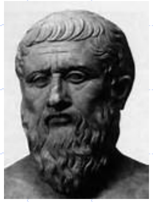
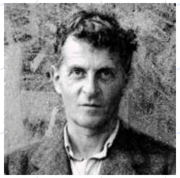
Formal Semantics in Modern Type Theories (MTT-semantics): advanced topics

Zhaohui Luo
Royal Holloway, Univ. of London

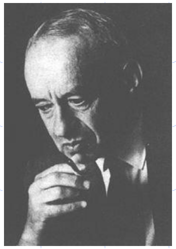
Lecture I. Introduction

Natural Language Semantics

- Semantics – study of meaning
  - communicate = convey meaning
- Various kinds of theories of meaning
  
  - Meaning is reference ("referential theory")
    - Word meanings are things (abstract/concrete) in the world.
    - c.f., Plato, …
  
  - Meaning is concept ("internalist theory")
    - Word meanings are ideas in the mind.
    - c.f., Aristotle, …, Chomsky.
  
  - Meaning is use ("use theory")
    - Word meanings are understood by their uses.
    - c.f., Wittgenstein, …, Dummett, Brandom.

Formal semantics

- Model-theoretic semantics
  - Meaning is given by denotation.
  - c.f., Tarski, Church, …, Montague
  - e.g., Montague grammar (MG)
    - NL → simple type theory → set theory




- Proof-theoretic semantics
  - In logics, meaning is inferential use (two aspects: proof + consequence)
  - c.f., Gentzen , …, Prawitz
  - e.g., meaning theories (c.f., previous page)

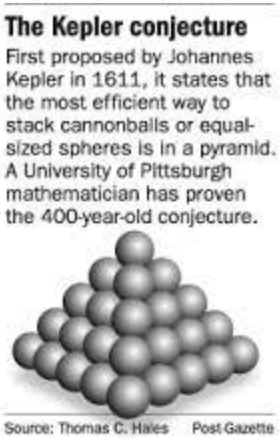

Simple v.s. Modern Type Theories

- Church's simple type theory (1940)
  - As in Montague semantics
  - Types ("single entity-sort": **e**, **t**, **e**→**t**); HOL/predicates
- Modern type theories
  - Many types of entities – "many-sorted"
    - Human, Table, Σx:Man.handsome(x), $Evt_T(t)$, Phy•Info, …
    - Dependent types, inductive types, universes, …
  - Examples of MTTs:
    - Predicative [non-standard FOL]: MLTT (Martin-Löf 1984)
    - Impredicative [HOL]: pCIC (Coq manual) and UTT (Luo 1994)

An episode: MTT-based technology and applications

- Proof technology based on type theories
  - Proof assistants
    - MTT-based: ALF/Agda, Coq, Lean, Lego, NuPRL, Plastic, …
    - HOL-based: Isabelle, Isabelle-HOL, …
- Applications of proof assistants
  - Math: formalisation of mathematics – eg,
    - 4-colour theorem (on map colouring) in Coq
    - Kepler conjecture (on sphere packing) in Isabelle/HOL
  - Computer Science:
    - Program verification and advanced programming
  - Computational Linguistics
    - NL reasoning based on MTT-semantics

      (In Coq: Chatzikyriakidis & Luo 2014/2016/2020; Luo 2023)



Type theories as foundational languages

- Type theories as foundational languages for
  - Maths: classical/Church's STT and constructive/Martin-Löf's
  - NL semantics: Montague semantics and MTT-semantics
- A comment – what typing is not:
  - "a : A" is not a logical formula (A is not a predicate).
    - j : **e**;   ugly(j) : **t**;   7 : Nat;   j : Human;   …
    - 7:Nat/j:Human are different from formulae nat(7)/human(j), where nat/human are predicates.
  - "a : A" is different from "a∈S" (the latter is a logical formula).
- What typing is related to (some example notions):
  - Meaningfulness (ill-typed ➔ meaningless)
  - Semantic/category errors (eg, "A table talks." – later)
  - Type presuppositions (Asher 2011)

MTT-semantics

- MTT-semantics
  - Formal semantics in modern type theories (MTTs), not simple TT
  - Has both model-/proof-theoretic characteristics. (Luo 2014)
- Development of MTT-semantics
  - Early use of dependent type theory in formal semantics:
    - Mönnich 1985, Sundholm 1986, Ranta 1994
  - Development since 2009 – full-scale alternative to Montague semantics
    - Z. Luo. Modern Type Theories: Their Development & Applications. Tsinghua Univ Press. 2023. (Monograph on MTTs with chapters on MTT-semantics, in Chinese)
    - S. Chatzikyriakidis & Z. Luo. Formal Semantics in Modern Type Theories. Wiley/ISTE. 2020. (Monograph on MTT-semantics)
    - S. Chatzikyriakidis & Z. Luo (eds.) Modern Perspectives in Type Theoretical Semantics. Springer, 2017. (Collection of papers on rich typing in NL semantics)

Course Plan

- Motivations of the course
- This lecture (Lecture I):
  - MTT-semantics: intro & case study (adjectival modification)
  - Introductory overview of the topics in Lectures II - IV
- Several traditionally "advanced" topics
  - Lect II: Events (Davidsonian ➔ dependent event types)
  - Lect III: Anaphora (Russellian/dynamic ➔ type-theoretic solution)
  - Lect IV: Copredication (dot-types ➔ formalisation in MTTs)
  - Lect V: More + analysis (e.g., dependent CGs, …; "open")

Each: history/Montague/MTT-semantics (informal & understandable)

Material and References

- Material available on the ESSLLI23 course web link:
  - Lecture slides for the first lecture (Lecture I)
  - Course proposal (good summary, but the organisation and descriptions of lectures are slightly different.)
- The slides for all 5 lectures, and the course proposal, will be available at

  https://www.cs.rhul.ac.uk/home/zhaohui/lecture-notes.html

  with references (cited in lectures) listed in the end of the slides.
- Papers/books on MTT-semantics available at

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

I.1. Introduction to MTT-semantics

Dependent types

- MTTs are dependent type theories – what's a dependent type?
- What is not a dependent type:
  - A dependent type is not a type dependent on types.
    - E.g., List(A) depends on types A and is not a dependent type.
- A dependent type is a type dependent on objects.
  - Parent(x) – it depends on objects x : Human.
  - Event ➔ Evt(h) with h:Human (events performed by h)
  - Π-types of dependent functions (see next page)
- Dependent types give, among other things:
  - Logical quantifiers (e.g., Π for $\forall$) in a propositions-as-types logic
  - Powerful tools in semantic construction (eg, Π-polymorphism)

Π-types: a taste of dependent types

- Πx:Human.Parent(x)
  - Type of functions, where Parent(x) is the type of x's parents.
  - f : Πx:Human.Parent(x), then
    f(h) : Parent(h), for h : Human.
- A→Prop (i.e., Πx:A.Prop)
  - Type of predicates over type A
- Π-polymorphism
  - small : ΠA:CN.(A→Prop)
  - small(Elephant) : Elephant→Prop
  - small(Mouse) : Mouse→Prop
  - small(Table) : Table→Prop

$$\frac{\Gamma \vdash A\ type \quad \Gamma,\ x{:}A \vdash B\ type}{\Gamma \vdash \Pi x{:}A.B\ type}$$

$$\frac{\Gamma,\ x{:}A \vdash b : B}{\Gamma \vdash \lambda x{:}A.b : \Pi x{:}A.B}$$

$$\frac{\Gamma \vdash f : \Pi x{:}A.B \quad \Gamma \vdash a : A}{\Gamma \vdash f(a) : [a/x]B}$$

$$\frac{\Gamma,\ x{:}A \vdash b : B \quad \Gamma \vdash a : A}{\Gamma \vdash (\lambda x{:}A.b)(a) = [a/x]b : [a/x]B}$$

Montague semantics and MTT-semantics

- Two basic semantic types in MG/MTT-semantics

Category	MG's type	MTT-semantic type
S (sentence)	t	Prop
IV (verb)	e→t	A→Prop (A: "meaningful domain")

Simple example

- John talks.
  - Sentences are (interpreted as) logical propositions.
  - Individuals are entities or objects in certain domains.
  - Verbs are predicates over entities or certain domains.

	Montague	**MTT-semantics**
john	**e**	Human
talk	**e→t**	Human→Prop
talk(john)	**t**	Prop

Selection Restriction

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

CNs as types and subtyping

- MTTs have many types (informally, collections).
  - Dependent types (Π-types, Σ-types, …)
  - Inductive types (Nat, Fin(n), …)
  - And more … (universes, logical types, …)

  Some can be used to represent CNs. (Ranta 1994, Luo 2012)
- Subtyping (necessary for multi-type languages such as MTTs)
  - Example: What if John is a man in "John talks"?
    - john : Man and talk : Human→Prop
    - talk(john)? (john is not of type Human …?)
  - Problem solved if Man ≤ Human
  - Coercive subtyping – adequate for MTTs (Luo 1997, Luo, Soloviev & Xue 2012)

Adjectival modification of CNs – case study

- A traditional classification
  - Kamp 1975, Parsons 1970, Clark 1970, Montague 1970

classification	**property**	**example**
Intersective	Adj(N) ➔ Adj & N	handsome man
Subsectional	Adj(N) ➔ N	large mouse
Privative	Adj(N) ➔ $\neg$N	fake gun
Non-committal	Adj(N) ➔ ?	alleged criminal

Intersective adjectives

- Example: handsome man (see next page for Σ-types)

	Montague	**MTT-semantics**
man	man : **e→t**	Man : Type
handsome	handsome : **e→t**	Man→Prop
handsome man	λx. man(x) & handsome(x)	Σ(Man,handsome)

- In general:

	Montague	**MTT-semantics**
CNs	predicates	types
Adjectives	predicates	simple predicates
CNs modified by intersective adj	Predicate by conjunction	Σ-type

Σ-types

- An extension of the product types A x B of pairs
- Σ-types of "dependent pairs"
  - Σ(A,B) of (a,b) for a:A & b:B(a)
- Rules for Σ-types:
  - Σ(A,B) also written as Σx:A.B(x)
- Examples:
  - Σ(Human,dog)

    with dog(j)={d}, dog(m)=∅, …
  - Σ(Man,handsome)

$$(\Sigma)\quad \frac{\Gamma \vdash A\ \mathit{type} \quad \Gamma,\ x : A \vdash B\ \mathit{type}}{\Gamma \vdash \Sigma x : A.B\ \mathit{type}}$$

$$(\mathit{pair})\quad \frac{\Gamma \vdash a : A \quad \Gamma \vdash b : [a/x]B \quad \Gamma, x : A \vdash B\ \mathit{type}}{\Gamma \vdash (a, b) : \Sigma x : A.B}$$

$$(\pi_1)\quad \frac{\Gamma \vdash p : \Sigma x : A.B}{\Gamma \vdash \pi_1(p) : A}$$

$$(\pi_2)\quad \frac{\Gamma \vdash p : \Sigma x : A.B}{\Gamma \vdash \pi_2(p) : [\pi_1(p)/x]B}$$

$$(\mathit{proj}_1)\quad \frac{\Gamma \vdash a : A \quad \Gamma \vdash b : [a/x]B \quad \Gamma, x : A \vdash B\ \mathit{type}}{\Gamma \vdash \pi_1(a, b) = a : A}$$

$$(\mathit{proj}_2)\quad \frac{\Gamma \vdash a : A \quad \Gamma \vdash b : [a/x]B \quad \Gamma, x : A \vdash B\ \mathit{type}}{\Gamma \vdash \pi_2(a, b) = b : [a/x]B}$$

- An adjective maps CNs to CNs:
  - In MG, predicates to predicates.
  - In MTT-semantics, types to types.
- MTT-semantics (Chatzikyriakidis & Luo 2020, Luo 2023)

classification	**example**	**types employed**
Intersective	handsome man	Σ-types with simple predicates
Subsectional	large mouse	Π-polymorphic predicates and Σ-types
Privative	fake gun	Disjoint union types with Π/Σ-types
Non-committal	alleged criminal	special predicates

I.2. Introductory Overview of "Advanced" Topics

Note:

For each topic/lecture, I shall try to cover

- History/examples in introduction
- Montague or traditional approaches
- Type-theoretical approaches

and informal/understandable.

Lecture II: Events (overview)

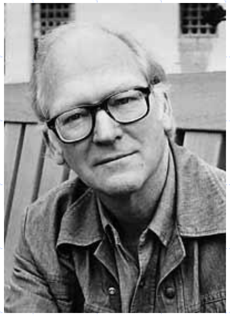
- Davidsonian event semantics
  - Original motivation: adverbial modifications

    (1) John buttered the toast.

    (2) John buttered the toast with the knife in the kitchen.

    Do we have (2) $\Rightarrow$ (1)?

    - Cumbersome in MG with meaning postulates
    - Davidson (1967): verbs tacitly introduce existentially quantified events.



- Neo-Davidsonian notation with thematic roles (eg, Parsons 1990):

  (1′) ∃v:Event. butter(v) & agent(v)=john & patient(v)=toast

  (2′) ∃v:Event. butter(v) & agent(v)=john & patient(v)=toast & with(v,knife) & at(v,kitchen)

  Obviously, (2′) $\Rightarrow$ (1′)

- Event semantics is an extremely popular topic
  - Casati and Varzi. Events: An Annotated Bibliography. 1997. [25 years ago, already 235 pages!]
  - Some researchers even take it for granted …
- But:
  - What is an event?
  - Are events atomic? Structured? If the latter, how?
  - Is the introduction of events completely harmless?
  - … …

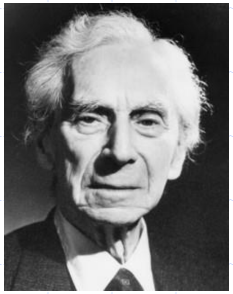
still unsettled/debated/…

Dependent event types

- Dependent event types (DETs)
  - Events are classified into "event types", dependent on (or classified according to) thematic roles.
- Focus: Montague (simple type theory) + DETs
  - Rather than MTT+DETs (stepping-stone for easier understanding)
- Several application examples
  - Solve problems such as "Event Quantification Problem"
  - Facilitate semantic constructions of tensed sentences
  - Restore "selection restriction" in MTT-event semantics

See (Luo & Soloviev 2017, Chatzikyriakidis & Luo 2020, Luo 2023).

Lecture III. Indefinites & Anaphora (overview)



- We'll discuss indefinites like "a man". Are they
  - Quantifier phrases (as Russell suggests)?
  - Referring expressions?
- Russell (1919): the ∃-view
  - A man came in. ➔ $\exists x$:**e**. man(x)∧come_in(x)
  - A lot of arguments/examples for the ∃-view.
- But what about, for example,
  - <u>A man</u> came in. <u>He</u> lit a cigarette. [compositional semantics?]
  - Every farmer who owns <u>a donkey</u> beats <u>it</u>.

$$(\#)\ \forall x.\ [farmer(x)\ \&\ \exists y.(donkey(y)\ \&\ own(x,y))] \Rightarrow beat(x,y)$$

Both referring to a variable outside its scope (e.g. the last "y").

Dynamic semantics

- Dynamic approaches (widely accepted for anaphora treatment)
  - Discourse Representation Theory (Kamp 1981, Heim 1982)
  - Dynamic Predicate Logic (Groenendijk and Stokhof 1991)

$$\boxed{(\exists x \varphi);\psi} \Leftrightarrow ([x];\varphi);\psi \Leftrightarrow [x];(\varphi;\psi) \Leftrightarrow \boxed{\exists x(\varphi;\psi)}$$

where ";" is the dynamic conjunction (so, previous "y" would be OK …)

- However, logics in dynamic semantics are non-standard.
  - For example, DPL (G&S91) is rather non-standard:
    - non-monotonic, irreflexive/intransitive entailment, …
  - Substantial changes required for underlying logic(s) in semantics
- Two "extremes"? Anything in the middle?

Russell (∃) |------------?------------| Dynamic

Type-theoretical approaches

- Using dependent types (Mönnich 1985, Sundholm 1986)
  - (Donkey) Every farmer who owns a donkey beats it.
  - $\forall z : F_\Sigma.\ beat(\pi_1(z), \pi_1(\pi_2(z)))$ where $F_\Sigma = \Sigma x{:}F\ \Sigma y{:}D.\ own(x,y)$
  - Σ is the "strong sum" with two projections π_1 and π_2 .
- This gives a compromise:
  - Σ is "strong" so that witnesses can be referred to outside its scope.
  - The change for underlying logic is much less substantial.
- However, a problem – Σ plays a double role:
  - Subset constructor (1^{st} Σ) and existential quantifier (2^{nd} Σ).
  - But this is problematic ➔ counting problem.
  - A satisfactory solution with both strong/weak sums (Luo 2021)

Lecture IV. Copredication (overview)

- Copredication is a special case of logical polysemy.
  - See (Pustejovsky 1995, Asher 2011), among others.
- Examples
  - (*) The lunch was delicious but took a long time.
    - delicious : Food→**t**; take_long_time : Process→**t**
    - Their domains Food/Process do not share any common objects, but they can both apply to the same noun (lunch) …
  - (**) All three books are heavy and boring.
    - heavy : Phy→**t**; boring : Info→**t**
    - Phy/Info (similar to the above) and heavy/boring can both apply to a book.

How to analyse it formally?

- Very interesting issue
  - Easy to understand, but intriguing (nice research topic)
  - Numerous papers in the literature
- Many approaches, including (just to name a few):
  - Dot-types and related approaches
    - E.g., Pustejovsky 95, Asher 2011, Luo 2010, …
  - Mereological approaches
    - E.g., Gotham 2014, 2017
  - Others
    - E.g., Retoré 2013, Liebesman & Magidor 2023, …

Dot-types in MTTs

- Dot-types – idea by Pustejovsky (1995)
  - Objects of type A•B have two aspects: being both A and B.
  - Informally, the above sentences (*)/(**) can now be interpreted.
- How to
  - formalise dot-types?
  - formalise dot-types in MTTs?
  - We'll try to explain them informally – see (Luo 2010, 2023)
- What happens when copredication interacts with ...?
  - Interacting with quantification ➔ identity criteria of CNs (Luo 2012)
  - See (Chatzikyriakidis and Luo 2018, Luo 2023)

Lecture V. Reasoning, CGs, and Beyond (overview)

- More MTT-related topic(s) may be briefly introduced; examples include:
  - Natural language reasoning based on MTT-semantics in "proof assistants" – computer-assisted reasoning systems; c.f. (Chatzikyriakidis and Luo 2020, Chap 6; Luo 2023, Chap 5)
  - Dependent types in categorial grammars – substructural dependent type theory; c.f. (Luo 2023, Sect 4.5)
- This lecture is intentionally left as "open" at the moment; besides the above, it may also include some "tidying up" of previous lectures.

NL Reasoning in Proof Assistants

- Interactive theorem proving based on MTTs
- An ITP system consists of three parts for:
  (1) contextual defns (2) proof development (3) proof checking

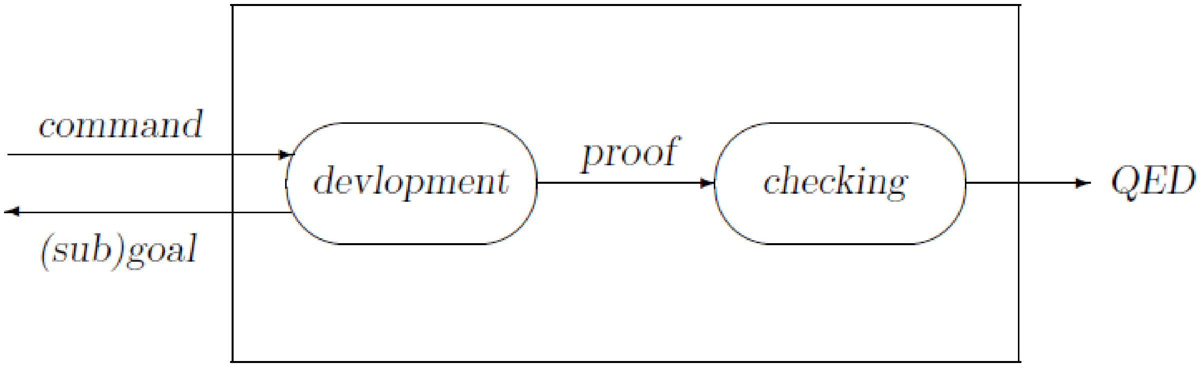


Figure 1: Interactive proof development and proof checking

Dependent Categorial Grammar

- Categorial Grammars (or typelogical grammars)
  - An approach to syntactic analysis
  - CGs are based on substructural logics
    - Moortgat: 'Typelogical grammars are substructural logics, designed for reasoning about the composition of form and meaning in natural language.' (Stanford Encyclopedia of Philosophy, 2010)
- What is a substructural logic?
  - In a proof system, there are usually three kinds of "structural" rules: weakening, contraction (strengthening), exchange
  - Weakening: adding more assumptions is OK.
  - Contraction: removing a repeated assumption is OK.
  - Exchange: swapping the order of two assumptions is OK.

  In substructural (resource-sensitive) logics, the above may not be OK.

Lambek calculus and beyond

- Categorial grammar and historical developments:
  - Ajdukiewicz, Bar, Hillel, …
- Lambek calculus (1958)
  - Ordered formulae B/A and A\B
    - John runs – "run applies to a np on the left".
      John : NP and run : NP\S
  - Resource sensitive – substructural (eg, no exchange – word order)
- Linear/hybrid CGs (Oehrle 1994, Kubota & Levine's HTLG, …)
- Substructural type theory (Luo 2023)

	Π-type	Non-dependent type	Abstraction	Application
Linear	$\overline{\Pi} x{:}A.B$	$A \multimap B$	$\overline{\lambda} x{:}A.b$	$\overline{app}(f,a)$
Ordered (right)	$\Pi^r x{:}A.B$	B/A	$\lambda^r x{:}A.b$	$app^r(f,a)$
Ordered (left)	$\Pi^l x{:}A.B$	$A \setminus B$	$\lambda^l x{:}A.b$	$app^l(a,f)$

Table 1 Three substructural function types in $\bar{\lambda}_{\Pi}$: summary of notations.