in an impredicative type system. The employment of Martin-Löf's type theory, especially the semantic treatment of anaphora by means of strong sum types (Σ-types) (Sundholm 1986; Ranta 1994), was one of the early notable successes in application of dependent type theory to formal semantics. This chapter presents a comprehensive study of anaphora and presupposition in an impredicative type system and, in particular, emphasises the importance of underspecification in their semantic treatments.

In Chap. 2, Chatzikyriakidis and Luo discuss the issue of interpreting common nouns (CNs) in type theoretical semantics. The authors first consider several approaches to the interpretation of CNs, either as predicates or as types, discussing their advantages and potential problems. The paper then focuses on a proposal of dealing with some of the negative sentences, a challenging topic in MTT-semantics where CNs are interpreted as types. The authors propose a theoretical framework in the type theory UTT to deal with negated sentences and use the proof assistant Coq to verify various examples of reasoning based on the proposal. The chapter also discuses the use of index types to formalise CNs in more advanced situations involving temporal sensitivity and gradability.

[1] Such properties include, for example for MTTs, the meta-theoretic properties such as normalisation, canonicity, decidability, among others. Normally, these properties like decidability of type checking would not hold if one had intersection types, union types or other potentially destructive types (see, for example, Pierce 1991 for more information).

[2] In contrast, the truth of the membership relation $e \in S$ in set theory is undecidable since it is just a logical formula in the first-order logic.

In Chap. 3, Cooper presents and discusses TTR, a framework of concepts and notations that has been used extensively in recent years to deal with a number of NL phenomena that need fine-grained and richer systems like dialogue modelling, non-logical inference and copredication. Although TTR is set-theoretic, its development, especially in the early days, was very much influenced by Martin-Löf's type theory and the notion of records studied by Betarte and Tasistro (1998); Betarte (1998); Tasistro (1997). Presenting the TTR framework in its current form, the author clarifies several very interesting issues and discusses motivations and applications.

In Chap. 4, Grudzinska and Zawadowski, motivated by their goal to give a uniform treatment of unbounded anaphora and generalised quantification, present a system based on the idea that (dependent) types are fibrations in category theory.[3] The authors, even though using a dependently typed and many-sorted system which is usually associated with a proof theoretic approach, take a model theoretic stance instead, where truth and reference rather than proofs are used. Using this hybrid approach, i.e. via employing elements from both Montagovian and Martin-Löf influenced approaches, the authors manage to provide an account of long standing issues in anaphora and quantification like quantificational subordination, cumulative and branching continuations, and donkey anaphora.

Part II. Types and Applications

This part consists of four chapters. In Chap. 5, Asher, Abrusan and van de Cruys discuss co-composition, a phenomenon that occurs during composition of words into phrases or sentences. In a Montagovian setting where CNs are interpreted as predicates, the authors take co-composition to be different from linguistic coercions in that coercions are triggered by type mismatches, while co-compositions are not.[4] Therefore, in a Montagovian setting, dealing with co-composition is challenging. The authors initiate a very interesting study of combining distributional and type-theoretical semantics in the hope that the former may provide an effective way to address the co-composition phenomena when these are not triggered by type mismatch.

In Chap. 6, Mery and Retoré study the notion of lexical sorts (or sometimes called base types) in type-theoretic lexical semantics. The authors put forth their account using a multi-sorted type system based on Girard's system F. After analysing the problem and the features of lexical sorts, the authors make the proposal that classifiers may provide vital clues in studying and even fixing lexical sorts. Although such a claim needs to be verified either empirically or by providing further evidence of its effectiveness, the proposal is attractive and merits further elaboration.

In Chap. 7, Hough and Purver present a dialogue system based on a number of different ideas: a probabilistic extension of TTR, a dynamic model of syntax (Dynamic

[3]In semantic studies of dependent type theories, interpreting types as fibrations in category theory is one of the typical approaches.

[4]It is worth pointing out that, if CNs are interpreted as types (e.g., in MTT-semantics), co-composition is also triggered by type mismatches and, therefore in such semantic frameworks, one would not distinguish co-compositions from coercions in such a way (private communication between Asher and Luo).

Syntax) and order theoretic models of probability. The end result is an incremental dialogue system, equipped with an expressive semantic backbone (ProbTTR) that can be used to model incremental reference processing. This chapter, as well as Chap. 10, is a nice example of how semantic formal systems can be combined with work in probability theory or distributional semantics in order to produce richer systems that can overcome problems individual approaches may face.

In Chap. 8, Fernando considers the string model approach based on finite automata for propositional logical systems like Hindley-Milner logic. The author links this idea to the study of various NL semantic phenomena such as event semantics. The model is then employed to study temporal properties which facilitate the description of NL phenomena such as tense and aspect.

Part III. Implementational Aspects

This part consists of two chapters. In Chap. 9, Moot discusses a variety of tools that can be used in the implementation and testing of variants of categorial grammars. Moot further discusses a number of advances in the area of type-logical grammar, concentrating on approaches that either add syntactic flexibility e.g. multimodal categorial grammars or approaches that add semantic expressiveness/fine-grainedness e.g. the Montagovian Generative Lexicon (Retoré 2013). The paper is a very important case study of how computational tools can help in the development and verification of the properties of formal linguistic models of syntax/semantics.

In Chap. 10, Angelov presents an initial study combining probability theory with NLP systems of syntax and semantics based on type theories. Angelov takes a different stance than Robin Cooper et al. (2015) and instead of introducing probabilistic type assignments, he introduces probability distributions over predefined members of a type. The result is a paper which is a step forward in the direction of combining stochastic and logical methods, and might further provide insights for probabilistic type theories as well as their use for more practical NLP applications.

References

The Agda proof assistant (version 2) (2008). Available from the web page: http://appserv.cs.chalmers.se/users/ulfn/wiki/agda.php.

Bekki, D. (2014). Representing anaphora with dependent types. *LACL 2014, LNCS 8535*.

Betarte, G. (1998). *Dependent record types and algebraic structures in type theory*. PhD thesis, Chalmers University of Technology.

Betarte, G., & Tasistro, A. (1998). Extension of Martin-Löf's type theory with record types and subtyping. In G. Sambin & J. Smith (eds.), *Twenty-five Years of Constructive Type Theory*. Oxford: Oxford University Press.

Boldini, P. (2000). Formalizing context in intuitionistic type theory. *Fundamenta Informaticae*, *42*(2), 1–23.

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

Chatzikyriakidis, S., & Luo, Z. (2014). Natural language inference in Coq. *Journal of Logic, Language and Information, 23*(4).

Church, A. (1940). A formulation of the simple theory of types. *The Journal of Symbolic Logic*, *5*(1), 56–68.
R. Cooper. (2005). Records and record types in semantic theory. *Journal of Logic and Compututation*, 15(2).
Cooper, R., Dobnik, S., Larsson, S., & Lappin, S. (2015). Probabilistic type theory and natural language semantics. *LiLT (Linguistic Issues in Language Technology)*, *10*.
The Coq Team (2007). *The Coq Proof Assistant Reference Manual (Version 8.1), INRIA*.
Coquand, Th, & Huet, G. (1988). The calculus of constructions. *Information and Computation*, *76*(2/3),
Dapoigny, R., & Barlatier, P. (2009). Modeling contexts with dependent types. *Fundamenta Informaticae*, *21*.
Gallin, D. (1975). *Intensional and higher-order modal logic: with applications to Montague semantics*. Amsterdam: North-Holland.
Groenendijk, J., & Stokhof, M. (1990). Dynamic Montague grammar. *Proceedings of the Second Symposion on Logic and Language*.
Groenendijk, J., & Stokhof, M. (1991). Dynamic predicate logic. *Linguistics and Philosophy*, *14*(1).
Henkin, L. (1950). Completeness in the theory of types. *Journal of Symbolic Logic*, *15*.
Kahle, R., & Schroeded-Heister, P. (Eds.). (2006). Proof-Theoretic Semantics. *Special Issue of Synthese*, *148*(3).
Kamp, H., & Reyle, U. (1993). *From discourse to logic*. The Netherlands: Kluwer.
Luo, Z. (1994). *Computation and reasoning: a type theory for computer science*. Oxford: Oxford University Press.
Luo, Z. (2012). Formal semantics in modern type theories with coercive subtyping. *Linguistics and Philosophy*, *35*(6), 491–513.
Luo, Z. (2014). Formal semantics in modern type theories: is it model-theoretic, proof-theoretic, or both? In *Invited talk at Logical Aspects of Computational Linguistics 2014 (LACL 2014), Toulouse. LNCS* (Vol. 8535, pp. 177–188).
Luo, Z., & Callaghan, P. (1998). Coercive subtyping and lexical semantics (extended abstract). *LACL'98 (extended Abstracts), Available in Request to the First Author or as* http://www.cs.rhul.ac.uk/home/zhaohui/LACL98.abstract.ps.
Luo, Z., & Pollack, R. (1992). LEGO Proof Development System: User's Manual. LFCS Report ECS-LFCS-92-211, Department of Computer Science, University of Edinburgh.
Luo, Z., Soloviev, S., & Xue, T. (2012). Coercive subtyping: theory and implementation. *Information and Computation*, *223*, 18–42.
Martin-Löf, P. (1975). An intuitionistic theory of types: predicative part. In H. Rose & J. C. Shepherdson (Eds), *Logic Colloquium'73*.
Martin-Löf, P. (1984). *Intuitionistic type theory*. Berkeley: Bibliopolis.
Montague, R. (1974). *Formal philosophy: Selected papers of Richard Montague*. New Haven: Yale University Press.
Muskens, R. (1996). Combining Montague semantics and discourse representation. *Linguistics and Philosophy*, *19*(2), 143–186.
Nordström, B., Petersson, K., & Smith, J. (1990). *Programming in Martin-Löf's type theory: An introduction*. Oxford: Oxford University Press.
Pierce, B. (1991). Programming with intersection types, union types, and polymorphism. *Technical report CMU-CS-91-106, School of Computer Science, CMU*.
Ramsey, F. (1925). The foundations of mathematics. *Proceedings of the London Mathematical Society*, *25*, 338–384.
Ranta, A. (1994). *Type-theoretical grammar*. Oxford: Oxford University Press.
Retoré, C. (2013). The Montagovian generative lexicon $\lambda\mathsf{Ty}_n$: A type theoretical framework for natural language semantics. In R. Matthes & A. Schubert (Eds), *Proceedings of TYPES2013*.
Russell, B. (1992). *The principles of mathematics*. Routledge, 1903. Paperback edition.
Sundholm, G. (1986). Proof theory and meaning. In D. Gabbay & F. Guenthner (Eds.), *Handbook of Philosophical Logic* (Vol. III).

Sundholm, G. (1989). Constructive generalized quantifiers. *Synthese*, *79*(1), 1–12.
Tasistro, A. (1997). *Substitution, record types and subtyping in type theory*. Ph.D thesis, Chalmers University of Technology.
White, A., & Russell, B. (1925). *Principia Mathematica* (2nd Edn). Cambridge: Cambridge University Press.