CAS 734 Winter 2005

06 Practice-Oriented Logics

Instructor: W. M. Farmer

Revised: 10 February 2005

Theory-Oriented vs. Practice-Oriented Logics

- Most traditional logics are **theory oriented**: they are designed to be studied and used in theory only.
  - Examples: First-order logic, Zermelo-Fraenkel set theory.

- A **practice-oriented logic** is intended for actual use in practice by engineers, scientists, mathematicians, and students.
  - Often are modifications of traditional logics.
  - Examples: Versions of Church's type theory used in the HOL, IMPS, PVS, and TPS systems.

- The logic of an ITPS needs to be practice oriented.

Issue 1: Undefinedness

- A mathematical term is **undefined** if it has no prescribed meaning or if it denotes a value that does not exist.
  - Undefined terms are commonplace in mathematics.

- Sources of undefinedness:
  1. **Improper function applications**: $\sqrt{-4}$.
  2. **Improper definite descriptions**:
     "the x such that $x^2 = 4$".
  3. **Improper indefinite descriptions**:
     "some x such $x^2 = -4$".

- A practice-oriented logic needs a way of handling undefinedness:
  - Traditional approach to undefinedness
  - Unspecified values
  - Error values
  - Ill-formed terms

Issue 2: Compound Values

- A **compound value** is a value composed of other values.
  - Examples: Sets, relations, and functions.

- A practice-oriented logic needs to provide the following services for compound values:
  - Quantification.
  - Abstraction mechanism (e.g. lambda-abstraction).
  - Organizational system for higher-order values.
  - Way of handling improper function application.

Issue 3: Types and Sorts

- There is no conventional distinction between a **type** and a **sort**.

- Types and sorts are used to:
  - Restrict the scope of variables.
  - Control the formation of expressions.
  - Organize higher-order values.
  - Classify expressions by their values.

- In mathematical practice, types and sorts are informal and used mainly for restricting the scope of variables.

- There is a wide range of type and sort systems.

- A practice-oriented logic needs types or sorts.

Issue 4: Polymorphism

- An operator is **polymorphic** if it can be applied to expressions of different types.

- Polymorphic operators are not usually needed in mathematical practice since, by convention, operators can be applied to all expressions (but the applications may be undefined).

- A practice-oriented logic needs polymorphic operators in some form:
  - Type variables.
  - Macro-abbreviations.
  - All values are members of a universal class.

Issue 5: Definite and Indefinite Description

- A **definite description** is an expression of the form "the x such that A" written formally as $\iota\, x \,.\, A$.

- An **indefinite description** is an expression of the form "some x such that A" written formally as $\epsilon\, x \,.\, A$.

- Definite descriptions, and to a less extent indefinite descriptions, are quite common in mathematical practice, but they often occur in a disguised form.

- Improper definite and indefinite descriptions are undefined.

- A practice-oriented logic needs either definite description or indefinite description.

Issue 6: Syntax as Values

- An expression has a two meanings:

  1. The value it denotes.
  2. Its syntactic construction.

- Both meanings are important in mathematics, but the distinction between them is often confused.

- A practice-oriented logic needs to be able to reason about both meanings of an expression.

Candidates for a Practice-Oriented Logic

- Higher-order logics.
  - Simple type theory.
  - Extensions of simple type theory.
  - Other type theories.

- Set theories.
  - Zermelo-Fraenkel (ZF) set theory.
  - Von-Neumann-Bernays-Gödel (NBG) set theory.

- Note: First-order logic is not good candidate for a practice-oriented logic.

Type Theory

- Russell introduced a logic now known as the **ramified theory of types** in 1908 to serve as a foundation for mathematics.
  - Included a hierarchy of types to avoid set-theoretic paradoxes such Russell's Paradox and semantic paradoxes such as Richard's paradox.
  - Employed as the logic of Whitehead and Russell's *Principia Mathematica*.
  - Not used today due to its high complexity.

- Chwistek and Ramsey suggested in the 1920s a simplified version of the ramified theory of types called the **simple theory of types** or, more briefly, **simple theory theory**.

- Church published in 1940 a formulation of simple theory theory with lambda-notation and lambda-conversion.

Intuitionistic Type Theory

- Several intuitionistic or constructive type theories have been developed.

- Examples:
  - Martin-Löf's **Intuitionistic Type Theory** (1980).
  - Coquand and Huet's **Calculus of Constructions** (1984).

- Many intuitionistic type theories exploit the Curry-Howard Formulas-as-Types Isomorphism.
  - Formulas serve as types or specifications.
  - Terms serve as proofs or programs.

Formalizations of Set Theory

- The standard formalization of set theory is known as **Zermelo-Fraenkel (ZF) set theory** [Zermelo, 1908].

- Other major formalizations:
  - **von-Neumann-Bernays-Gödel (NBG) set theory** [von Neumann, 1925].
  - **Morse-Kelley (MK) set theory** [Kelley, 1955].
  - **Tarski-Grothendieck set theory** [Tarski, 1938].
  - **New Foundations (NF)** [Quine, 1937].

ZF

- Proposed by Zermelo in 1908.
  - Developed to avoid the set-theoretic paradoxes.
  - Improvements made by Fraenkel (1922) and Skolem (1923).

- ZF is formalized as a theory in first-order logic.
  - Language contains two predicate symbols $=$ and $\in$.
  - Not finitely axiomatizable.

- Proper classes (e.g., the collection of all sets) are not first-class objects.
  - They cannot be denoted by terms.
  - They are used in the metatheory.
  - They can be denoted by predicate symbols.

- ZF is an exceedingly rich theory.

Axioms of ZF

1. Extensionality.

2. Foundation.

3. Comprehension scheme.

4. Pairing.

5. Union.

6. Replacement scheme.

7. Powerset.

8. Infinity.

9. Choice.

NBG

- Proposed by von Neumann in 1925.
  - Improvements made by R. Robinson (1937), Bernays (1937–54), and Gödel (1940).

- NBG is formalized as a theory in first-order logic.
  - Has the same language as ZF.
  - Finitely axiomatizable.

- Proper classes are first-class objects.

- NBG is closely related to ZF.
  - NBG is consistent iff ZF is consistent.
  - NBG and ZF share the same intuitive model of the iterated hierarchy of sets.
  - NBG and ZF have very similar axioms.

Some Proposed Practice-Oriented Logics

- **LUTINS**, the IMPS logic.
  - Version of Church's type theory with undefinedness and a subtype system.

- **BESTT**, a Basic Extended Simple Type Theory.
  - Version of Church's type theory with undefinedness, tuples, lists, and sets.

- **STMM**, a Set Theory for Mechanized Mathematics.
  - Version of NBG set theory with undefinedness and types.