Modern Type Theories and Their Applications [现代类型论及其应用]

Zhaohui Luo [罗朝晖]
Royal Holloway, Univ of London
Zhaohui.Luo@hotmail.co.uk
https://www.cs.rhul.ac.uk/home/zhaohui/

This talk – two parts

I. Modern Type Theories (MTTs) [现代类型论]

- Basics and intuitive understanding (in comparison with set theory)
- Brief introductions to several applications of MTTs

II. Formal semantics in MTTs (MTT-semantics) [现代类型论语义学]

- Intro. to MTT-semantics (basics & advanced topics, c.f. Montague)
- Philosophical Foundation (if time permits)

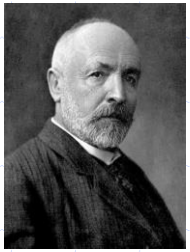
Studying type theory, I've collaborated with many people (only mentioning a few):

- Adams, Callaghan, Goguen, Pollack (type theory & proof assistants)
- Soloviev, Xue, Y. Luo, Lungu, Bradley (coercive subtyping)
- Asher, Chatzikyriakidis, Maclean, Shi (MTT-semantics)

Part I. Modern Type Theories [现代类型论]

I.1 Type theory: development and basics

- Foundations of mathematics and paradoxes
  - Naïve set theory (Cantor)
  - Paradox (Russell 1903) ➔ logical inconsistency
  - Crisis in the foundation of mathematics
- Set theory by Zermelo
  - Axiomatic set theory (1908; later ZFC etc.)
  - Widely accepted foundations in math community
- Type theory by Russell
  - Ramified type theory (*Principia Math.* 1910-13, 1925)
  - Vicious circle principle ("impredicativity" like ∀X.X)
  - Ramified hierarchy – problematic "axiom of reducibility"



Simple type theory

- Ramsey (1926)
  - Logical v.s. semantic paradoxes
  - Russell's paradox v.s. Liar's paradox
  - Impredicativity is circular, but not vicious
  - So, Russell's ramified TT can be "simplified" to simple TT.

- Church's simple type theory (STT; 1940)
  - Formal system based on λ-calculus
  - Types as in ramified TT (e, t, $e\to t$, …)
  - Higher-order logic (formulas like $\forall X.X$)
  - Wide applications (Montague semantics, proof assistants, …)

Note: "Simple" could have another meaning: only "simple" types …

Modern Type Theories

- Martin-Löf has introduced/employed
  - Judgements, contexts, definitional equality
  - Dependent/inductive types, type universes
  - Curry-Howard principle of propositions-as-types

- Examples of MTTs [& implementing proof assistants]:
  - Predicative TTs:
    - MLTT – Martin-Löf's type theory [1975]; Agda
  - Impredicative TTs:
    - CC [Coquand & Huet 1988] and pCIC; Coq/Lean
    - UTT [Luo 1990, 1994]; Lego/Plastic

UTT [Luo 1989, 1994, 罗2024] – an example MTT

- UTT – Unifying Theory of dependent Types
  - UTT = MLTT + CC (formally)

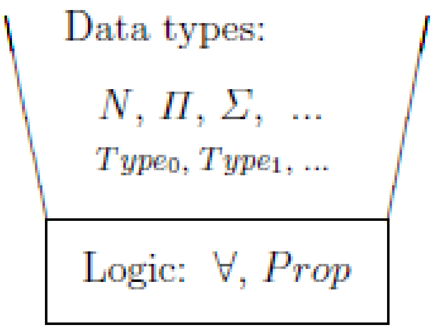
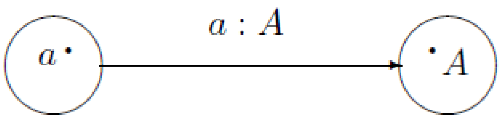


Fig. 1. The type structure in UTT.

  - UTT has nice meta-theoretic properties (Goguen's 1994 PhD)
  - Strong normalisation (➔ consistency) etc.
- UTT "extends" STT as in Montague semantics (Prop ~ "t")

Judgements – basic notion in type theory

- Membership judgement
  - a : A – a is an object of type A.



- What is A? A can be: [see next 3 slides]
  - data type: Nat (inductive type), $\Pi x{:}A.B(x)$ (dependent type)
  - propositional type: $A \wedge B$, $\forall x{:}A.P$
  - type universe: a type of some other types
- Comparison with set theory: [see slide]
  - Judgement "a : A" is not a logical formula
  - Different from "$s \in S$", which is a formula (say in FOL)
  - A logic is only a part of type theory ("propositional types")

Example of inductive type – Nat of natural numbers

- Peano axioms for nats [predicate N; n∈N ~ N(n)]

$$(P1)\ 0 \in N$$

$$(P2)\ \forall x.\ x \in N \Rightarrow succ(x) \in N$$

$$(P3)\ \forall x, y.\ x, y \in N \wedge succ(x) = succ(y) \Rightarrow x = y$$

$$(P4)\ \forall x.\ x \in N \Rightarrow 0 \neq succ(x)$$

$$(P5)\ \forall P.\ P(0) \wedge [\forall x.\ x \in N \wedge P(x) \Rightarrow P(succ(x))] \Rightarrow \forall z.\ z \in N \Rightarrow P(z)$$

- Martin-Löf’s idea – type rather than set
  - Inductive types as “computational theories”
  - Example – Nat, the type of natural numbers (next slide)

Rules for Nat

- Formation and introduction rules

$$\frac{}{Nat\ type} \qquad \frac{}{0 : Nat} \qquad \frac{n : Nat}{succ(n) : Nat}$$

This specifies canonical objects 0 and succ(n). [规范对象]

- Elimination rule

$$\frac{\begin{array}{l}\Gamma, z : Nat \vdash C(z)\ type \quad \Gamma \vdash n : Nat \\ \Gamma \vdash c : C(0) \quad \Gamma, x : Nat, y : C(x) \vdash f(x,y) : C(succ(x))\end{array}}{\Gamma \vdash \mathcal{E}_{Nat}(c, f, n) : C(n)}$$

This gives induction and (with computation rule) primitive recursion:

$$\begin{array}{rcl}\mathcal{E}_{Nat}(c, f, 0) & = & c \\ \mathcal{E}_{Nat}(c, f, succ(n)) & = & f(n, \mathcal{E}_{Nat}(c, f, n))\end{array}$$

- All Peano axioms are either rules or provable.

Π-types and ∀-props: examples of dependent types

- $\Pi x{:}A.B(x)$ – dependent function type
  - Type for collection

    $\{\ f \in A \to \bigcup_{a \in A} B(a) \mid \forall a \in A.\ f(a) \in B(a)\ \}$
  - f : $\Pi x{:}Human.Parent(x)$

    ➔ f(h) is father/mother of h (not others!)

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

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

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

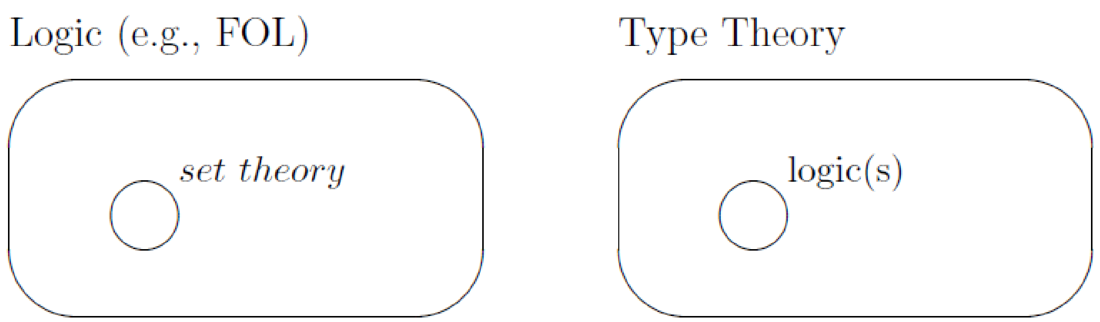
$$\frac{\Gamma,\ x{:}A \vdash b : B \quad \Gamma \vdash a : A}{\Gamma \vdash (\lambda x{:}A.b)(a) = [a/x]b : [a/x]B}$$

- Universal quantification

$$\frac{\Gamma \vdash A\ type \quad \Gamma,\ x{:}A \vdash P : Prop}{\Gamma \vdash \forall x{:}A.B : Prop}$$

  - Prop, the collection of propositions, is a type itself [impredicative universe with "circular" props like $\forall X{:}Prop.X$]
  - Propositions-as-types: propositions are (some) types [So, logic(s) is only a part of type theory – see next slide.]

Relationship between logic and set/type theory



Set theory is a theory in logic — *Logic is a part of type theory*

Note: A logic in type theory can be

(1) PaT logic in MLTT, (2) HOL in UTT, or (3) h-logic in HoTT.

Type theory is more powerful than set theory in manipulating logical exprs!

For example, are events necessary in formal semantics? (cf, Davidson 1967)

-- Variable polyadicity (VP) [论元数量可变性] ➔ Davidsonian event semantics

-- VP can be achieved in type theory. Events unnecessary after all!? (Luo & Shi 2025)

I.2 Some (selected) applications of MTTs

(1*) Proof technology (proof assistants)
(2*) Univalent foundations of mathematics
(3) Formalisation of math and prog verification
(4) Natural language reasoning
… …

Interactive proof tech. based on type theories

- Proof assistants – interactive theorem proving
  - MTT-based: Agda, Coq, Lean, Lego, NuPRL, Plastic, …
  - HOL-based: HOL, Isabelle, …
- An ITP system consists of three parts for:
  (1) contextual defns (2) proof development (3) proof checking

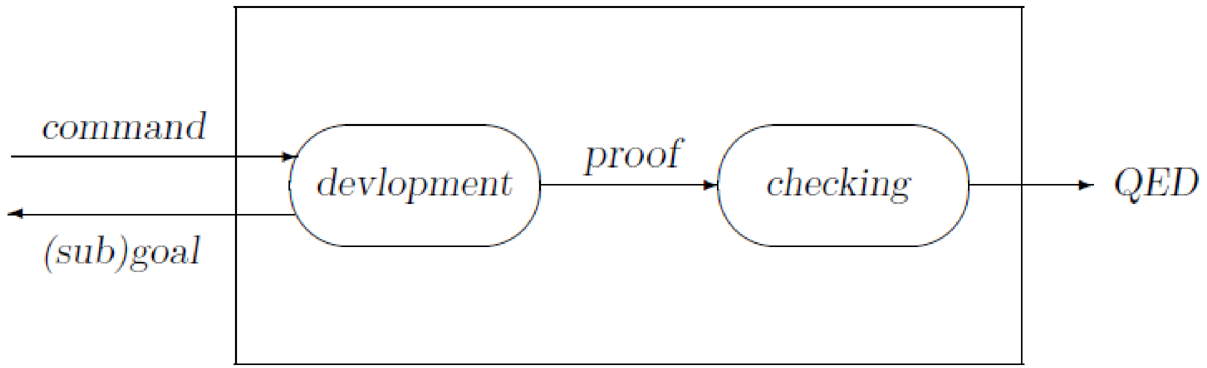
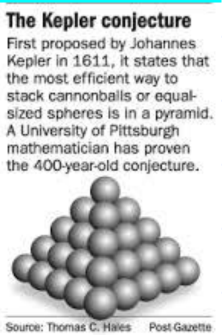


Figure 1: Interactive proof development and proof checking

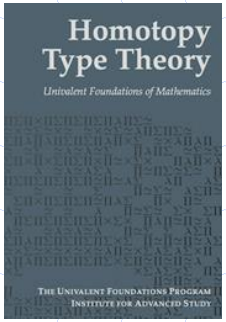
Applications of proof assistants

- Formalisation of mathematics
  - 4-colour theorem (Coq)
  - Kepler conjecture (Isabelle)
- Computer Science:
  - program verification
  - advanced programming
- Computational Linguistics
  - NL reasoning (Coq) based on MTT-semantics (Chatzikyriakidis & Luo 2014/2016/2020; 罗2024)



Univalent foundations & homotopy type theory

- Univalent Foundations [单价基础] of mathematics (new alternative to set theory)
  - V. Voevodsky, Russian Fields medalist (1966–2017)
  - Key motives: proof-checking & isomorphic invariance
  - Neither feasible in set theory, but OK in type theory!



- Homotopy type theory (HoTT 2013) [同伦类型论]
  - UF special year 2012-13 at Inst for Adv Study, Princeton
  - Formalisation of UF: HoTT = MLTT + UA + HITs
    - UA – univalence axiom [单价公理] (for iso. invariance)
    - HITs – higher inductive types [高等归纳类型] (generalisation of h-logic)
- UF Mathematics in proof assistants (Agda/Coq/Lean)

Research monograph on MTTs in Chinese

罗朝晖：现代类型论的发展与应用。
清华大学出版社，2024年。

Z. Luo. Modern Type Theories: Their Development and Applications. Tsinghua Univ Press, 2024. (In Chinese)

网址：http://www.tup.tsinghua.edu.cn/booksCenter/book_09109701.html

Part II. MTT-semantics [现代类型论语义学]

1. Basic and advanced (selected; c.f., Montague)
2. Philosophical Foundation (proof-theoretic sem)

II.1. Formal Semantics and MTT-semantics

- Montague semantics (Montague 1930–1971)
  - MG: in set theory (simple type theory as intermediate)
  - Dominating in linguistic semantics since 1970s
  - Studies in China: 邹崇理、陈波、王路、张建军 … …

- MTT-semantics: formal semantics in modern type theories
  - Ranta (1994): formal semantics in Martin-Löf's type theory
  - Recent study on MTT-semantics ➔ full-scale alternative to MG
    - Z. Luo. Formal Semantics in MTTs with Coercive Subtyping. L&P 35(6). 2012.
    - S. Chatzikyriakidis & Z. Luo. Formal Sem in MTTs. Wiley, 2020. [M on MTT-sem]
    - 罗朝晖. 现代类型论的发展与应用. 清华大学出版社，2024. [第3、4章]
  - Rich typing for formal sem – many researchers (eg, papers below)
    - S. Chatzikyriakidis & Z. Luo (eds) Modern Perspectives in TT Sem. Springer, 2017.

Simple example

- John talks.
  - Sentences are propositions & verbs predicates (in domains).
  - Individuals are entities (or in certain domains).

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

  - In MTT-sem, common nouns are <u>types</u> (Munich, Sundholm 1985) rather than <u>predicates</u> as in Montague sem (next page).

CNs as types [通名为类型]

- Meaningfulness – is the following meaningful?

  (*) The table talks.

- Yes in MG: (*) has a truth value.
  - talk(the-table) is false in the intended model.
  - But, unlike (*), meaningful sentences can also be false!
  - So, meaningless and meaningful sentences are not distinguished …
- No in MTT-semantics: (*) is meaningless (ill-typed).
  - "the-table : Human" is incorrect.
  - talk(the-table) is ill-typed (talk : Human→Prop).
- In MTT-sem, meaningfulness = well-typedness (cf, Asher 11)
  - MTT-semantics gives a better basis for selection restriction [选择限制].
  - Note: in MTTs, type-checking is decidable. (Note: truth/falsity is not decidable.)

Rich type structures – case study in adjectival mod.

- An adjective maps CNs to CNs:
  - In MG, predicates to predicates.
  - In MTT-semantics, types to types.
- MTT-semantics (Chatzikyriakidis & Luo 2020, 罗 2024)

classification	example	types employed
intersective [相交]	black cat	Σ-types with simple predicates
subsectional[下属]	small elephant	Π-polymorphic predicates & Σ-types
privative [否定性]	fake gun	disjoint union types with Π/Σ-types
non-committal [非承诺性]	alleged murderer [被指控的凶手]	special predicates

Subtyping [子类型] in MTT-semantics

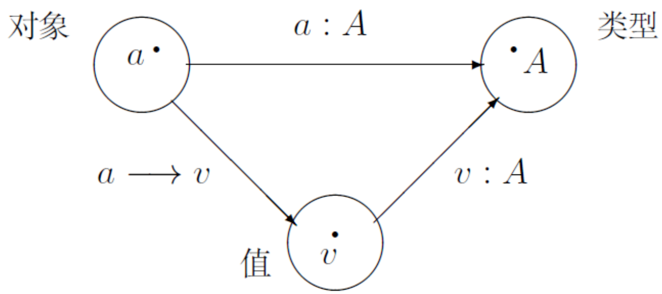
- Simple example
  - A human talks. Paul is a handsome man. Does Paul talk?
  - Semantically, can we type talk(p)?
    - talk : Human→Prop and p : [handsome man]
  - Yes, because p : [handsome man] ≤ Man ≤ Human
- Subtyping is crucial for MTT-semantics
  - Coercive subtyping [Luo 1999, Luo, Soloviev & Xue 2012] is adequate for MTTs and we use it in MTT-semantics.
  - Note: Traditional subsumptive subtyping is inadequate for MTTs (canonicity fails with subsumptive subtyping).

Advanced topics in MTT-semantics: examples

- **Copredication [同谓] & dot-types** [L 09, Xue & L 12, Chatzikyriakidis & L 18]
  - 午餐很可口，但花了很长时间。(Food • Process – such dot-types formalised in MTTs.)
- Linguistic coercions via coercive subtyping [Asher & L 12]
- Signatures for linguistic contexts [L 14, Lungu & L 16]
- **MTT event semantics (dependent event types)** [L & Soloviev 17]
  - 张三大声地说话。($\exists e.\ talk(e) \wedge loudly(e) \wedge ag(e)=z \Rightarrow \exists e{:}Evt_A(z).\ talk(e) \wedge loudly(e)$.)
- Propositional forms of judgemental inter. [Xue, L & Chatzikyriakidis 18, 23)]
- MTT-semantics in $MLTT_h$ [L 18]
- Subtype universes [Maclean & L 20, Bradley & L 23, 罗24]
- Dependent categorial grammar [罗24]
- **CNs as setoids** [类型体] [L 12, Chatzikyriakidis & L 18] (CNs as HITs [L 24])
  - "CNs-as-types", but (A,φ) in general. (Use HITs to represent "quotient types", so OK.)
- Type-theoretic analysis of event semantics [L & Shi 25]

II.2. Philosophical foundation of MTT-semantics

- Meaning explanation



Example: A = Nat, a = 3+4, v = 7.

How to guarantee that computation a→v terminates !?

- Foundational researches (c.f., type theory ~~ FOL)
  - Meta-theoretic study [元理论] (eg, strong normalisation of UTT)
  - Meaning-theoretic harmony [意义理论之和谐性] between intro/elim rules

Meta-theory

- Meta-theory of type theories
  - Computation is central.
    - Strong normalisation: All computations terminate.
    - This usually implies canonicity and logical consistency.
  - Sophisticated, tedious and rather hard to do
    - Numerous theorems/lemmas/concepts/… [examples in next slide]
  - ECC/UTT's meta-theoretic studies [Luo 1990, Goguen 1994]
- Caveat:
  - Meta-theory depends on consistency of meta-language (set theory – believed to be OK, but …)
  - Desire/wish: can we argue for "correctness" directly? (pre-mathematical meaning theory)

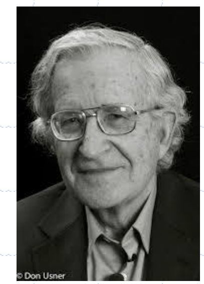
Meta-theoretic theorems: examples

- Church-Rosser theorem (CR) [CR定理]
  - If a=b : A, then there exists c : A s.t. a → c and b → c.
- Subject Reduction (SR) [主题归约]
  - If a : A and a → b, then b : A.
- Strong Normalisation (SN) [强正规化]
  - Every computation from a well-typed term terminates.
- Decidability (of type-checking) [（类型检测的）可判定性]
  - It is decidable whether a judgement is correct (derivable).
- Logical consistency [逻辑相容性]
  - Theorem (of UTT). ∀X:Prop.X (false) is not provable (in the empty context).
  - Proof. By contradiction. Assume M : ∀X:Prop.X be in normal form (by SN & SR).
    Then, analysis would imply Prop = X or Prop = Qx:A.B which, by CR, is impossible.
    Therefore, M does not exist (hence, ∀X:Prop.X is not inhabited).

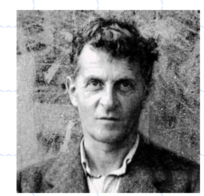
Theories of meaning

- Meaning is reference (model-theoretic）
  - Word meanings are (abstract/concrete) objects.
  - c.f., platonism: Frege, …

- 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 its uses.
  - c.f., Wittgenstein, …



Proof-theoretic semantics – use theory for logics

- Proof-theoretic semantics (PTS)
  - Use theory for logical systems
  - Dummett 76/91 (meaning) & Prawitz72 (general proof theory)
  - China: 张燕京06《达米特意义理论研究》、周志荣23/25、党学哲24 …
- Ideas
  - Pre-mathematical justification of logical rules (informally from "first principles", not meta-theoretically)
  - For logic: two aspects of use – verification and consequence
  - Harmony: intro/elim should be harmonious (research prob: higher-order)
- PTS for type theories
  - Martin-Löf's meaning explanations (1984)
  - Type theory potentially has PTS, while set theory does not.
  - Current research: hypothetical judgements; impredicativity

MTT-semantics is both model- & proof-theoretic

- Claim:

  MTT-semantics is both model-theoretic and proof-theoretic.

  Natural Language —(1)— Modern TT —(2)— Meaning Theory

  (1) representational, model-theoretic

  (2) inferential roles, proof-theoretic (in sense of use theory)

- Two papers (among others)
  - Z. Luo. Formal semantics in MTTs: Is it model-theoretic, proof-theoretic, or both? Invited talk at LACL, LNCS 8535, pp177-188. Toulouse, 2014.
  - 罗朝晖, 石运宝. 现代类型论语义学的哲学基础.《逻辑、智能与哲学》, 2025.

Why important for MTT-semantics?

- 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 proof technology: proof assistants
  - Applications to NL reasoning
- Leading to both
  - Wide-range modelling as in model-theoretic semantics
  - Effective inference based on proof-theoretic semantics

Remark: new perspective & possibility not available before!