Proof and Definition in Logic and Type Theory

Robin Adams
robin@cs.rhul.ac.uk

Royal Holloway, University of London

14 November 2008

Introduction

A system of logic consists of:

- a formal language in which the statements of mathematics can be written;
- a set of rules by which some of these statements can be proved.

I will talk about how three classes of system of logic:

- predicate logic
- type theory
- logic-enriched type theory

handle two questions

Definition How are new objects (sets, functions, . . .) introduced into the language?

Proof Which steps in a proof are acceptable?

Logic turns philosophical disagreements into formal questions.

Definition and Proof

Definition and proof are interleaved in a body of mathematics.

- If we have proved $\exists ! x \phi[x]$, we are allowed to define a to be the unique object such that $\phi[a]$.
- If we have defined an object a such that $\phi[a]$, we can prove $\exists x \phi[x]$.

A foundation of mathematics fixes:

- which methods of definition are acceptable;
- which methods of proof are acceptable.

Changing the axioms will change:

- the set of theorems
- the set of definable objects
- but should not change the acceptable methods of definition.

The methods of definition and methods of proof should be separate — but they are not in predicate logic or type theory.

Methods of Definition and Proof

Methods of Definition

Introduction of Sets For which predicates $\phi[x]$ can we introduce the set $\{x : \phi[x]\}$?

Definiton by Recursion Can we always assume a function defined by primitive recursion

$$f(x, 0) = g(x) \qquad f(x, y+1) = h(x, y, f(x, y))$$

is total?

Methods of Proof

Excluded Middle Can we assume $\phi \vee \neg\phi$ for every statement ϕ?

Induction For which predicates $\phi[x]$ can we prove $\forall x \phi[x]$ by induction?

Predicate Logic

Includes first-order, second-order, ... logic.

	Example
	PA
A language is specified by:	
a set of primitive function symbols	$0,\ ',\ +,\ \times$
a set of primitive relation symbols	$=,\ \leq$
The formulas are built up by $\wedge$, $\vee$, $\neg$, $\rightarrow$, $\forall$, $\exists$	
A theory is given by:	
a set of axioms	$x' = y' \rightarrow x = y$ $x' \neq 0$ $\vdots$
a set of rules of deduction	$\dfrac{\phi \rightarrow \psi \qquad \phi}{\psi}$

Predicate logic emphasises proof — no primitive mechanism for definition.

Definition in Predicate Logic

Definition is performed by extending the language and the theory.

If T is a theory over $\mathcal{L}$ and $T \vdash \forall x \exists ! y \phi[x, y]$, form

language	$\mathcal{L}'$	$=$	$\mathcal{L} + \{f\}$
theory	T'	$=$	$T + \{\forall x \phi[x, f(x)]\}$

T' is a conservative extension of T.

Example

Let $GCD(a, b, c)$ be the formula

$$c \mid a \wedge c \mid b \wedge \forall x(x \mid a \rightarrow x \mid b \rightarrow x \mid c)$$

Theorem of PA: $\forall x \forall y \exists ! z GCD(x, y, z)$.
We may safely add a function symbol *gcd* and the axiom $GCD(a, b, gcd(a, b))$.

Other ways of extending systems are possible.

Functions Definable in Peano Arithmetic

A function $f : \mathbb{N} \to \mathbb{N}$ is:

term definable by $t[x]$ — $t[\overline{n}]$ denotes $f(n)$

positive integer polynomials

expressible by $\phi[x, y]$ — $\phi[\overline{n}, \overline{f(n)}]$ is true

$\phi[\overline{n}, \overline{m}]$ is false for other m

arithmetic functions

representible by $\phi[x, y]$ — $\phi[\overline{n}, \overline{f(n)}]$ is provable

$\neg\phi[\overline{n}, \overline{m}]$ is provable for other m

recursive functions

definable by $\phi[x, y]$ — $\phi[\overline{n}, \overline{f(n)}]$ is provable

$\neg\phi[\overline{n}, \overline{m}]$ is provable for other m

$\forall x \exists! y \phi[x, y]$ is provable

recursive functions

Methods of Proof affect Methods of Definition

RCA_0 is a system of *second-order arithmetic*. Its language deals with:

- natural numbers
- sets of natural numbers.

The sets definable in RCA_0 are exactly the recursively enumerable sets.
If we add Σ_k-induction, we obtain bounded Σ_k-abstraction:

$$\{x \leq n \mid \phi[x]\}$$

can be proved to exist for $\phi[x]$ a Σ_k-formula.

Type Theory

Language deals with judgements

$M : A$	$M = N : A$
M is an object of type A	M and N are equal objects of type A.

	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

Proof in Type Theory

Handled by propositions-as-types. Identify

all types / members of a universe *Prop*	with propositions
objects	with proofs

Type	Proposition
$I(A, a, b)$	$a = b$
$A \times B$	$A \wedge B$
$A \rightarrow B$	$A \rightarrow B$
$\Sigma x : A.B[x]$	$\exists x : A.B[x]$
$\Pi x : A.B[x]$	$\forall x : A.B[x]$

Possible due to Curry-Howard isomorphism.

Heyting Semantics

- A proof of $A \wedge B$ consists of a proof of A and a proof of B.
- A proof of $A \vee B$ is either a proof of A or a proof of B.
- A proof of $A \rightarrow B$ is a function that takes a proof of A and returns a proof of B.

Definition Affects Proof

If we add:	we get:
definition by recursion	proof by induction
functions on types	second-order logic
‘freeze’ and ‘unfreeze’	classical logic

We cannot change the logic without changing the objects, or *vice versa*.

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.

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

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.

Shortcomings: Weyl uses

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

$\mathrm{LTT_W}$

Weyl's Foundation as a Logic-Enriched Type Theory

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.

Functions Definable in $\mathrm{LTT_W}$

A function $f : \mathbb{N} \to \mathbb{N}$ is:

term definable by $t : \mathbb{N} \to \mathbb{N}$ $\quad t[\overline{n}] = \overline{f(n)} : \mathbb{N}$

$\supseteq \epsilon_0$-**recursive functions**

set definable by $S : \mathrm{Set}(\mathbb{N} \times \mathbb{N})$ $\quad \langle \overline{n}, \overline{f(n)} \rangle \in S$ is provable

$\neg \langle \overline{n}, \overline{m} \rangle \in S$ is provable for other m

$\forall x \exists! y \langle x, y \rangle \in S$ is provable

recursive functions

ACA_0 is Embeddable in LTT_W

Define a mapping

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

But LTT_W goes further:

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

Conclusion

Logic-enriched type theories are systems of logic in which the mechanisms for definition and proof are separate.
This allows them to capture some foundations of mathematics better than predicate logic or type theory.
Aim: Produce a hierarchy of LTTs (similar to hierarchy in Reverse Mathematics) that capture a variety of foundations.
Next step: Nelson's predicativism/ultra-finitism