Modern Type Theories and Montague Semantics: Comparisons and Beyond

Zhaohui Luo
Dept of Computer Science
Royal Holloway, Univ of London

Comparisons between (and discussions on)

- MTT-based semantics
  - Formal semantics in Modern Type Theories (MTTs)
- Montague semantics
  - Formal semantics in simple type theory

They are in the same spirit, but …

This talk

- Brief introduction to MTT-based semantics
- Discuss
  - Subtyping – why needed and how (cf, Asher)
  - Rich type structures in MTTs for "meaning assembly" in formal semantics (cf, Retore)

I. Formal semantics based on MTTs

- Modern Type Theories: examples
  - Predicative type theories
    - Martin-Löf's type theory, where propositions and types are identified
  - Impredicative type theories
    - Prop
      - Impredicative universe of logical propositions (cf, t in simple TT)
      - Internal totality (a type, and can hence form types, eg Table→Prop, Man →Prop)
    - F/F^{ω} (Girard), CC (Coquand & Huet)
    - ECC/UTT (Luo, implemented in Lego/Plastic)
    - pCIC (implemented in Coq/Matita)
  - Cf, Copper's talk
- MTT = Logic + Types

Simple v.s. Modern Type Theories

- Church's simple type theory (Montague semantics)
  - Base types ("single-sorted"): e and t
  - Composite types: e, t, e→t, (e→t)→t, …
- Modern type theories
  - Many types of entities – "many-sorted"
    - Table, Man, Human, Phy, … are all types.
  - Besides →-types, many other types/type constructions

Formal semantics based on MTTs

- Sentences as propositions
  - [A man walks] : Prop
- Common nouns as <u>types</u>
  - [man], [book], [table] : Type (fine-grained)
- Verbs as predicates over "meaningful" domains
  - [shout] : [human]→Prop
  - Note: "Meaninglessness" v.s. "falsity" (eg, "A table shouts.")
- Adjectives as predicates
  - [handsome] : [man]→Prop
  - [handsome man]? (see later)

Interpretations of CNs: Types v.s. Predicates

- Common nouns, interpreted as
  - predicates in the Montague semantics
  - types in the MTT-based semantics
- "man"
  - In MG, man : e→t
    - [handsome man] = λx:e. man(x) & handsome(x) : e→t
  - In MTTs, Man : Type
    - [handsome man] = ∑x:Man.Handsome(x) : Type
- Implications include:
  - Issue of compatibility with subtyping

II. Subtyping: Needs in Linguistic Semantics

- Subtyping in linguistic semantics
  - Work by Asher, Pustejovsky, ...
  - Linguistic subtypes: Phy, Info, Event, ...
- Subtyping is also needed for MTT-based sem
  - CNs as types ➔ subtypes needed!
  - Eg,
    - [shout] : [human]→Prop
    - [John shouts] = [shout](j) : Prop, for j : [man] ???
    - But this is ill-typed! ([man] is not [human])
    - We need [man] ≤ [human]

Subtyping: Incompatibility in MG

- Problematic example (in Montague semantics)
  - [heavy] : Phy→t [or, (Phy→t)→(Phy→t), similar problem]
  - [book] : Phy•Info→t
  - [heavy book] = λx:Phy. [heavy](x) & [book](x) ???
  - In order for the above to be well-typed, we need

    Phy ≤ Phy•Info

    But, this is not the case (the opposite is)!
- In MTTs, because CNs are interpreted as types, things work as intended.

Subtyping in MTT-based semantics

- Simple example
  - [book] : Type, [book] ≤ Phy•Info ≤ Phy/Info
  - [heavy] : Phy→Prop
  - [heavy book] = Σx:[book]. [heavy](x)
  - [heavy](x) is well-typed because [book] ≤ Phy.
- Copredication with dot-types (Asher, Pustejovsky)

  "John picked up and mastered the book."

  [pick up] : [human]→Phy→Prop
  ≤ [man]→Phy•Info→Prop
  ≤ [man]→[book]→Prop

  [master] : [human]→Info→Prop
  ≤ [man]→Phy•Info→Prop
  ≤ [man]→[book]→Prop

Coercive subtyping: adequate for MTTs

- Traditional "subsumptive subtyping"
  - Subsumption rule
  - Inadequate for MTTs: eg, canonicity fails
- Coercive subtyping
  - History: developed for proof development & program verification
  - Adequate for MTTs
  - Conservative, in fact, definitional extension

    (Soloviev & Luo 2002, Luo & Soloviev & Xue 2013)

Linguistic coercions

- Coercions in coercive subtying
  - Role in formalisation of coercions in linguistics
  - Supports most of linguistic coercions
  - cf, Nicholas' talk and (Asher & Luo in SuB17)
- Dependent types in coercion semantics
  - Previously, we only applied coercive subtyping to cases with non-dependent types.
  - Dependent types provide a useful mechanism for semantics.
  - Dependent types + coercions ➔ powerful tool
  - (Example later)

III. Rich type structure in MTTs

- MG is based on simple type theory, which has few type structures
- MTTs has rich type structures (as well as logic)
  - Types for "meaning assembly" (cf, Retore)
  - We explain some by examples of semantic interpretations:
    - Σ-types for modified CNs
    - Universes (eg, collection of CNs; interpretation of adverbs)
    - Dependent types in coercion semantics
    - Disjoint union types for some non-subsective adjectives

Types in MTTs: summary

- Propositional types
  - P⊃Q, ∀x:A.P(x), …
- Inductive types
  - Nat, AxB, A+B, List(A), …
- Dependent types
  - Σx:A.B(x) (intuitively, { (a,b) | a : A & b : B(a) })
  - Πx:A.B(x) (intuitively, { f : A→$\cup_{a\in A}$B(a) | a : A & b : B(a) })
- Universes
  - A universe is a type of (some other) types

III.1. Σ-types: interpretation of modified CNs

- Σ-types (also called "dependent sums")
  - Σx:A.B(x) consists of (a,b) such that a : A and b : B(a)
  - Note that B(x) depends on objects x of type A
- Modified CNs as Σ-types (Ranta)
  - "handsome man"
  - [man] : Type
  - [handsome](x) : Prop for x : [man]
  - [handsome man] = Σx:[man]. [handsome](x)

III.2. Universes

- A universe U is a type consisting of a collection of types – each object of U is a type.
- Example:
  - CN: the universe of types that interpret CNs, including modified CNs.
  - Universe CN is very useful: eg,
    - Type-lifting from A to (A→Prop)→Prop (Partee et al)
      - What is the range of A? Answer: A : CN.
      - Coercions A $\leq_{c(A)}$ (A→Prop)→Prop,
        where c : (A:CN)A→((A→Prop)→Prop) is defined as c(A,a,P) = P(a).
    - Semantics of adverbs in MTTs (next page)

Predicate-modifying adverbs

- Montague semantics:
  - [quickly] : (e→t)→(e→t)
  - [John walked quickly] = [quickly]([walk], j) : t
- How to do this in MTTs?
  - Problem: We have many types that interpret CNs (Table, Man, Animated, …), not a single e.
  - Solution:
    - [quickly] : ∏A:CN. (A→Prop)→(A→Prop)
    - [John walked quickly] = [quickly](Animated, [walk], j) : Prop,
      where [walk] : Animated→Prop.
  - Remark: the above type of [quickly] is both polymorphic and dependent.

III.3. Dependent types

- Example in (Asher & Luo 2013): using dependent types in coercion semantics
  - (32) *Jill just started War and Peace, which Tolstoy finished in 1820. But that won't last because she never finishes long novels.*
- Simple scoping restrictions (eg, local coercions) are not enough.
- Use dependent types (types of "start" etc – see next page):

$$start(j, wp)$$
$$\&\ finish(t, wp)$$
$$\&\ \neg last(j, wp)$$
$$\&\ \forall lb : (\Sigma b{:}Book.long(b)).\ finish(j, \pi_1(lb))$$

For example, for the above sentences (32), instead of $Event$, we may consider the family of types

$$Evt : Human \to Type;$$

intuitively, for any $h : Human$, the dependent type $Evt(h)$ is the type of events conducted by h. Now, we can assume that the verbs *start* etc have the following types:

$$\begin{array}{rcl} start,\ finish,\ last & : & \Pi h : Human.\ (Evt(h) \to Prop) \\ read,\ write & : & \Pi h : Human.\ (Book \to Evt(h)) \end{array}$$

Furthermore, we can consider the following coercions,[15] for any $h : Human$,

$$Book <_{c(h)} Evt(h),$$

where the coercion $c(h)$ is the function from $Book$ to $Evt(h)$ defined as follows: for any $b : Book$,

$$c(h,b) = \begin{cases} write(h,b) & \text{if } h \text{ wrote } b, \\ read(h,b) & \text{otherwise.} \end{cases}$$

III.4. Disjoint union types

- Disjoint union types A+B
  - Intuitively, disjoint union of A and B
  - (1) a : A → inl(a) : A+B; (2) b : B → inr(b) : A+B.
- Privative Adjectives (eg, fake)
  - Partee 2010: Privative Adjectives: Subsective plus Coercion
  - Interpreted subsectively together with 'type shifting' or 'type coercion' of the modified CNs.
  - This can be represented by disjoint union types (next page).

Example 1.1 *Consider the following types:*

- G_R: *the type of (real) guns*
- G_F: *the type of faked guns*
- $G = G_R + G_F$, *the disjoin union type (of real or faked guns)*

We declare the following coercive subtyping relations:

$$G_R \leq_{inl} G \quad and \quad G_F \leq_{inr} G,$$

With these types, one can define, for example, $real\_gun,\ fake\_gun : G \to Prop$ so that, for every $g : G$,

$$real\_gun(g) \quad iff \quad \neg fake\_gun(g),$$

and furthermore, because of subtyping, we have, for $r : G_R$ and $f : G_F$,

$$real\_gun(r) = True \quad and \quad real\_gun(f) = False.$$

Then, the following interpretations can be given, where $x : G$ and $f : G_F$:

- $[\![x \text{ is a real gun}]\!] = real\_gun(x)$
- $[\![f \text{ is not a real gun}]\!] = \neg real\_gun(f)$[1]

IV. Discussions for Future Work

- Logical semantics
  - Traditional MG: model-theoretic semantics
  - MTTs have been developed in proof theory.
  - Proof-theoretic semantics for NLs?
    - Existing work by Francez & Dyckhoff, not quite the same as Ranta's or ours.
- Model theory for MTTs
  - Recent, ongoing research on "univalent models" of MTTs (cf, Voevodsky's Univalent Axiom)
  - Does this lead to a general model theory for MTTs?