Ordinal Strength of Logic-Enriched Type Theories

Robin Adams

Royal Holloway, University of London

27 March 2012

Introduction

- Type theories are formal languages in which both *algorithms* and *proofs* may be expressed.

Introduction

- Type theories are formal languages in which both *algorithms* and *proofs* may be expressed.
- There should therefore be applications to proof complexity.

Introduction

- Type theories are formal languages in which both *algorithms* and *proofs* may be expressed.
- There should therefore be applications to proof complexity.
- The type theory and 'mainstream' logical communities have been very separate.

Introduction

- Type theories are formal languages in which both *algorithms* and *proofs* may be expressed.
- There should therefore be applications to proof complexity.
- The type theory and 'mainstream' logical communities have been very separate.
- One reason is it is difficult to translate directly between a type theory and a mainstream logic (syntax or semantics).

Introduction

- Type theories are formal languages in which both *algorithms* and *proofs* may be expressed.
- There should therefore be applications to proof complexity.
- The type theory and 'mainstream' logical communities have been very separate.
- One reason is it is difficult to translate directly between a type theory and a mainstream logic (syntax or semantics).
- *Logic-enriched type theories* (LTTs) may be the bridge we need.

Introduction

- Type theories are formal languages in which both *algorithms* and *proofs* may be expressed.
- There should therefore be applications to proof complexity.
- The type theory and 'mainstream' logical communities have been very separate.
- One reason is it is difficult to translate directly between a type theory and a mainstream logic (syntax or semantics).
- *Logic-enriched type theories* (LTTs) may be the bridge we need.
- I will describe my work so far on LTTs, and some problems I believe LTTs may be able to help with.

1. Introduction to Type Theory and LTTs
2. Weyl's School of Predicativism
   - Proof Technique for Conservativity
   - Ordinal Strength of Weyl's Foundation
3. Potential Applications
   - Ordinal Strength of Theories
   - Application to Finite Model Theory
4. Conclusion

Type Theory

A *type theory* is a formal system that deals with *judgements* of the for

$$x_1 : A_1, \ldots, x_n : A_n \vdash a : A$$

$$x_1 : A_1, \ldots, x_n : A_n \vdash a = b : A$$

'a is an object of *type* A'
'a and b are equal objects of type A'
Examples:

$$\begin{aligned} \vdash 0 &: \mathbb{N} \\ x : \mathbb{N} \vdash \langle x, 7 \rangle &: \mathbb{N} \times \mathbb{N} \\ \vdash \lambda x : \mathbb{N}.x + 5 &: \mathbb{N} \to \mathbb{N} \\ f : \mathbb{N} \to \mathbb{N} \vdash f(5) &: \mathbb{N} \end{aligned}$$

$\lambda x : A.b$ is the function f with domain A such that $f(x) = b$ for all $x : A$.

Type Theory

Objects	Types

Type Theory

Objects	Types

Type Theory

	Objects	Types
Natural numbers	$0, s0, ss0, \ldots$	$\mathbb{N}$

Type Theory

	Objects	Types
Natural numbers	$0, s0, ss0, \ldots$	$\mathbb{N}$
Pairs	$\langle a, b \rangle$	$A \times B$

Type Theory

	Objects	Types
Natural numbers	$0,\ s0,\ ss0,\ \ldots$	$\mathbb{N}$
Pairs	$\langle a, b \rangle$	$A \times B$
Functions	$\lambda x.b[x]$	$A \to B$

Type Theory

	Objects	Types
Natural numbers	$0, s0, ss0, \ldots$	$\mathbb{N}$
Pairs	$\langle a, b \rangle$	$A \times B$
Functions	$\lambda x.b[x]$	$A \to B$
Variables	x	A

Type Theory

	Objects	Types
Natural numbers	$0,\ s0,\ ss0,\ \ldots$	$\mathbb{N}$
Pairs	$\langle a, b\rangle$	$A \times B$
Functions	$\lambda x.b[x]$	$A \to B$
Variables	x	A
Vectors	$\langle\rangle : Vec(A, 0)$ $l :: a : Vec(A, sn)$	$Vec(A, n)$

Type Theory

	Objects	Types
Natural numbers	$0, s0, ss0, \ldots$	$\mathbb{N}$
Pairs	$\langle a, b \rangle$	$A \times B$
Functions	$\lambda x.b[x]$	$A \to B$
Variables	x	A
Vectors	$\langle\rangle : Vec(A, 0)$ $l :: a : Vec(A, sn)$	$Vec(A, n)$
	$\langle a, b \rangle$	$\Sigma x : A.B[x]$

Type Theory

	Objects	Types
Natural numbers	$0,\ s0,\ ss0,\ \ldots$	$\mathbb{N}$
Pairs	$\langle a, b \rangle$	$A \times B$
Functions	$\lambda x.b[x]$	$A \to B$
Variables	x	A
Vectors	$\langle\rangle : Vec(A, 0)$ $l :: a : Vec(A, sn)$	$Vec(A, n)$
	$\langle a, b \rangle$	$\Sigma x : A.B[x]$
	$\lambda x.b[x]$	$\Pi x : A.B[x]$

Type Theory

	Objects	Types
Natural numbers	$0,\ s0,\ ss0,\ \ldots$	$\mathbb{N}$
Pairs	$\langle a, b \rangle$	$A \times B$
Functions	$\lambda x.b[x]$	$A \to B$
Variables	x	A
Vectors	$\langle\rangle : Vec(A, 0)$ $l :: a : Vec(A, sn)$	$Vec(A, n)$
	$\langle a, b \rangle$	$\Sigma x : A.B[x]$
	$\lambda x.b[x]$	$\Pi x : A.B[x]$
	one object if $a = b$ no objects if $a \neq b$	$I(A, a, b)$

Type Theory

	Objects	Types
Natural numbers	$0,\ s0,\ ss0,\ \ldots$	$\mathbb{N}$
Pairs	$\langle a, b\rangle$	$A \times B$
Functions	$\lambda x.b[x]$	$A \to B$
Variables	x	A
Vectors	$\langle\rangle : Vec(A, 0)$ $l :: a : Vec(A, sn)$	$Vec(A, n)$
	$\langle a, b\rangle$	$\Sigma x : A.B[x]$
	$\lambda x.b[x]$	$\Pi x : A.B[x]$
	one object if $a = b$ no objects if $a \neq b$	$I(A, a, b)$
Universe	$\mathbb{N},\ \mathbb{N} \times \mathbb{N},\ \mathbb{N} \to \mathbb{N},\ \ldots$	U

Rules in Type Theory

Functions

$$\frac{f : A \to B \qquad a : A}{f(a) : B} \qquad \frac{b : B}{\lambda x : A.b : B}$$

Pairs

$$\frac{a : A \qquad b : B}{(a, b) : A \times B}$$

$$\frac{a : A \times B}{a.1 : A} \qquad \frac{a : A \times B}{a.2 : B}$$

Wait A Minute . . .

Wait A Minute . . .

$$
\begin{array}{llll}
\text{Term} & M & ::= & x \mid \lambda x.M \mid MM \\
\text{Type} & A & ::= & \alpha \mid A \to A
\end{array}
$$

Wait A Minute . . .

$$
\begin{array}{llll}
\text{Term} & M & ::= & x \mid \lambda x.M \mid MM \\
\text{Type} & A & ::= & \alpha \mid A \to A
\end{array}
$$

$$
\frac{}{x_1 : A_1, \ldots, x_n : A_n \vdash x_i : A_i}
\qquad
\frac{\Gamma \vdash F : A \to B \quad \Gamma \vdash M : A}{\Gamma \vdash FM : B}
$$

$$
\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.M : A \to B}
$$

Wait A Minute . . .

$$\begin{array}{lccl} \text{Term} & M & ::= & x \mid \lambda x.M \mid MM \\ \text{Type} & A & ::= & \alpha \mid A \to A \end{array}$$

$$\frac{}{x_1 : A_1, \ldots, x_n : A_n \vdash x_i : A_i} \qquad \frac{\Gamma \vdash F : A \to B \quad \Gamma \vdash M : A}{\Gamma \vdash FM : B}$$

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.M : A \to B}$$

Curry (1958) noticed something strange . . .

Wait A Minute . . .

$$
\begin{array}{llll}
\text{Term} & M & ::= & x \mid \lambda x.M \mid MM \\
\text{Type} & A & ::= & \alpha \mid A \to A
\end{array}
$$

$$\frac{}{x_1 : A_1, \ldots, x_n : A_n \vdash x_i : A_i} \qquad \frac{\Gamma \vdash F : A \to B \quad \Gamma \vdash M : A}{\Gamma \vdash FM : B}$$

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.M : A \to B}$$

Curry (1958) noticed something strange . . .

Wait A Minute . . .

$$\begin{array}{llll} \text{Term} & M & ::= & x \mid \lambda x.M \mid MM \\ \text{Type} & A & ::= & \alpha \mid A \to A \end{array}$$

$$\frac{}{x_1 : A_1, \ldots, x_n : A_n \vdash x_i : A_i} \qquad \frac{\Gamma \vdash F : A \to B \quad \Gamma \vdash M : A}{\Gamma \vdash FM : B}$$

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.M : A \to B}$$

Curry (1958) noticed something strange . . .

Wait A Minute ...

Term	M	$::=$	$x \mid \lambda x.M \mid MM$
Type	A	$::=$	$\alpha \mid A \to A$

$$\frac{}{A_1, \ldots, A_n \vdash A_i} \qquad \frac{\Gamma \vdash A \to B \quad \Gamma \vdash A}{\Gamma \vdash B}$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}$$

Curry (1958) noticed something strange ...

The Plot Thickens . . .

Howard (1969):

$$\begin{array}{llll} \text{Term} & M & ::= & x \mid \lambda x.M \mid MM \\ \text{Type} & A & ::= & \alpha \mid A \to A \end{array}$$

$$\frac{}{x_1 : A_1, \ldots, x_n : A_n \vdash x_i : A_i} \qquad \frac{\Gamma \vdash F : A \to B \quad \Gamma \vdash M : A}{\Gamma \vdash FM : B}$$

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.M : A \to B}$$

The Plot Thickens . . .

Howard (1969):

$$\begin{array}{lcll} \text{Term} & M & ::= & x \mid \lambda x.M \mid MM \mid (M, M) \mid \pi_1(M) \mid \pi_2(M) \\ \text{Type} & A & ::= & \alpha \mid A \to A \mid A \times A \end{array}$$

$$\frac{}{x_1 : A_1, \ldots, x_n : A_n \vdash x_i : A_i} \qquad \frac{\Gamma \vdash F : A \to B \quad \Gamma \vdash M : A}{\Gamma \vdash FM : B}$$

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.M : A \to B}$$

The Plot Thickens ...

Howard (1969):

$$\begin{array}{lcll} \text{Term} & M & ::= & x \mid \lambda x.M \mid MM \mid (M, M) \mid \pi_1(M) \mid \pi_2(M) \\ \text{Type} & A & ::= & \alpha \mid A \to A \mid A \times A \end{array}$$

$$\frac{}{x_1 : A_1, \ldots, x_n : A_n \vdash x_i : A_i} \qquad \frac{\Gamma \vdash F : A \to B \quad \Gamma \vdash M : A}{\Gamma \vdash FM : B}$$

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.M : A \to B}$$

$$\frac{\Gamma \vdash M : A \quad \Gamma \vdash N : B}{\Gamma \vdash (M, N) : A \times B}$$

$$\frac{\Gamma \vdash M : A \times B}{\Gamma \vdash \pi_1(M) : A} \qquad \frac{\Gamma \vdash M : A \times B}{\Gamma \vdash \pi_2(M) : B}$$

The Plot Thickens . . .

Howard (1969):

$$\text{Type} \quad A \quad ::= \quad \alpha \mid A \to A \mid A \times A$$

$$\frac{}{x_1 : A_1, \ldots, x_n : A_n \vdash x_i : A_i} \qquad \frac{\Gamma \vdash F : A \to B \quad \Gamma \vdash M : A}{\Gamma \vdash FM : B}$$

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.M : A \to B}$$

$$\frac{\Gamma \vdash M : A \quad \Gamma \vdash N : B}{\Gamma \vdash (M, N) : A \times B}$$

$$\frac{\Gamma \vdash M : A \times B}{\Gamma \vdash \pi_1(M) : A} \qquad \frac{\Gamma \vdash M : A \times B}{\Gamma \vdash \pi_2(M) : B}$$

The Plot Thickens . . .

Howard (1969):

$$\text{Type} \quad A \quad ::= \quad \alpha \mid A \to A \mid A \times A$$

$$\frac{}{A_1, \ldots, A_n \vdash A_i} \qquad \frac{\Gamma \vdash A \to B \quad \Gamma \vdash A}{\Gamma \vdash B}$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}$$

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \times B}$$

$$\frac{\Gamma \vdash A \times B}{\Gamma \vdash A} \qquad \frac{\Gamma \vdash A \times B}{\Gamma \vdash B}$$

The Plot Thickens ...

Howard (1969):

$$\text{Type} \quad A \quad ::= \quad \alpha \mid A \to A \mid A \wedge A$$

$$\frac{}{A_1, \ldots, A_n \vdash A_i} \qquad \frac{\Gamma \vdash A \to B \quad \Gamma \vdash A}{\Gamma \vdash B}$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}$$

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B}$$

$$\frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A} \qquad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B}$$

The Plot Thickens . . .

Howard (1969):

$$\text{Type} \quad A \quad ::= \quad \alpha \mid A \to A \mid A \wedge A$$

$$\frac{}{A_1, \ldots, A_n \vdash A_i} \qquad \frac{\Gamma \vdash A \to B \quad \Gamma \vdash A}{\Gamma \vdash B}$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}$$

$$\frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B}$$

$$\frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A} \qquad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B}$$

Propositions-as-Types

Type theories can be given a second interpretation — read '$a : A$' as 'a is a proof of the proposition A'

When read this way:

- $A \times B$ is the proposition 'A and B'
- $A \to B$ is the proposition 'A implies B'
- $\Pi x : A.B$ is the proposition 'for all $x : A$, B'
- $\Sigma x : A.B$ is the proposition 'there exists $x : A$ such that B'
- $I(A, a, b)$ is the proposition 'a equals b'

The fact that the formal rules for typing and logic are identical is known as the *Curry-Howard isomorphism*.

Relating Type Theories and Mainstream Logics

In logic, we write down axioms, then discover which objects these allow us to define (i.e. which objects we can prove exist / which objects exist in every model).

Relating Type Theories and Mainstream Logics

In logic, we write down axioms, then discover which objects these allow us to define (i.e. which objects we can prove exist / which objects exist in every model).

In type theory, we write down types (mostly inductively defined collections) based on which objects we wish to have, then discover what logical principles these allow us.

Relating Type Theories and Mainstream Logics

In logic, we write down axioms, then discover which objects these allow us to define (i.e. which objects we can prove exist / which objects exist in every model).

In type theory, we write down types (mostly inductively defined collections) based on which objects we wish to have, then discover what logical principles these allow us.

The two worlds are difficult to relate directly.

Aczel and Gambino [GA06] investigated the relationship between Constructive ZF set theory (CZF) and the type theory ML_1V.

Aczel and Gambino [GA06] investigated the relationship between Constructive ZF set theory (CZF) and the type theory ML_1V. They discovered it helps to introduce an intermediate system, a *logic-enriched type theory* (LTT).

Logic-Enriched Type Theories

A logic-enriched type theory (LTT) consists of:

- a type theory
- a separate set of formulas
- a set of rules that determine which formulas are provable.

Logic-Enriched Type Theories

A logic-enriched type theory (LTT) consists of:

- a type theory
- a separate set of formulas
- a set of rules that determine which formulas are provable.
  It thus has two 'worlds' — the logical world and the type theory world.

Logic-Enriched Type Theories

A logic-enriched type theory (LTT) consists of:

- a type theory
- a separate set of formulas
- a set of rules that determine which formulas are provable.
  It thus has two 'worlds' — the logical world and the type theory world.
  These two worlds interact but can be modified separately.

Predicativism

A definition is impredicative if it involves a certain kind of circularity:

- quantifying over all sets when defining a set
- quantifying over all real numbers when defining a real number.

Predicativism

A definition is impredicative if it involves a certain kind of circularity:

- quantifying over all sets when defining a set
- quantifying over all real numbers when defining a real number.

Example: The definition of the least upper bound of a set of reals is impredicative, as it involves quantifying over all real numbers.

Predicativism

A definition is impredicative if it involves a certain kind of circularity:

- quantifying over all sets when defining a set
- quantifying over all real numbers when defining a real number.

Example: The definition of the least upper bound of a set of reals is impredicative, as it involves quantifying over all real numbers.

Predicativism is the view that impredicative definitions are illegitimate.

How can we restrict our methods of definition so that impredicative definitions are ruled out?

Predicativism

A definition is impredicative if it involves a certain kind of circularity:

- quantifying over all sets when defining a set
- quantifying over all real numbers when defining a real number.

Example: The definition of the least upper bound of a set of reals is impredicative, as it involves quantifying over all real numbers.

Predicativism is the view that impredicative definitions are illegitimate.

How can we restrict our methods of definition so that impredicative definitions are ruled out? Weyl's solution (1914):

- Divide mathematical objects into categories.

Predicativism

A definition is impredicative if it involves a certain kind of circularity:

- quantifying over all sets when defining a set
- quantifying over all real numbers when defining a real number.

Example: The definition of the least upper bound of a set of reals is impredicative, as it involves quantifying over all real numbers.

Predicativism is the view that impredicative definitions are illegitimate.

How can we restrict our methods of definition so that impredicative definitions are ruled out? Weyl's solution (1914):

- Divide mathematical objects into categories.
- Divide categories into basic and ideal categories.

Predicativism

A definition is impredicative if it involves a certain kind of circularity:

- quantifying over all sets when defining a set
- quantifying over all real numbers when defining a real number.

Example: The definition of the least upper bound of a set of reals is impredicative, as it involves quantifying over all real numbers.

Predicativism is the view that impredicative definitions are illegitimate.

How can we restrict our methods of definition so that impredicative definitions are ruled out? Weyl's solution (1914):

- Divide mathematical objects into categories.
- Divide categories into basic and ideal categories.
  - Natural numbers form a basic category.

Predicativism

A definition is impredicative if it involves a certain kind of circularity:

- quantifying over all sets when defining a set
- quantifying over all real numbers when defining a real number.

Example: The definition of the least upper bound of a set of reals is impredicative, as it involves quantifying over all real numbers.

Predicativism is the view that impredicative definitions are illegitimate.

How can we restrict our methods of definition so that impredicative definitions are ruled out? Weyl's solution (1914):

- Divide mathematical objects into categories.
- Divide categories into basic and ideal categories.
  - Natural numbers form a basic category.
  - For any category A, the sets of As form an ideal category.

Predicativism

A definition is impredicative if it involves a certain kind of circularity:

- quantifying over all sets when defining a set
- quantifying over all real numbers when defining a real number.

Example: The definition of the least upper bound of a set of reals is impredicative, as it involves quantifying over all real numbers.

Predicativism is the view that impredicative definitions are illegitimate.

How can we restrict our methods of definition so that impredicative definitions are ruled out? Weyl's solution (1914):

- Divide mathematical objects into categories.
- Divide categories into basic and ideal categories.
  - Natural numbers form a basic category.
  - For any category A, the sets of As form an ideal category.
- When defining a set, we may only quantify over the basic categories.

ACA_0 is "a modern formulation of Weyl's system" (Feferman).

- Two sorts: natural numbers and sets of natural numbers (second-order language).

ACA_0 is "a modern formulation of Weyl's system" (Feferman).

- Two sorts: natural numbers and sets of natural numbers (second-order language).
- Axioms:
  Peano's axioms

ACA_0 is "a modern formulation of Weyl's system" (Feferman).

- Two sorts: natural numbers and sets of natural numbers (second-order language).
- Axioms:
  Peano's axioms
  Restricted induction $\phi[0] \to \forall x(\phi[x] \to \phi[x']) \to \forall x \phi[x]$ for $\phi[x]$ arithmetic

ACA_0 is "a modern formulation of Weyl's system" (Feferman).

- Two sorts: natural numbers and sets of natural numbers (second-order language).
- Axioms:

Peano's axioms	
Restricted induction	$\phi[0] \to \forall x(\phi[x] \to \phi[x']) \to \forall x \phi[x]$ for $\phi[x]$ arithmetic
Arithmetic Comprehension Axiom	$\{x : \phi[x]\}$ exists for $\phi[x]$ arithmetic

ACA_0

Weyl's Foundation as a System of Predicate Logic

ACA_0 is "a modern formulation of Weyl's system" (Feferman).

- Two sorts: natural numbers and sets of natural numbers (second-order language).
- Axioms:

Peano's axioms	
Restricted induction	$\phi[0] \to \forall x(\phi[x] \to \phi[x']) \to \forall x \phi[x]$ for $\phi[x]$ arithmetic
Arithmetic Comprehension Axiom	$\{x : \phi[x]\}$ exists for $\phi[x]$ arithmetic

Conservative extension of PA.

ACA_0 is "a modern formulation of Weyl's system" (Feferman).

- Two sorts: natural numbers and sets of natural numbers (second-order language).
- Axioms:

Peano's axioms	
Restricted induction	$\phi[0] \to \forall x(\phi[x] \to \phi[x']) \to \forall x \phi[x]$ for $\phi[x]$ arithmetic
Arithmetic Comprehension Axiom	$\{x : \phi[x]\}$ exists for $\phi[x]$ arithmetic

Conservative extension of PA.

Shortcomings: Weyl uses

- sets of sets
- definition of sets by recursion
- full induction

Define an LTT named $\mathrm{LTT_W}$ [AL10b]:

Type Theory World

Universe U (objects are the basic categories)

$$\mathbb{N}$$
$$\times$$
$$\rightarrow$$
$$Set(A)$$

Logical World

Universe *prop* (objects are the arithmetic formulas)

$$=$$
$$\wedge, \vee, \neg, \rightarrow$$
$$\forall, \exists$$

Define an LTT named $\mathrm{LTT_W}$ [AL10b]:

Type Theory World

Universe *U* (objects are the basic categories)

$$\mathbb{N}$$
$$\times$$
$$\rightarrow$$
$$Set(A)$$

Logical World

Universe *prop* (objects are the arithmetic formulas)

$$=$$
$$\wedge, \vee, \neg, \rightarrow$$
$$\forall, \exists$$

All Weyl's results can be formalised in $\mathrm{LTT_W}$ — checked by proof assistant Plastic.

ACA_0 is Embeddable in LTT_W

$$\text{Formulas of } \mathrm{ACA}_0 \quad \longrightarrow \quad \text{Formulas of } \mathrm{LTT}_\mathrm{W}$$

Define a mapping

ACA_0 is Embeddable in $\mathrm{LTT_W}$

Define a mapping

Formulas of ACA_0	$\longrightarrow$	Formulas of $\mathrm{LTT_W}$
$=$		$=$
$\wedge$		$\wedge$
$\vee$		$\vee$
$\vdots$		$\vdots$
$\forall x$		$\forall x : \mathbb{N}$
$\exists x$		$\exists x : \mathbb{N}$
$\forall X$		$\forall X : Set(\mathbb{N})$
$\exists X$		$\exists X : Set(\mathbb{N})$

ACA_0 is Embeddable in $\mathrm{LTT_W}$

Define a mapping

$$\begin{array}{ccc} \text{Formulas of } \mathrm{ACA}_0 & \longrightarrow & \text{Formulas of } \mathrm{LTT_W} \\ = & & = \\ \wedge & & \wedge \\ \vee & & \vee \\ \vdots & & \vdots \\ \forall x & & \forall x : \mathbb{N} \\ \exists x & & \exists x : \mathbb{N} \\ \forall X & & \forall X : Set(\mathbb{N}) \\ \exists X & & \exists X : Set(\mathbb{N}) \end{array}$$

But $\mathrm{LTT_W}$ goes further:

- has types $Set(Set(\mathbb{N}))$, ...
- allows definition by recursion in $Set(A)$
- allows full induction

Subsystems of $\mathrm{LTT_W}$

In $\mathrm{LTT_W}$, we can eliminate over any type. Let us define the following subsystems of $\mathrm{LTT_W}$:

Subsystems of $\mathrm{LTT_W}$

In $\mathrm{LTT_W}$, we can eliminate over any type. Let us define the following subsystems of $\mathrm{LTT_W}$:

- $\mathrm{LTT_0}$ — restrict $E_{\mathbb{N}}$ to the members of U and induction to *prop*.
- $\mathrm{LTT_W}^1$ — restrict $E_{\mathbb{N}}$ to the members of U.
- $\mathrm{LTT_W}^2$ — restrict $E_{\mathbb{N}}$ to A and $\mathcal{P}A$ $(A : U)$
- $\mathrm{LTT_W}^3$ — restrict $E_{\mathbb{N}}$ to A, $\mathcal{P}A$, $\mathcal{PP}A$ $(A : U)$
- $\cdots$

Subsystems of $\mathrm{LTT_W}$

In $\mathrm{LTT_W}$, we can eliminate over any type. Let us define the following subsystems of $\mathrm{LTT_W}$:

- $\mathrm{LTT_0}$ — restrict $E_{\mathbb{N}}$ to the members of U and induction to *prop*.
- ${\mathrm{LTT_W}}^1$ — restrict $E_{\mathbb{N}}$ to the members of U.
- ${\mathrm{LTT_W}}^2$ — restrict $E_{\mathbb{N}}$ to A and $\mathcal{P}A$ $(A : U)$
- ${\mathrm{LTT_W}}^3$ — restrict $E_{\mathbb{N}}$ to A, $\mathcal{P}A$, $\mathcal{PP}A$ $(A : U)$
- $\cdots$

Theorem ([AL10a])

$\mathrm{ACA_0}$ *can be conservatively embedded in* $\mathrm{LTT_0}$.
ACA can be conservatively embedded in ${\mathrm{LTT_W}}^1$.
ACA_0^+ *can be embedded in* ${\mathrm{LTT_W}}^2$ *(**Conjecture**: This is not conservative.)*

Subsystems of $\mathrm{LTT_W}$

In $\mathrm{LTT_W}$, we can eliminate over any type. Let us define the following subsystems of $\mathrm{LTT_W}$:

- $\mathrm{LTT_0}$ — restrict $E_{\mathbb{N}}$ to the members of U and induction to *prop*.
- ${\mathrm{LTT_W}}^1$ — restrict $E_{\mathbb{N}}$ to the members of U.
- ${\mathrm{LTT_W}}^2$ — restrict $E_{\mathbb{N}}$ to A and $\mathcal{P}A$ $(A : U)$
- ${\mathrm{LTT_W}}^3$ — restrict $E_{\mathbb{N}}$ to A, $\mathcal{P}A$, $\mathcal{P}\mathcal{P}A$ $(A : U)$
- $\cdots$

Theorem ([AL10a])

$\mathrm{ACA_0}$ *can be conservatively embedded in* $\mathrm{LTT_0}$.
ACA can be conservatively embedded in ${\mathrm{LTT_W}}^1$.
ACA_0^+ *can be embedded in* ${\mathrm{LTT_W}}^2$ *(**Conjecture**: This is not conservative.)*

Conjecture

${\mathrm{LTT_W}}^{n+1}$ is never conservative over ${\mathrm{LTT_W}}^n$.

Conservativity of ACA_0 over PA

We interpret the formulas of ACA_0 as being statements about the syntax and theorems of PA.
Define a *valuation* to be a mapping of:

- first-order variables to terms of PA;
- second-order variables to formulas of PA.

Define what it means for a valuation v to *satisfy* a formula ϕ thus:

- If ϕ is arithmetic, then $v \models \phi$ iff $v(\phi)$ is a theorem of PA.
- ...

A formula is *true* iff it is satisfied by every valuation.
Prove:

- Every theorem of ACA_0 is true.
- Every first-order formula that is true is a theorem of PA.

Ordinal Strength of a Type Theory

Definition

A type theory is *consistent* iff there is no term M such that $\vdash M : \emptyset$ in derivable

We can code the terms, types, ... of a type theory with Gödel numbers, and represent the relation '$\mathcal{J}$ is derivable' in PRA.

Definition

The *ordinal strength* of a type theory T is the least ordinal α such that

$$PRA + TI(\alpha) \vdash \neg \exists M (\vdash_T M : \emptyset)$$

Proof Techniques for Ordinal Strength

To show that a predicate logic theory T has ordinal strength α:

- Prove that $|T| \geq \alpha$ by showing that $T \vdash TI(\beta)$ for all $\beta < \alpha$ (Easy).
- Prove that $|T| \leq \alpha$ by proving $Con(T)$ in a theory of strength α (Hard).

To show that a type theory T has ordinal strength α:

- Prove that $|T| \geq \alpha$ by constructing in T a well-ordering of length α (Hard).
- Prove that $|T| \leq \alpha$ by constructing a model of T in a variant of KP set theory of strength α (Easy).

Result — LTT_W has ordinal $\phi_{\epsilon_0}(0)$

Setzer and I recently proved that LTT_W has strength $\phi_{\epsilon_0}(0)$, the same strength as ACA, $\hat{ID}_1$ and ML_1.

Sketch Proof Extend $\mathrm{LTT_W}$ with a type AE that represents the ordinals below Γ_0:

$$\frac{}{0 : AE} \qquad \frac{\alpha : AE \quad \beta : AE}{\alpha + \omega^\beta : AE} \qquad \frac{\alpha : AE \quad \beta : AE}{\phi_\alpha(\beta) : AE}$$

This is a conservative extension (proof: members of AE can be coded as natural numbers).

Definition (Progressive)

$X : \mathcal{P}AE$ is *progressive* iff

$$\forall \alpha (\forall \beta < \alpha . \beta \in X) \to \alpha \in X$$

$\alpha : AE$ is *accessible* iff it is in every progressive set.

Result — LTT_W has ordinal $\phi_{\epsilon_0}(0)$

Define a *lens* for a function $\phi : AE \to AE$ to be a function $\Phi : \mathcal{P}AE \to \mathcal{P}AE$ such that:

- if X is progressive then $\Phi(X)$ is progressive;
- if X is progressive then $\Phi(X) \subseteq \phi^{-1}(X)$.

Prove that:

1. If there is a lens for ϕ then the accessible ordinals are closed under ϕ.
2. If there is a lens for ϕ then there is a lens for ϕ^n and ϕ^ω.

It follows that, for all $\alpha < \epsilon_0$, $\mathrm{LTT_W}$ proves that there is a lens for ϕ_α.

Result — LTT_W has ordinal $\phi_{\epsilon_0}(0)$

Define a *lens* for a function $\phi : AE \to AE$ to be a function $\Phi : \mathcal{P}AE \to \mathcal{P}AE$ such that:

- if X is progressive then $\Phi(X)$ is progressive;
- if X is progressive then $\Phi(X) \subseteq \phi^{-1}(X)$.

Prove that:

1. If there is a lens for ϕ then the accessible ordinals are closed under ϕ.
2. If there is a lens for ϕ then there is a lens for ϕ^n and ϕ^ω.

It follows that, for all $\alpha < \epsilon_0$, $\mathrm{LTT_W}$ proves that there is a lens for ϕ_α.

As soon as we had the result, we saw that there is an easier way:

$$\hat{ID}_1 \to LTT_W \to ML_1$$

Idea

Can we relate a predicate logic and a type theory, via an LTT, so that we know they have the same ordinal strength?

$$L \longleftrightarrow LTT \longleftrightarrow T$$

We could then prove a lower bound for L and an upper bound for T.

Application to Finite Model Theory

Work in Progress

For a set K of ordered structures:

- membership of K is in P iff K is axiomatised by a sentence of $FO(IFP)$;
- membership of K is in NP iff K is axiomatised by a Σ_1-sentence.

We can make *prop* the set of formulas of $FO(IFP)$ by adding

$$IFP : (A : U)(\mathcal{P}A \to \mathcal{P}A) \to \mathcal{P}A \to \mathcal{P}A$$

We can also make *prop* the set of Σ_1 sentences, by closing it under

$$=, \neq, \wedge, \vee, \exists$$

There is ongoing work in type theory on fixed points for type constructors: Could LTTs provide a bridge between the model theory of $FO(IFP)$ and type theories with fixed points? Perhaps providing a proof theory for $FO(IFP)$?

Conclusion

- LTTs might possibly be a useful tool for bridging type theories and mainstream logics.

Conclusion

- LTTs might possibly be a useful tool for bridging type theories and mainstream logics.
- At the moment, I only have new proofs of old results, and a vague intuition that some technical machinery is simpler.

Conclusion

- LTTs might possibly be a useful tool for bridging type theories and mainstream logics.
- At the moment, I only have new proofs of old results, and a vague intuition that some technical machinery is simpler.
    - I think these are enough to show that LTTs have potential.

Conclusion

- LTTs might possibly be a useful tool for bridging type theories and mainstream logics.
- At the moment, I only have new proofs of old results, and a vague intuition that some technical machinery is simpler.
  - I think these are enough to show that LTTs have potential.
  - They are a solution looking for a problem.

Conclusion

- LTTs might possibly be a useful tool for bridging type theories and mainstream logics.
- At the moment, I only have new proofs of old results, and a vague intuition that some technical machinery is simpler.
  - I think these are enough to show that LTTs have potential.
  - They are a solution looking for a problem.
  - I really need to get better at inventing names.

Conclusion

- LTTs might possibly be a useful tool for bridging type theories and mainstream logics.
- At the moment, I only have new proofs of old results, and a vague intuition that some technical machinery is simpler.
    - I think these are enough to show that LTTs have potential.
    - They are a solution looking for a problem.
    - I really need to get better at inventing names.
- I hope someone in the audience is thinking LTTs might be the tool they've been looking for.

Conclusion

- LTTs might possibly be a useful tool for bridging type theories and mainstream logics.
- At the moment, I only have new proofs of old results, and a vague intuition that some technical machinery is simpler.
  - I think these are enough to show that LTTs have potential.
  - They are a solution looking for a problem.
  - I really need to get better at inventing names.
- I hope someone in the audience is thinking LTTs might be the tool they've been looking for.
- Please come talk to me over lunch.

References

Robin Adams and Zhaohui Luo.
Classical predicative logic-enriched type theories.
Annals of Pure and Applied Logic, 161:1315–1345, 2010.

Robin Adams and Zhaohui Luo.
Weyl's predicative classical mathematics as a logic-enriched type theory.
ACM Transactions on Computational Logic., 11(2), 2010.

Nicola Gambino and Peter Aczel.
The generalised type-theoretic interpretation of constructive set theory.
J. Symbolic Logic, 71(1):67–103, 2006.