From Montague Semantics to MTT-Semantics: A Meaningful Comparison

Stergios Chatzikyriakidis and Zhaohui Luo

(ESSLLI 2019: Lecture 2 by ZL)

MTT-semantics: some key features

- MTT-semantic features in comparison with Montague
  - Overview, some detailed further in later lectures
  - Only focussing on several, others omitted
- Features of MTT-semantics (& differences with MG)
  - Rich type structure
  - Powerful tools for semantic constructions
  - Both model-theoretic and proof-theoretic
  - Others (if time permits: eg, judgemental interpretation and identity criteria)

I. Rich type structures & CNs-as-types

- In MTTs, many types with rich structures
  - Dependent types (Π-types, Σ-types, …)
  - Inductive types (Nat, Fin(n), …)
  - Universes – types of types (logical, linguistic, …)

  In contrast, simple type theory has only e, t, $\alpha \rightarrow \beta$.
- In linguistic semantics, this allows CNs-as-types.
  - For example, adjectival modification (see SC's lecture)

Classification	Inference	Example	MTT-types/mechanisms
Intersective	$Adj[N] \Rightarrow N \ \& \ Adj$	handsome man	Σ-types
Subsective	$Adj[N] \Rightarrow N$	large mouse	Π-polymorphism
Privative	$Adj[N] \Rightarrow \neg N$	fake gun	disjoint union types
Non-committal	$Adj[N] \Rightarrow ?$	alleged criminal	modal collection

- Several advantages of CNs-as-types (not predicates)
  - Selection restriction by typing
    - Capturing category errors
    - MG cannot do this due to CNs-as-predicates.
  - Compatibility with subtyping
    - Subtypes or "subsorts" (cf, Asher, Partee): Phy, Event, Containers, …
    - MG has been problematic in this because of CNs-as-predicates.
  - Proper treatment of copredication
    - Dot-types in MTTs (see SC's latter lecture)
    - Problematic if CNs-as-predicates

I.1. Selection Restriction

- (*) The table talks.
  - Is (*) meaningful? (a category error)
- In MG, yes: (*) has a truth value
  - talk(the table) is false in the intended model.
- In MTT-semantics, no: (*) is not meaningful.
  - "the table" is of type Table, not of type Human and, hence, talk(the table) is ill-typed as talk requires that its argument be of type Human.
  - In MTT-semantics, meaningfulness = well-typedness

I.2. Compatibility with subtyping

- Researchers considered various subtypes/subsorts:
  - Phy, Info $\leq$ e (Asher 2011 on copredication)
  - Basket $\leq$ Container, … (Partee-Borschev 2014, on adjectival modification)
  - Events … (c.f., ZL's lecture on Friday)
- Such are incompatible with CNs-as-predicates, although they are OK with CNs-as-types.
  - Let's consider an example – copredication.
  - I'm only using this as an example – see SC's latter Lecture 4.

Copredication

- Copredication (Asher, Pustejovsky, …)
  - John picked up and mastered the book.
  - The lunch was delicious but took forever.
  - The newspaper you are reading is being sued by Mia.
  - … …
- How to deal with this in formal semantics
  - Dot-objects (eg, Asher 2011, in the Montagovian setting)
  - It has a problem: subtyping and CNs-as-predicates strategy do not fit with reach other …

Subtyping problem in the Montagovian setting

- Problematic example (in Montague with CNs-as-pred)
  - heavy : $Phy \rightarrow t$ and book : $Phy \bullet Info \rightarrow t$
  - heavy book = $\lambda x{:}Phy.\ book(x) \wedge heavy(x)$?
  - In order for this, we'd need <u>$Phy \leq Phy \bullet Info$</u> (#)

    But, intuitively, this is <u>not</u> the case (the opposite should be)!
  - A higher type of heavy : $(Phy \rightarrow t) \rightarrow (Phy \rightarrow t)$ would not help.
- In MTT-semantics, because CNs are interpreted as types, things work as intended.
  - heavy : $Phy \rightarrow Prop$ and $Book \leq Phy \bullet Info \leq Phy$
  - So, <u>$heavy(b) : Prop$</u> is well-typed, for b : Book.

❖ In MTT-semantics, CNs are types – we have:

"John picked up and mastered the book."

$[\![book]\!] \leq PHY \bullet INFO$

$[\![pick\ up]\!] : Human \rightarrow PHY \rightarrow Prop$

$\leq Human \rightarrow PHY \bullet INFO \rightarrow Prop$

$\leq Human \rightarrow [\![book]\!] \rightarrow Prop$

$[\![master]\!] : Human \rightarrow INFO \rightarrow Prop$

$\leq Human \rightarrow PHY \bullet INFO \rightarrow Prop$

$\leq Human \rightarrow [\![book]\!] \rightarrow Prop$

Hence, both have the same type (in LType – cf, SC's Lect 1.2) and therefore can be coordinated by "and" to form "picked up and mastered" in the above sentence.

Remark: CNs as types in MTT-semantics – so things work.

II. MTT-tools for semantic constructions

- Rich typing → powerful tools
- Examples:
  - Π-polymorphism via universes
  - Overloading by coercive subtyping
  - Π/Σ-organisation (omitted here)
  - … …

II.1. Sense selection via overloading

- Sense enumeration (cf, Pustejovsky 1995 and others)
  - Homonymy
  - Automated selection
  - Existing treatments (eg, Asher et al via disjoint union types)
- For example,
  1. John runs quickly.
  2. John runs a bank.

  with homonymous meanings
  1. $[\text{run}]_1$: Human→Prop
  2. $[\text{run}]_2$: Human→Institution→Prop

  "run" is <u>overloaded</u> – how to disambiguate?

Overloading via coercive subtyping

- Overloading can be represented by coercions

  Eg.

  $$run : 1_{run} \xrightarrow{c_1} [run]_1 : Human \rightarrow Prop$$

  $$run : 1_{run} \xrightarrow{c_2} [run]_2 : Human \rightarrow Institution \rightarrow Prop$$

- Now, "John runs quickly" = "John $[run]_1$ quickly".

  "John runs a bank" = "John $[run]_2$ a bank".

- Homonymous meanings can be represented so that automated selection can be done according to typing.
- Remark: This could not be done if CNs-as-predicates.

II.2. Π-polymorphism – an example mechanism

- Π-polymorphism offers several applications.
  - They are not available in MG where CNs are interpreted as predicates.
- Π-types: Informally (borrowing set-theoretical notations):

  $\Pi x{:}A.B[x] = \{ f \mid \text{for any } a : A,\ f(a) : B[a] \}$

  These f's are dependent functions.
- Example
  - $\Pi x{:}Human.Child(x)$, type of functions mapping h to Child(h), the type of children of h (may be an empty type).
- Notational conventions:
  - $A \to B$ stands for $\Pi x{:}A.B(x)$ when $x \notin FV(B)$.
  - In other words, $A \to B$ are just special cases of Π-types.
  - So, a type theory with Π-types and Prop contains simple type theory.

Π-polymorphism – a first informal look

- How to model predicate-modifying adverbs (eg, quickly)?
  - Informally, it can take a verb and return a verb.
- Montague: quickly : $(e \rightarrow t) \rightarrow (e \rightarrow t)$
  quickly(run) : $e \rightarrow t$
- MTT-semantics: quickly : $(A_q \rightarrow Prop) \rightarrow (A_q \rightarrow Prop)$, where A_q is the domain/type for quickly.
  - What about other verbs? (eg, A_{talk}=Human, …)
  - Can we do it generically – one type of all adverbs?
- Π-polymorphism: quickly : $\Pi A{:}CN.\ (A \rightarrow Prop) \rightarrow (A \rightarrow Prop)$
- Question: What is CN?

  Answer: CN is a universe of types – next slide.

Universes – types of types

- Universe of types
  - Martin-Löf introduced the notion of universe (1973, 1984)
  - A universe is a type of types (Note: the collection Type of all types is not a type itself – otherwise, logical paradox.)
- Examples
  - Math: needing a universe to define type-valued functions
    - f(n) = N x … x N (n times)
  - MTT-semantics: for example,
    - CN is the universe of types that are (interpretations of) CNs. We have:

      Human : CN, Book : CN, Σ(Man,handsome) : CN, … …
    - We can then have: quickly : ΠA:CN. (A→Prop)→(A→Prop)

Modelling subsective adjectives

- Examples: large, skilful, …
- Nature of such adjectives
  - Their meanings are dependent on the nouns they modify.
  - "a large mouse" is not a large animal ("large" in "a large mouse" is only large for mice, not for other animals/entities.)
- This leads to proposal of using Π-polymorphism:
  - large : $\Pi A{:}CN.\ (A \rightarrow Prop)$
    - CN – type universe of all (interpretations of) CNs
  - large(Mouse) : Mouse → Prop
  - [large mouse] = Σx:Mouse. large(Mouse)(x)

Another example – type of quantifiers

- Generalised quantifiers
  - Examples: some, three, a/an, all, …
  - In sentences like: "Some students work hard."
- With Π-polymorphism, the type of binary quantifiers is (Lungu 2014):

$$\Pi A{:}CN.\ (A \to Prop) \to Prop$$

For Q of the above type, N : CN and V : N→Prop

$$Q(N,V) : Prop$$

E.g., Some(Student, work_hard) : Prop

CNs-as-predicates in MTTs?

- What if using an MTT but CNs-as-predicates?
  - In an MTT, one could still formally follow Montague:
    - use a single type e of all entities, and
    - use predicates of type e→t to interpret CNs.
- First, this seems unnecessary, at least.
  - Why doesn't one just use simple type theory STT?
  - STT is a simpler "subsystem" – why much bigger system?
- Secondly, most (if not all) of the advantages would be lost …

III. MTT-sem is both model/proof-theoretic

- Model-theoretic semantics (traditional)
  - Meaning as denotation (Tarski, …)
  - Montague: NL → (simple TT) → set theory
- Proof-theoretic semantics
  - Meaning as inferential use (proof/consequence)
  - Gentzen, Prawitz, …, Martin-Löf
  - e.g., Martin-Löf’s meaning theory
- MTT-semantics
  - Both model-theoretic and proof-theoretic – in what sense?
  - What does this imply?

Formal semantics in Modern Type Theories (MTT-semantics) is both model-theoretic and proof-theoretic.

- NL → MTT (representational, model-theoretic)
  - MTT as meaning-carrying language with its <u>types</u> representing collections (or "sets") and <u>signatures</u> representing situations
- MTT → meaning theory (inferential roles, proof-theoretic)
  - MTT-judgements, which are semantic representations, can be understood proof-theoretically by means of their inferential roles
- Z. Luo. Formal Semantics in Modern Type Theories: Is It Model-theoretic, Proof-theoretic, or Both? Invited talk at LACL14.

MTT-semantics being model-theoretic

- MTTs offer powerful representations.
- Rich type structure
  - Collections represented by types
  - Eg, CNs and their adjective modifications (see earlier slides)
  - Wide coverage – a major advantage of model-theoretic sem
- Useful contextual mechanisms – signatures
  - Various phenomena in linguistic semantics (eg, coercion & infinity)
  - Situations (incomplete world) represented by signatures (next slide)

MTT-semantics being model-theoretic (cont^{ed})

- Signatures Σ as in (cf, Edin LF [Harper et al 1987])

$$\Gamma \vdash_{\Sigma} a : A$$

  with $\Sigma = c_1{:}A_1, \ldots, c_n{:}A_n$

- New forms besides c:A [Luo LACL14]

$$\ldots, c{:}A, \ldots, A \leq_c B, \ldots, c \sim a : A, \ldots$$

  - Subtyping entries (cf, Lungu's PhD thesis 2018)
  - Manifest entries (can be emulated by coercive subtyping)

- *Theorem (conservativity)*

  The extension with new signature entries preserves the meta-theoretic properties for coherent signatures.

MTT-semantics being proof-theoretic

- MTTs are representational with proof-theoretic sem
  - Not available before – cf, use theory of meaning
- MTT-based proof technology
  - Reasoning based on MTT-semantics can be carried out in proof assistants like Coq:
    - pretty straightforward but nice application of proof technology to NL reasoning (not-so-straightforward in the past …)
  - Some Coq codes can be found in: (example next slide)
    - Z. Luo. Contextual analysis of word meanings in type-theoretical semantics. Logical Aspects in Computational Linguistics. 2011.
    - S. Chatzikyriakidis & Z. Luo. NL Inference in Coq. JoLLI 23(4). 2014.
    - S. Chatzikyriakidis & Z. Luo. Proof assistants for NL semantics. LACL 2016.
    - T. Xue et al. Propositional Forms of Judgemental Interpretations. NLCS 2018.

Coq code (homonymy by overloading)

```
Definition CN := Set.
Parameters Bank Institution Human Man : CN.
Parameter John : Man.
Axiom mh : Man->Human.  Coercion mh : Man >-> Human.
Axiom bi : Bank->Institution.  Coercion bi : Bank >-> Institution.

(* unit type for "run" *)
Inductive Onerun : Set := run.
Definition T1 := Human->Prop.
Definition T2 := Human->Institution->Prop.
Parameter run1 : T1.
Parameter run2 : T2.
Definition r1 (r:Onerun) : T1 := run1.  Coercion r1 : Onerun >-> T1.
Definition r2 (r:Onerun) : T2 := run2.  Coercion r2 : Onerun >-> T2.

(* John runs quickly *)
Parameter quickly : forall (A:CN), (A->Prop)->(A->Prop).
Definition john_runs_quickly := quickly (run:T1) John.
(* John runs a bank *)
Definition john_runs_a_bank := exists b:Bank, (run:T2) John b.
```

- Why important?
  - Model-theoretic – powerful semantic tools
    - Much richer typing mechanisms for formal semantics
    - Powerful contextual mechanism to model situations
  - Proof-theoretic – practical reasoning on computers
    - Existing technology: proof assistants (Coq, Agda, Lego/Plastic, NuPRL)
    - Applications to NL reasoning
  - Leading to both of
    - Wide-range modelling as in model-theoretic semantics
    - Effective inference based on proof-theoretic semantics

Remark: MTT-semantics offers a new perspective – new possibility not available before!

IV. Several Further Features of MTTs

- Other features/topics:
  - Judgemental interpretations (Xue, Luo & Chatzikyriakidis 18)
  - Identity criteria of CNs (Luo 2012, Chatzikyriakidis & Luo 2018 – see SC's latter lecture)
  - Proof irrelevance (Luo 2019)
- First, introducing the last one (and, the others if time permits)

Proof Irrelevance

- Example to show:
  - Potential problem introduced by proof terms in MTTs (and how to solve it by proof irrelevance)
  - From another angle, MTTs are very powerful for semantics.
- Proof irrelevance
  - Any two proofs of the same proposition are the same.
  - To have adequate MTT-semantics, proof irrelevance needs be enforced in the underlying type theory.
  - Eg, in impredicative TTs like UTT, we can have

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

Examples in NL semantics

- Identity criteria for CNs [Luo 12, Chatzikyriakidis & Luo 18]
  - A handsome man is a pair (m,p) of type Σ(Man, handsome).
  - Two handsome men are the same iff they are the same man
    ➔ proof irrelevance (eg, proofs of handsome(m) are the same).
- Counting (the same problem as above)
  - Most students who passed some exams are happy.
    - Most z : [Σx:Student Σy:Exam.pass(x,y)]. happy(π_1(z))
    - Incorrect counting that takes proofs of Σy:Exam.pass(x,y) into account
  - I believe proof irrelevance provides a clean/easier solution.
    - Most z : [Σx:Student ∃y:Exam.pass(x,y)]. happy(π_1(z))
    - Correct counting by proof irrelevance (for the ∃-proposition)

Counting and Anaphora

- A problem when <u>both</u> are involved.
  - Thanks to Justyna Grudzińska for bringing this example to my attention.
- Most farmers who own a donkey beat it.
  - (#) Most z : [Σx:F Σy:D. own(x,y)]. beat(π_1(z), π_1(π_2(z)))
  - Incorrect counting as proofs in Σ are taken into account.
  - Note that, if you use traditional $\exists$ for both Σ to get correct counting, anaphora are problems (untyped – π_i don't exist)!
- A problem not solved satisfactorily before
  - Sundholm (1989) realised it, but only proposed an ad hoc solution.
  - Tanaka (2015) studied a similar solution (ad hoc & complicated).

- A solution in UTT (or $MLTT_h$), using both Σ and $\exists$:
  - Most z : $[\Sigma x{:}F\ \exists y{:}D.\ own(x,y)]$.

    $\forall z' : [\Sigma y'{:}D.\ own(\pi_1(z), y')].\ beat(\pi_1(z), \pi_1(z'))$
  - Correct counting because of proof irrelevance (for $\exists$-prop).
  - Correct anaphoric reference because of Σ.
- Details in my LACompLing18 paper (in press)
  - Title: "Proof Irrelevance in Type-Theoretical Semantics"
  - First (?) to do this in a "standard" logical system, rather than non-standard ones such as the non-monotonic Dynamic Predicate Logic.

Judgemental Interpretation

- Judgements v.s. propositions
- Example
  - John is a student.
  - j : Student (Here, Student is a type, not a predicate.)
- What about
  - John is not a student.
  - If John is a student, he is happy.
- The following are wrong:
  - (#) $\neg$(j : Student) -- illegitimate
  - (#) (j : Student) is not the case. -- a meta-level assertion

- So, we introduce IS(Student,x) : Prop
  - $\neg$IS(Student,j)
  - IS(Student,j) $\Rightarrow$ happy(j)
  - These are well-typed propositions.
- $IS_A(B,a)$
  - Introduced axiomatically.
  - Justified by heterogeneous equality in type theory.
  - Xue, Luo and Chatzikyriakidis. Propositional Forms of Judgemental Interpretations. NLCS18, Oxford. 2018.