CS 773 Winter 2002

05. Alternative Logics

Instructor: W. M. Farmer

Revised: 25 January 2002

Candidates for an Alternative Logic

- Partial First-Order Logic (PFOL) ⋆
- Higher-order logic
  - 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
- Hybrid logics
  - LUTINS (IMPS logic) ⋆
  - Basic Extended Simple Type Theory (BESTT) ⋆
  - A Set Theory for Mechanized Mathematics (STMM) ⋆

Traditional Approach to Partial Functions and Undefinedness

- **Expressions may be undefined**
  - Constants, variables, λ-expressions are always defined
  - Definite descriptions may be undefined: $(\mathrm{I}x : \mathbf{R} \,.\, x * x = 2)$
  - Functions may be partial and thus their applications may be undefined: $1/0$, $\sqrt{-1}$
  - An application of a function is undefined if any argument is undefined: $0 * (1/0)$
- **Formulas are always true or false**
  - Predicates must be total
  - An application of a predicate is false if any argument is undefined: $1/0 = 1/0$

Partial First-Order Logic (PFOL)

- Version of first-order logic
  - Usual logical constants: $=, \neg, \wedge, \vee, \supset, \equiv, \forall, \exists$
  - Definite description operator: I
- Semantics is based on the traditional approach to partial functions and undefinedness
  - Terms may be undefined
  - Formulas are always true or false
- Undefined terms are indiscernible
  - PFOL is a "logic of definedness", not a "logic of existence"
- The new machinery—partial functions and definite descriptions—is purely a convenience and is eliminable

What is Higher-Order Logic?

- Higher-order functions (or predicates) can be represented by terms
  - Note: A function $f : A \to B$ is **higher-order** if A or B contains functions
- Quantified variables can range over functions
- Types or sorts are used to:
  - Organize the functions of the logic
  - Control the formation of expressions
  - Classify expressions by value

Type Theory

- A higher-order logic can be viewed as a theory of types
- Russell introduced a logic called the **Theory of Types (TT)** in 1908 to serve as a foundation for mathematics
  - Included a hierarchy of types to avoid set-theoretic paradoxes like Russell's Paradox
  - Employed as the logic of Whitehead and Russell's *Principia Mathematica*
  - Not used today due to its high complexity
- Carnap, Chwistek, Ramsey and others suggested a simplified version of TT called **Simple Type Theory (STT)** in the 1920s
  - A formulation of STT with lambda-notation was introduced by Church in 1940

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

Syntax of STT

- The set $\mathcal{T}$ of **types** of STT is defined inductively by:
  - ι (type of individuals) is a type
  - $*$ (type of truth values) is a type
  - If α and β are types, then $(\alpha \to \beta)$ is a type
- A **language** of STT is a tuple $L = (\mathcal{V}, \mathcal{C}, \tau)$ where:
  - $\mathcal{V}$ is an infinite set of **variables**
  - $\mathcal{C}$ is a set of **constants**
  - $\tau : \mathcal{V} \cup \mathcal{C} \to \mathcal{T}$ is a total function
- The set of **expressions** of L is defined inductively

Semantics of STT

- Three kinds of semantics:
  - Standard: Terms are defined, formulas are two-valued
  - Partial: Terms may be undefined, formulas are two-valued
  - Three-valued: Formulas are three-valued
- A **model** of a language $L = (\mathcal{V}, \mathcal{C}, \tau)$ of STT is a pair $M = (\mathcal{D}, I)$ where:
  - $\mathcal{D} = \{D_\alpha : \alpha \in \mathcal{T}\}$
  - D_ι is nonempty and $D_* = \{\mathsf{T}, \mathsf{F}\}$
  - $D_{(\alpha \rightarrow \beta)}$ is the set of functions from D_α to D_β
  - I maps each $c \in \mathcal{C}$ to an element of $D_{\tau(c)}$
- The **valuation** function for L in M is defined inductively

Extensions to STT

- Types
  - Additional type constants and constructors
  - Type variables
  - Subtypes
- Expressions
  - Additional expression constants and constructors
  - Multivariate functions

What is Set Theory?

- Based on two simple notions:
  - Set
  - Membership
- Nearly all mathematical concepts can be expressed in terms of set and membership
- Set theory is, at least among mathematicians, the most popular foundation for mathematics
- There are many different formalizations of set theory

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