

# Logics and mathematics in type theory

Zhaohui Luo

Dept of Computer Science

Royal Holloway, Univ of London

# Type theory and mathematics

## ◆ Current type theories

- ◆ Martin-Löf's TT (predicative, first-order logic, intuitionistic)
- ◆ TTs found in Coq and Lego (impredicative, higher-order logic)

## ◆ Technology based on type theories

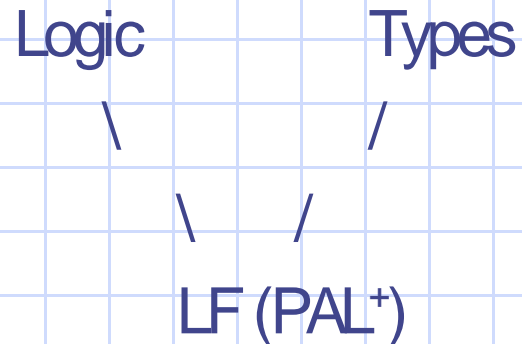
- ◆ Mainly applied to constructive math (so far)
- ◆ Applications to classical math?

## ◆ Logic-enriched type theories (Peter Aczel)

- ◆ Separation of logics and types
- ◆ Flexibility

# Logics and type theories in LFs

## ◆ Structure of logic-enriched TTs in logical framework



\* Compare this with existing TTs.

# PAL<sup>+</sup>

## ◆ Logical frameworks

Specifications of TTs and other formal systems

- Martin-Löf's LF
- PAL<sup>+</sup> – parameterisation and definitions

## ◆ Some LF notations of “kinds”

- Type – the universe of types
- $\text{El}(A)$  – the kind of elements of type  $A$

# Logics in PAL<sup>+</sup>

## ◆ Propositional universe

- ◆ Prop kind
  - ” Note: Prop is not a type (of kind Type or in a type universe).
- ◆ Prf[P] kind [P : Prop]
  - ” Note: Propositions are not types.

## ◆ Classical FOL: an example

- ◆  $P \wedge Q : \text{Prop} [P : \text{Prop}, Q : \text{Prop}]$
- ◆  $\forall [A, P] : \text{Prop} [A : \text{Type}, P[x:A] : \text{Prop}]$
- ◆  $\neg P : \text{Prop} [P : \text{Prop}]$ 
  - ”  $\text{DN}[P, p] : \text{Prf}(P) [P : \text{Prop}, p : \text{Prf}[\neg\neg P]]$

# Types

## ◆ Inductive types/families

- As found in type theory
  - ” Introduced by strictly positive operators (e.g. Nats, Trees, W-types, ...)
- With Induction Rule to eliminate over propositions.

## ◆ Example: the natural numbers

- $N : \text{Type}$ ,  $0 : N$ ,  $\text{succ}[n] : N \ [n : N]$
- Elimination over types: for  $C[n] : \text{Type} \ [n : N]$ 
  - ”  $\text{Elim}_T[C, c, f, n] : C[n]$  (plus computational rules for  $\text{Elim}_T$ )
- Induction over propositions: for  $P[n] : \text{Prop} \ [n : N]$ 
  - ”  $\text{Elim}_P[P, c, f, n] : P[n]$  (computational rules for  $\text{Elim}_P$  are optional)

## ◆ Universes

- As before, Peano's fourth axiom can be proved if there are universes (due to Peter Aczel).

# (Im)predicativity and other issues

## ◆ Predicativity

- The system presented so far is predicative.
- Predicative classical mathematics (c.f. Weyl/Feferman's work)

## ◆ Impredicativity

- E.g. impredicative (typed) sets

$\text{Set}[A] \text{ --- } "A \rightarrow \text{Prop}"$

This introduces impredicativity (and requires further study).

## ◆ Other issues: e.g., definite description.

# Pythagoras

- ◆ “Pythagoras: machine support for semi-formal proof-oriented mathematics”
  - ◆ Joint project with Peter Aczel at Manchester, funded by U.K. EPSRC
- ◆ Work so far (as related to this talk):
  - ◆ Implementation of PAL<sup>+</sup>, ind types and classical FOL (Yong Luo)
  - ◆ Formalisation of math (Robin Adams and Yong Luo)
    - ” Proofs and Fundamentals: a first course in mathematics.  
Ethan D. Bloch. Birkhauser, 2000. ISBN: 0817641114.
    - ” Predicative development of the number systems
    - ” Impredicative development of sets etc.
- ◆ Other work in progress
  - ◆ Semi-formal, user-friendly tool





◆ Flexibility  $\Leftarrow$  other applications in reasoning

- ◆ E.g., reasoning about dependently-typed programs

◆ Comparisons with existing systems

- ◆ Existing TTs
- ◆ HOL etc.

