A Pluralist Approach to Type-Theoretic Foundations

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

This talk

- Types and logical propositions in type theory
  - Relationship: identification/separation…?
- Framework of Logic-enriched Type Theories
  - Separations of types/propositions in LTTs
- Type theories classified by **P**: totality of props
  - "constrained genericity"

I. Types v.s. Propositions

- Typing is independent of logic/inference.
  - In traditional logic, logical propositions are different from "data types" (orthogonal?)
    - Two worlds: objects in the "real world" v.s. their properties
  - Only since development of modern type theories (say, Martin-Löf's TT), the relationship becomes an issue: Should they be identified or separated?
  - The independence viewpoint suggests the latter.
- *Separation* allows/supports *pluralism*.

Foundational Pluralism

- Two extreme positions in FOM
  - Neo-platonism (eg, set-theoretic foundation: Gödel/Maddy)
  - Revisionists (eg, intuitionism: Brouwer/Martin-Löf)
- A pragmatic position – "pluralism"
  - Various maths based on different logical foundations
  - Cf, "logical pluralism" (eg, Beall and Restall)
- Support in type theory and the associated tech?
  - Theorem proving technology based on TTs is not just for constructive reasoning.
  - Eg, Classical logic as well as intuitionistic logic

- Separation of propositions and types – how?

  Identification (MLTT)

  → "Half" separation in ECC/UTT/pCIC (cf, Coq)

  → Complete separation in LTTs

  (Logic-enriched Type Theories, Aczel & Gambino)

II. Framework for LTTs

LTTs = Logics + Types

- Logics – logical props/inference
- Types – inductive types + types of sets

```
Logic   Types
   \     /
    \   /
     LF
```

LTT-framework = LTTs specified in LF

- LF – Logical framework (cf, Edin LF, Martin-Löf's LF, PAL^+, …)
  - **P** in LF – totality of propositions, plus Prf(p) for p : **P**
  - Type in LF – totality (kind) of types, plus El(A) kind for A : Type
- Inductive types in LTTs (Aczel & Gambino 02/06)
- Typed sets in LTTs: sets with base types (see later)

(Luo 2006, LNCS 4435)

Type theories as LTTs: an example

- LTT_c: TT with first-order classical logic (FOL_c)
  - Types
    - Eg, inductive types like N, Σx:A.B, List(A), Tree(A), …
    - Eg, types of sets like Set(A)
  - Propositions:
    - Describing properties of objects (∀x:A.P(x) with type A)
    - Classical laws may be introduced
  - Induction rules
    - Linking the world of logical propositions and that of types
    - Enabling proofs of properties about objects of types

7

Propositions in LTT_c

- **P** <u>kind</u>, Prf(p) <u>kind</u> for p : **P**
- Implication (omitting Prf)
  - $\supset$: **P**→**P**→**P**
  - $\supset_I$: (p:**P**)(q:**P**) (p→q) → p⊃q
  - $\supset_E$: (p:**P**)(q:**P**) p → p⊃q → q
- Peirce's law (for classical logic)
  - peirce : (p:**P**)(q:**P**) ((p→q)→p) → p
- Remark: laws for other connectives can be introduced similarly; eg,
  - DN : (p:**P**) ¬¬p → p

8

Types in LTT_c: an example – natural numbers

- Formation and introduction
  - N : Type
  - 0 : N
  - succ(n) : N, for n : N
- Elimination over types and computation:
  - $Elim_T$(C,c,f,n) : C(n), for C(n) : Type
  - Plus computational rules for $Elim_T$: eg,
    - $Elim_T$(C,c,f,0) = c
    - $Elim_T$(C,c,f,succ(n)) = f(n,$Elim_T$(C,c,f,n))
- Induction over propositions:
  - $Elim_P$(P,c,f,n) : P(n), where P(n) : **P**
  - Key to prove logical properties of natural numbers
- Comment: canonicity of inductive types

9

Typed sets in LTT_c

- Typed sets
  - Set(A) : Type for A : Type
  - { x:A | P(x) } : Set(A)
    - t ∈ { x:A | P(x) } means P(t)
- Sets in LTT_c (predicative sets)
  - Universes of small types and small propositions
  - A must be *small* (in particular, A is not Set(…))
  - P(x) must be *small* (not allowing quantifications over sets)

10

Pluralism in type-theoretic foundations

- Consider the "combinations" of the following and their "negations":
  - (C) Classical logic
  - (I) Impredicative definitions

We would have
- (CI) Ordinary (classical, impredicative) math
  Simple type theory, HOL/Isabelle
- (C°I°) Predicative constructive math
  Martin-Löf's TT, Agda/NuPRL
- (C°I) Impredicative constructive math
  pCIC/ECC/UTT, Coq/Lego/Plastic
- (CI°) Predicative classical math (eg, Weyl, Simpson)
  LTT_w, Plastic

Uniform foundational framework for formalisation to support pluralism?

11

Formalisation of Weyl's predicative math

- H. Weyl. *The Continuum* (*Das Kontinuum*), 1918.
  - Historical development (paradox etc.)
  - Predicative development of the real number system
  - The notion of category
  - Classical logic
- Weyl/Feferman/Simpson's work on predicativity
  - Predicativity (E.g., { x | φ(x) } with φ being "arithmetical" – without quantification over sets)
- Formalisation of Weyl's book in Plastic
  - In LTT, use classical logic and predicative sets
  - Weyl's categories as types
  - "Exact match" (and further research …)

12

III. TTs classified by **P**: totality of props

- Martin-Löf's type theory
  - Propositions are identified with types.
  - **P** ≡ Type kind (**P** is the same as kind Type)
- Impredicative type theories with HOLs
  - Propositions constitute an internal totality/type.
  - **P** ≡ Prop : Type
    - **P** is an impredicative type universe
  - ECC/UTT (Lego) and pCIC (current Coq)

13

TTs classified by **P** (cont[ed])

- TTs enriched with first-order logics
  - Propositions are separated from types.
  - **P** kind (**P** is a kind, but different from Type)
  - TTs + induction rules
    - LTT_i: TT with FOL_i (Aczel & Gambino 02)
    - LTT_c: TT with FOL_c (Luo 06, as described earlier)
    - LTT_w (for Weyl's predicative math; Adams & Luo 10)
  - MTT (**P** ⊆ Type; Maietti & Sambin 05)

14

TTs with "extra axioms"

- Classical TTs
  - EM : **P** (various forms for various TTs)
  - LTT_c, $pCIC_c$, …
- Univalent Foundations

15

Univalent Foundations

- Univalent axioms
- Logic of mere propositions (or h-propositions)
  - isHProp(A) ≡ Π x,y:A. Id(A,x,y)
  - **P** ≡ $Prop_{U_0}$ ≡ Σ(U_0,isHProp)
  - **P** is a totality of props if we assume propositional resizing
- Propositional resizing:
  - The following inclusions are assumed to be equivalences (and hence the types are equal by UAs):
    - $Prop_{U_0} \rightarrow Prop_{U_1} \rightarrow Prop_{U_2} \rightarrow \ldots$
  - Higher order – quantification over props/predicates (eg: Π A:$Prop_{U_0}$. …)
  - To see whether A : $Prop_{U_0}$ requires one to prove isHProp(A) as well as A : U_0.

16

- Remark
  - Univalent universe U: equivalence → identity
  - A univalent universe cannot be formally closed
  - Elimination rule (induction) for a universe
    - Universe consisting of only those types generated by finitely many type formers
    - Cf. Nordström et al 1990, Palmgren (early 1990s), …
    - Incompatible with univalence

17

Concluding Remarks

- LTT-framework for various type theories
  - Formulations of LTTs (eg, LTT_c)
    (eg, Luo 2006, Adams & Luo 2010)
- Formulations of TTs via **P**
  - "Constrained genericity": between LF-based generic ITPs and single-logic based ITPs
  - Potential for effective implementation in proof assistant
- Typing is independent with logical inference!

18