An Introduction to Higher Order Logic

Conden Chao[*]

Logic Frontier Problems Seminar II, February 28, 2017

Abstract

We mainly talked about the following topics: syntax and Tarskian semantics of (monadic) second order logic; advantages and disadvantages of (monadic) second order logic; concepts of k'th(finite and transfinite) order logic; formalisms for second(k'th and finite) order logic; some ontology considerations about higher order logic; some meta-theory but non-logical applications of higher order logic in practice.

Key Words: second order logic, higher order logic, syntax, semantics

1 Introduction

Higher order logics, long considered by many to be an esoteric subject, are increasingly recognized for their foundational importance and practical usefulness, notably in Theoretical Computer Science.

We'll try to cover most aspects of higher order logic which are attractive to most of us. We'll mainly focus on the following questions:

- What is higher order logic?
- What advantages and disadvantages does higher order logic have contrast with first order logic?
- What effects does higher order logic have in common(possibly, mathematical and logical) practice?
- Also some other questions people may concern.

2 Second Order Logic(SOL)

2.1 Syntax for SOL

Definition 2.1. The symbols for second order language $\mathscr{L}_2$ consists of two parts.

(1) Logical symbols are consists of a set $\mathfrak{V} = \mathfrak{V}_1 \cup \mathfrak{V}_2 \cup \mathfrak{V}_3$ of variables where $\mathfrak{V}_1 = \{x, y, z, \cdots\}$, $\mathfrak{V}_2 = \{f, g, h, \cdots\}$ and $\mathfrak{V}_3 = \{p, q, r, \cdots\}$; a set $\{=\}$ of equality symbol; a set $\{\neg, \wedge\}$ of connectives and a set $\{\forall\}$ of quantifier.

[*]2014-2018, Department of Philosophy, Peking University. Supervisor: Zhuanghu Liu.

(2) Non-logical symbols are consist of three sets $\mathfrak{C}, \mathfrak{P}, \mathfrak{F}$, where $\mathfrak{C}, \mathfrak{P}, \mathfrak{F}$ are the sets of constant, predicate and function symbols respectively; an arity function $\pi_1 : \mathfrak{P} \cup \mathfrak{F} \to \omega$; a variable arity function $\pi_2 : \mathfrak{V}_2 \cup \mathfrak{V}_3 \to \omega$. Further, we call the set $\mathrm{sig}\mathscr{L}_2 = \mathfrak{C} \cup \mathfrak{P} \cup \mathfrak{F}$ the signature for $\mathscr{L}_2$.

Definition 2.2. The set of terms for $\mathscr{L}_2$ is the smallest set T such that

(1) for any variable $x \in \mathfrak{V}_1$ and $c \in \mathfrak{C}$, the expressions $x, c \in T$;

(2) for any n-ary $f \in \mathfrak{V}_2$, if $\tau_0, \cdots, \tau_{n-1}$ are terms, then $f(\tau_0, \cdots, \tau_{n-1}) \in T$;

(3) for any n-ary $F \in \mathfrak{F}$, if $\tau_0, \cdots, \tau_{n-1}$ are terms, then $F(\tau_0, \cdots, \tau_{n-1}) \in T$.

Definition 2.3. The set of formulas for $\mathscr{L}_2$ is the smallest set L satisfying

(1) if τ_0, τ_1 are terms, then $\tau_0 = \tau_1 \in L$;

(2) for any n-ary $p \in \mathfrak{V}_3$, if $\tau_0, \cdots, \tau_{n-1}$ are terms, then $p(\tau_0, \cdots, \tau_{n-1}) \in L$;

(3) for any n-ary $P \in \mathfrak{P}$, if $\tau_0, \cdots, \tau_{n-1}$ are terms, then $P(\tau_0, \cdots, \tau_{n-1}) \in L$;

(4) if $\phi \in L$, then $\neg\phi \in L$;

(5) if $\phi, \psi \in L$, then $\phi \wedge \psi \in L$;

(6) if $\phi \in L$ and $x, f, p \in \mathfrak{V}$, then $\forall x\phi, \forall f\phi, \forall p\phi \in L$.

Remark 2.4. Under the usual (classical, extensional) reading of functions and relations, these two concepts are of course reducible to each other: k-ary functions can be viewed as a special kind of $k+1$-ary relations, and k-ary relations can be interpreted by their k-ary characteristic functions. For instance, a formula $\exists f(\cdots X(f(\tau))\cdots)$ is interpreted by $\exists F(\forall x \exists! y F(x,y) \wedge (\cdots(\exists y(F(\tau,y) \wedge X(y))\cdots))$ (F fresh). Conversely, a formula $\forall X(\cdots X(t)\cdots)$ can be interpreted by $\forall f(\cdots(f(\tau) = c)\cdots)$, where c is a fixed constant symbol (and f is fresh). It therefore suffices to formulate second order logic using only relation-variables, or only function variables.

2.2 Tarskian Semantics for SOL

$\mathscr{L}_2$ structures $\mathcal{M} = (M, I)$ are defined the same as first order structures.

Definition 2.5. For any $\mathscr{L}_2$ structure $\mathcal{M} = (M, I)$, the $\mathcal{M}$ assignment is a function $\nu : \mathfrak{V} \to \bigcup_{m<\omega} M^m$ such that $\nu(x) \in M$ for all $x \in \mathfrak{V}_1$, $\nu(f)$ is a function from M^m into M for all $f \in \mathfrak{V}_2$, and $\nu(p) \subseteq M^m$ for all $p \in \mathfrak{V}_3$.

To define the semantics for second order logic, we also need to give the interpretations for second order languages terms.

Definition 2.6. For any term τ and any $\mathcal{M}$-assignment ν, $\overline{\nu}(\tau)$ is defined by recursion:

(1) if $\tau = x$ for some $x \in \mathfrak{V}_1$, then $\overline{\nu}(\tau) = \overline{\nu}(x) = \nu(x)$;

(2) if $\tau = (f(\tau_0, \cdots, \tau_{n-1}))$, then $\overline{\nu}(\tau) = \nu(f)(\overline{\nu}(\tau_0), \cdots, \overline{\nu}(\tau_{n-1}))$.

(3) if $\tau = (F(\tau_0, \cdots, \tau_{n-1}))$, then $\overline{\nu}(\tau) = I(F)(\overline{\nu}(\tau_0), \cdots, \overline{\nu}(\tau_{n-1}))$.

Definition 2.7. For any formula ϕ, any $\mathscr{L}_2$ structure $\mathcal{M} = (M, I)$ and any $\mathcal{M}$-assignment ν, $(\mathcal{M}, \nu) \vDash \phi$ is defined by recursion:

(1) if $\phi = (\tau_0 = \tau_1)$, then $(\mathcal{M}, \nu) \vDash \phi$ iff $\overline{\nu}(\tau_0) = \overline{\nu}(\tau_1)$;

(2) if $\phi = p(\tau_0, \cdots, \tau_{n-1})$, then $(\mathcal{M}, \nu) \vDash \phi$ iff $(\overline{\nu}(\tau_0), \cdots, \overline{\nu}(\tau_{n-1})) \in \nu(p)$;

(3) if $\phi = P(\tau_0, \cdots, \tau_{n-1})$, then $(\mathcal{M}, \nu) \vDash \phi$ iff $(\overline{\nu}(\tau_0), \cdots, \overline{\nu}(\tau_{n-1})) \in I(P)$;

(4) if $\phi = \neg\psi$, then $(\mathcal{M}, \nu) \vDash \phi$ iff $(\mathcal{M}, \nu) \nvDash \psi$;

(5) if $\phi = \psi \wedge \theta$, then $(\mathcal{M}, \nu) \vDash \phi$ iff $(\mathcal{M}, \nu) \vDash \psi$ and$(\mathcal{M}, \nu) \vDash \theta$;

(6) if $\phi = \forall x\psi$, then $(\mathcal{M}, \nu) \vDash \phi$ iff for all $a \in M$ we have $(\mathcal{M}, \nu_{x\to a}) \vDash \psi$;

(7) if $\phi = \forall f\psi$ where f has arity n, then $(\mathcal{M}, \nu) \vDash \phi$ iff for all n-ary function F from M^n into M, we have $(\mathcal{M}, \nu_{f\to F}) \vDash \psi$;

(8) if $\phi = \forall p\psi$ where p has arity n, then $(\mathcal{M}, \nu) \vDash \phi$ iff for all n-ary predicate P on M^n, we have $(\mathcal{M}, \nu_{p\to P}) \vDash \psi$.

2.3 Expressive Power

Second order languages have a stronger expressive power than first order languages, for example:

Theorem 2.8. *The collection of finite structures is not first order definable (by a set of sentences or a single sentence), but it is definable already by a single second order sentence.*

Proof. (1) By Compactness.

(2) A statement that holds exactly of the finite structures is 'every injection is a surjection', i.e.,

$$\sigma_{\text{finite}} =_{df} \forall f(\text{Inj}(f) \to \text{Surj}(f)),$$

where $\text{Inj}(f) =_{df} \forall x, y(f(x) = f(y) \to x = y)$ and $\text{Surj}(f) =_{df} \forall y\exists x f(x) = y$. □

Another example is

Theorem 2.9. $\text{Th}(\mathcal{N})$ *isn't definable in first order arithmetic language but definable in second order arithmetic language.*

3 Monadic Second Order Logic(MSOL)

3.1 Syntax for MSOL

Definition 3.1. The symbols for second order language $\mathscr{L}_{2m}$ consists of two parts.

(1) Logical symbols are consists of a set $\mathfrak{V} = \mathfrak{V}_1 \cup \mathfrak{V}_2$ of variables where $\mathfrak{V}_1 = \{x, y, z, \cdots\}$ and $\mathfrak{V}_2 = \{X, Y, Z, \cdots\}$; a set $\{=\}$ of equality symbol; a set $\{\neg, \wedge\}$ of connectives and a set $\{\forall\}$ of quantifier.

(2) Non-logical symbols are consist of three sets $\mathfrak{C}, \mathfrak{P}, \mathfrak{F}$, where $\mathfrak{C}, \mathfrak{P}, \mathfrak{F}$ are the sets of constant, predicate and function symbols respectively; an arity function $\pi : \mathfrak{P} \cup \mathfrak{F} \to \omega$. Further, we call the set $\mathrm{sig}\mathscr{L}_{2\mathrm{m}} = \mathfrak{C} \cup \mathfrak{P} \cup \mathfrak{F}$ the signature for $\mathscr{L}_{2m}$.

Definition 3.2. The set of terms for $\mathscr{L}_{2m}$ is the smallest set T such that

(1) for any variable $x \in \mathfrak{V}_1$ and $c \in \mathfrak{C}$, the expressions $x, c \in T$;

(2) for any n-ary $F \in \mathfrak{F}$, if $\tau_0, \cdots, \tau_{n-1}$ are terms, then $F(\tau_0, \cdots, \tau_{n-1}) \in T$.

Definition 3.3. The set of formulas for $\mathscr{L}_{2m}$ is the smallest set L satisfying

(1) if τ_0, τ_1 are terms, then $\tau_0 = \tau_1 \in L$;

(2) for any $X \in \mathfrak{V}_2$ and term τ, the expression $X\tau \in L$;

(3) for any n-ary $P \in \mathfrak{P}$, if $\tau_0, \cdots, \tau_{n-1}$ are terms, then $P(\tau_0, \cdots, \tau_{n-1}) \in L$;

(4) if $\phi \in L$, then $\neg\phi \in L$;

(5) if $\phi, \psi \in L$, then $\phi \wedge \psi \in L$;

(6) if $\phi \in L$ and $x, X \in \mathfrak{V}$, then $\forall x\phi, \forall X\phi \in L$.

3.2 Tarskian Semantics for MSOL

$\mathscr{L}_{2m}$ structures $\mathcal{M} = (M, I)$ are defined the same as first order structures.

Definition 3.4. For any $\mathscr{L}_{2m}$ structure $\mathcal{M} = (M, I)$, the $\mathcal{M}$ assignment is a function $\nu : \mathfrak{V} \to M \cup \wp(M)$ such that $\nu(x) \in M$ for all $x \in \mathfrak{V}_1$ and $\nu(X) \in \wp(M)$ for all $X \in \mathfrak{V}_2$.

To define the semantics for monadic second order logic, we also need to give the interpretations for monadic second order languages terms.

Definition 3.5. For any term τ and any $\mathcal{M}$-assignment ν, $\overline{\nu}(\tau)$ is defined by recursion:

(1) if $\tau = x$ for some $x \in \mathfrak{V}_1$, then $\overline{\nu}(\tau) = \overline{\nu}(x) = \nu(x)$;

(2) if $\tau = (F(\tau_0, \cdots, \tau_{n-1}))$, then $\overline{\nu}(\tau) = I(F)(\overline{\nu}(\tau_0), \cdots, \overline{\nu}(\tau_{n-1}))$.

Definition 3.6. For any formula ϕ, any $\mathscr{L}_{2m}$ structure $\mathcal{M} = (M, I)$ and any $\mathcal{M}$-assignment ν, $(\mathcal{M}, \nu) \vDash \phi$ is defined by recursion:

(1) if $\phi = (\tau_0 = \tau_1)$, then $(\mathcal{M}, \nu) \vDash \phi$ iff $\overline{\nu}(\tau_0) = \overline{\nu}(\tau_1)$;

(2) if $\phi = X\tau$, then $(\mathcal{M}, \nu) \vDash \phi$ iff $\overline{\nu}(\tau) \in \nu(X)$;

(3) if $\phi = P(\tau_0, \cdots, \tau_{n-1})$, then $(\mathcal{M}, \nu) \vDash \phi$ iff $(\overline{\nu}(\tau_0), \cdots, \overline{\nu}(\tau_{n-1})) \in I(P)$;

(4) if $\phi = \neg\psi$, then $(\mathcal{M}, \nu) \vDash \phi$ iff $(\mathcal{M}, \nu) \nvDash \psi$;

(5) if $\phi = \psi \wedge \theta$, then $(\mathcal{M}, \nu) \vDash \phi$ iff $(\mathcal{M}, \nu) \vDash \psi$ and$(\mathcal{M}, \nu) \vDash \theta$;

(6) if $\phi = \forall x\psi$, then $(\mathcal{M}, \nu) \vDash \phi$ iff for all $a \in M$ we have $(\mathcal{M}, \nu_{x \to a}) \vDash \psi$;

(7) if $\phi = \forall X\psi$, then $(\mathcal{M}, \nu) \vDash \phi$ iff for all $A \in \wp(M)$, we have $(\mathcal{M}, \nu_{X \to A}) \vDash \psi$.

3.3 Meta-theoretical Properties for MSOL

Fact 3.7. *Monadic second order logic has no compactness property.*

Example 3.8. Let $\mathscr{L}^1_{2m}$ be the language with one binary relation symbol $<$ and countable constant symbols $c_i (i < \omega)$, and let σ_{SLO} be the sentence which says that $<$ is a strict linear ordering. Define

$$\sigma_{\text{WF}} =_{df} \forall X(\exists y\, Xy \to \exists y[Xy \wedge \neg\exists z(Xz \wedge z < y)]),$$

and let $\sigma_{\text{WO}} =_{df} \sigma_{\text{SLO}} \wedge \sigma_{\text{WF}}$. Consider the set

$$\Delta = \{\sigma_{\text{WO}}\} \cup \{c_{i+1} < c_i \mid i < \omega\}.$$

Clearly every finite subset of Δ has a model, i.e., $(m, <, 0, \cdots, m-1)$ for some $m < \omega$, but Δ has no models.

Fact 3.9. *Monadic second order logic has no downward Löwenheim-Skolem property.*

Example 3.10. Let $\mathscr{L}^2_{2m}$ be the language with only one relation symbols $<$, and let σ_{DLO} be the sentence which says $<$ is a dense linear ordering without endpoints. Define

$$\sigma_{\text{CP}} =_{df} \forall X\Big[\exists y\, Xy \wedge \exists z \forall y(Xy \to y \leq z) \to \exists z\big(\forall y(Xy \to y \leq z) \wedge \forall w[\forall y(Xy \to y \leq w) \to z \leq w]\big)\Big].$$

Then set $\sigma_{\text{CDLO}} =_{df} \sigma_{\text{DLO}} \wedge \sigma_{\text{CP}}$. It's easy to see that models of σ_{CDLO} are exactly structures in which $<$ is a *complete dense linear ordering without endpoints*, and that σ_{CDLO} has models of size 2^ω and larger but none smaller models.

Fact 3.11. *Monadic second order logic has no upward Löwenheim-Skolem property.*

Example 3.12. Let $\mathscr{L}^3_{2m}$ be any language which includes a unary symbol S and a constant symbol 0. Define

$$\sigma_{\text{IA}} =_{df} \forall X[(X0 \wedge \forall y(Xy \to XSy)) \to \forall y\, Xy]$$

Clearly σ_{IA} has models with any finite or countable size, but has no uncountable models.

Further,

- the set of valid first order sentences is recursively enumerable, while the set of valid second order sentences isn't;
- the unification problem is decidable for first order logic, while not for second order logic;
- first order logic satisfies *Beth's definability property*, *Keisler-Shelah Theorem* and *0-1 law*, while second order logic doesn't.

More about this see [1, pp.29-31]. Therefore, many people insist on avoiding higher order logic.

4 Higher Order Logic(HOL)

Higher order logic is an extension of second order logic in which quantification is used over higher order ≥ 2 predicates and functions.

4.1 Types

Definition 4.1. Types are syntactic expressions recursively generated by:

(1) ι is a type;

(2) if $\upsilon_0, \cdots, \upsilon_{k-1}$ $(k \geq 0)$ are types then $\upsilon = (\upsilon_0, \cdots, \upsilon_{k-1})$ is a type.

In particular, the type $()$ is denoted as $\varnothing$, and if $\upsilon_0 = \cdots = \upsilon_{k-1} = \eta$ then $(\upsilon_0, \cdots, \upsilon_{k-1})$ is denoted as η^k.

The language of higher order logic, for each type υ, variables of type υ and quantifiers over them.

We intend ι to denote the set of individuals, i,e., structures elements, and $(\upsilon_0, \cdots, \upsilon_{k-1})$ the set of k-ary relations between k objects of types $\upsilon_0, \cdots, \upsilon_{k-1}$. Exactly,

Definition 4.2. For a set M and any type υ, the set M_υ is defined by recursion

$$\begin{aligned} M_\iota &= M, \\ M_{(\upsilon_0,\cdots,\upsilon_{k-1})} &= \wp(\textstyle\prod_{i=0}^{k-1} M_{\upsilon_i}) = \wp(M_{\upsilon_0} \times \cdots \times M_{\upsilon_{k-1}}). \end{aligned}$$

Definition 4.3. Higher order logic has υ-terms for each type υ as follows

(1) a variable of type υ is a term of type υ;

(2) if F is a k-ary function symbol, and $\tau_0, \cdots, \tau_{k-1}$ are terms of type ι, then $F(\tau_0, \cdots, \tau_{k-1})$ is a term of type ι;

(3) if P is a k-ary predicate symbol, then P is a term of type ι^k;

(4) if t is a term of type $(\tau_0, \cdots, \tau_{k-1})$, and $t_0, \cdots, t_{k-1}$ are terms of types $\upsilon_0, \cdots, \upsilon_{k-1}$ respectively, then $t(t_0, \cdots, t_{k-1})$ is a term of type $\varnothing$.

The terms of type $\varnothing$ are the atomic formulas, and compound formulas are generated as in first order and second order logic.

The semantics of formulas of higher order logic is defined formally like for second order logic, except that an assignment over a structure with universe M maps variables of type υ to objects in M_υ.

4.2 Finite Order Logic

Definition 4.4. Types naturally fall into *orders* which could be define by recursion:

$$\begin{aligned} \text{order}(\iota) &= 1; \\ \text{order}((\upsilon_0, \cdots, \upsilon_{k-1})) &= 1 + \max\{\text{order}(\upsilon_0), \cdots, \text{order}(\upsilon_{k-1})\}. \end{aligned}$$

Definition 4.5. The fragment of higher order logic in which of types variables are of order $\leq k$ is dubbed *k'th order logic*, and The fragment of higher order logic in which of types variables are of finite order is dubbed *finite order logic*.

The construction of types can be extended to allow types whose orders are transfinite ordinals. For example, let ω consist of all objects of finite type. Then (ω, ω) is the type of binary relations between objects of finite types.

5 Formalisms for Higher Order Logic

5.1 Formalisms have no Gödel's completeness

Theorem 5.1. *The set of valid second order formulas is not definable (under canonical numeric coding) by a second order formula in the language of arithmetic. It is therefore not recursively enumerable; in particular, there is no effective formalism whose theorems are precisely the valid second order formulas.*

So second order logic does not have a complete deductive calculus. But it does have sound deductive calculi that are logically natural, powerful, of metamathematical interest, and suitable for the formalization of much of mathematics, of computer science, and of cognitive science.

5.2 Basic Formalisms

A natural formalism for the relational variant of second order logic uses the usual axioms for first order logic, with quantifier rules applying to relational variables as well as individual variables (we give a precise formulation momentarily), with the stipulation that the range of relation variables includes at least all the relations definable by the formulas of the language(in which there are only predicate variables but nor fucntion variables).

- Quantification rules for the predicate variables are:
  - the axiom schema of universal instantiation: $(\forall p\phi) \to [q/p]\phi$ where $\pi_2(p) = \pi_2(q)$.
  - the inference rule of universal generalization: if $\psi \to [q/p]\phi$, then $\psi \to \forall p\phi$ where q is not free in ψ.
- This stipulation is a form of the Comprehension Principle of Set Theory, and is formally rendered by the schema: $\exists p\forall x_1 \cdots x_n(p(x_1, \cdots, x_n) \leftrightarrow \phi)$ where $n \geq 0$, p is an n-ary predicate variable, and ϕ is a second order formula in which p isn't free.

Now we give an outline for the formalisms for second order logic for which the language has function predicate variables at the same time. For the axioms, there are five kinds in total:

- Propositional Axioms:
  - $\varphi \to (\psi \to \varphi)$;
  - $(\varphi \to \psi \to \vartheta) \to (\varphi \to \psi) \to (\varphi \to \vartheta)$;
  - $(\neg\varphi \to \psi) \to (\neg\varphi \to \neg\psi) \to \varphi$.
- Substitution Axioms:
  - $\forall x\varphi \to [\tau/x]\varphi$ where $\varphi(x;\tau)$ is a free substitution;
  - $\forall f\varphi \to [g/f]\varphi$ where $\pi_2(f) = \pi_2(g)$;
  - $\forall p\varphi \to [q/p]\varphi$ where $\pi_2(f) = \pi_2(g)$.
- Distribution Axioms:
  - $\forall x(\varphi \to \psi) \to \forall x\varphi \to \forall x\psi$;

  - $\forall f(\varphi \to \psi) \to \forall f\varphi \to \forall f\psi$;
  - $\forall p(\varphi \to \psi) \to \forall p\varphi \to \forall p\psi$.

- Equality Axioms:
  - $\tau{=}\tau$;
  - $\tau_0{=}\sigma_0 \to \cdots \to \tau_{n-1}{=}\sigma_{n-1} \to f(\tau_0, \cdots, \tau_{n-1}){=}f(\sigma_0, \cdots, \sigma_{n-1})$;
  - $\tau_0{=}\sigma_0 \to \cdots \to \tau_{n-1}{=}\sigma_{n-1} \to F(\tau_0, \cdots, \tau_{n-1}){=}F(\sigma_0, \cdots, \sigma_{n-1})$;
  - $\tau_0{=}\sigma_0 \to \cdots \to \tau_{n-1}{=}\sigma_{n-1} \to p(\tau_0, \cdots, \tau_{n-1}) \to p(\sigma_0, \cdots, \sigma_{n-1})$;
  - $\tau_0{=}\sigma_0 \to \cdots \to \tau_{n-1}{=}\sigma_{n-1} \to P(\tau_0, \cdots, \tau_{n-1}) \to P(\sigma_0, \cdots, \sigma_{n-1})$.

- Comprehension Axioms:
  - $\varphi \to \forall x\varphi$ where x is not free in φ;
  - $\varphi \to \forall f\varphi$ where f is not free in φ;
  - $\varphi \to \forall p\varphi$ where p is not free in φ;
  - $\exists f\forall x_1 \cdots x_n \exists! y(f(x_1, \cdots, x_n) = y \leftrightarrow \phi)$ where $n \geq 0$, f is an n-ary predicate variable, and ϕ is a second order formula in which f isn't free.
  - $\exists p\forall x_1 \cdots x_n(p(x_1, \cdots, x_n) \leftrightarrow \phi)$ where $n \geq 0$, p is an n-ary predicate variable, and ϕ is a second order formula in which p isn't free.
  - $\forall x_0 \cdots \forall x_{n-1}\varphi$ where φ is an axiom with one of **all the above forms.**

For the inference rules we have

- Generalization Rules:
  - if $\psi \to [\tau/x]\phi$, then $\psi \to \forall x\phi$ where $[\tau/x]\phi$ is a free substitution;
  - if $\psi \to [g/f]\phi$, then $\psi \to \forall f\phi$ where g is not free in ψ;
  - if $\psi \to [q/p]\phi$, then $\psi \to \forall p\phi$ where q is not free in ψ;

- Modus Ponens: if $\vdash \varphi$ and $\vdash \varphi \to \psi$, then we have $\vdash \psi$.

We note that for the generalization rules are not necessary.

Formalisms for k-th order logic are the same as second order logic except the comprehension principle which is called comprehension for all types of order $\leq k$.

If $x_0, \cdots, x_{k-1}$ are variable of type $\upsilon_0, \cdots, \upsilon_{k-1}$ respectively, $\upsilon = (\upsilon_0, \cdots, \upsilon_{k-1})$ and p is a variable of type υ, then comprehension at type υ is the schema: $\exists p\forall x_1 \cdots x_n(p(x_1, \cdots, x_n) \leftrightarrow \phi)$ where p isn't free.

Formalisms for finite order logic are the same as second order logic except the comprehension principle which is called comprehension for all types.

6 Ontology Considerations

6.1 Is second order logic truly a logic?

Is second order logic truly a logic?

On a technical level the answer is trivially positive. From a model-theoretic viewpoint second order logic is merely one of many possible logics since its syntax and semantics are similar to other logics such as first order logic.

From a philosophically ontological angle the answer is less clear.

Quine(1970): it's a mathematical theory rather than a logic.

- From a philosophical viewpoint, we might wish to reserve the term 'logic' to a priori concepts and truths, ones that do not depend on experience and observation.
- Quine's test: the demarcation between logic and mathematics is determined by *ontological neutrality*, that is, on not assuming the existence of certain objects and structures.In particular, if the notion of *infinity* is delineated by a formalism, then that formalism is mathematical rather than logical. Notably that as above 'infinity' could be defined by $\neg\sigma_{\text{finite}}$ a second order sentence.
- Lindström's test: compactness and downward Löwenheim-Skolem property. A landmark theorem of Lindström states, roughly, that Out of all logics, first order logic is characterized as the maximal logic that is both compact and satisfies the downward Löwenheim-Skolem property.
- From Quine's viewpoint, these two characteristics of first order logic are indeed litmus tests for being a logic: compactness is the failure to distinguish between the finite and the infinite, and downwards Löwenheim-Skolem is the failure to distinguish between different infinities.

Hazen objects Quine.

- David Lewis (quoted by Hazen, 1989) has argued that even monadic third order logic is ontologically neutral.
- Hazen showed that that formalism suffices to interpret all of second order logic.

Girard supports Quine.

- Girard's proof-theoretic criterion: a true logic must be amenable to a cut-free sequential calculus without axioms, which satisfies the subformula property.
- The underlying intuition is that neither axioms nor rules should allow 'communication' between formulas other than by rules that explicate the logical constants in isolation.
- This criterion is clearly related to Quine's ontological neutrality. Indeed, even relatively weak fragments of second order logic fail Girard's criterion.

6.2 Slipping first order to second order logic

Meta-mathematics of first order logic are in fact second order supported by the following two points:

- Hilbert and Ackermann discovered that the most basic notions of the meta-mathematics of first order logic are second order.
  - To say that a formula ϕ is valid is to say that it is true in all interpretations, a statement involving a universal quantification over universes as well as over the relations interpreting predicate letters in ϕ.
  - So the notion of 'arbitrary relation' is implicit already in first order logic (though not in first order languages used to describe and prove properties of particular first order structures).
- Moreover, there is a logical construct that seems even more ontologically benign than relational quantification, and which yields nonetheless the full expressive power of second order logic, namely *partially order quantifiers*.
  - For instance, consider the formula $\forall x \exists y \forall u \exists v \phi$. It states that for all x and u, ϕ can be made true by suitably choosing y depending on the value of x, and choosing v depending on u.
  - More generally, we may define a partially ordered quantifier to be a triple $Q = (\vec{x}; \vec{y}; \beta)$, where $\vec{x}$, $\vec{y}$ are tuples of variables, all distinct, and is a function assigning to each variable in $\vec{y}$ a sublist of $\vec{x}$.
  - If ϕ is a formula whose semantics is defined, then the semantics of $Q\phi$ is: for all $\vec{x}$ one can find values for each variable y among $\vec{y}$, depending only on the values of the variables in $\beta(y)$ (i.e. invariant with respect to changes in values of the remaining x's), which make ϕ true.

6.3 Henkin's Semantics and Henkin's Completeness

By developing some new kind of Henkin's semantics which allows non-canonical interpretations, Henkin shows that finite order logic satisfies some Henkin's completeness which is different from Gödel's completeness.

Theorem 6.1 (Henkin's Completeness). *A finite order formula ϕ is true in all Henkin-prestructures iff ϕ^T is provable in first order logic from Θ_ω.*

More about this see [1, pp.34-38]. Clearly, Henkin's construction also reduces finite order logic to a first order theory.

6.4 Finite order logic as a second order logic

In fact, without allowing non-cannonical interpretations we can also reduce finite order logic to a second order logic.

Theorem 6.2. *A formula ϕ of finite order logic is valid iff the corresponding Σ^1_1 formula $\chi_\phi \rightarrow \phi^T$ is valid.*

More about this see [1, pp.39-40]. So it's enough to have second order logic other than k'th order and finite order logic.

7 Some Other Topics

7.1 Second order logic and reverse mathematics

- As we know, second order arithmetic supplies a frame for reverse mathematics.

7.2 Higher order aspects of set theory

- Whether the widespread insistence on avoiding direct reference to higher order constructs is justified?
- (Whitehead and Russell: *Ramified Type Theory*) Recall the construction of cumulative universe of sets: $V_0 = \varnothing$, $V_{\alpha+1} = \wp(V_\alpha)$, $V_\delta = \bigcup_{\alpha<\delta} V_\alpha$ for all limit ordinal δ, and $V = \bigcup_{\alpha \text{ an ordinal}} V_\alpha$.
- As known to deal with the 3rd fundamental conflict of mathematics, second order versions of ZF emerged as being of interest both as frameworks for formalizing set theory and other parts of mathematics, and as relevant to the meta-theory of set theory itself.
- The main rationale of second order set theories is to permit at least reference to arbitrary collections, even when these are of 'unmanageable size'. Because both first order and second order objects are 'classes of sets,' it is customary to refer to second order extensions of ZF and to related theories as *theories of classes*.
- Among the theories of classes there are variants of second order ZF with comprehension restricted to first order formulas, and variants with full comprehension.
- (von Neumann, 1925; Bernays, 1937 & 1958; Gödel, 1940; Mostowski, 1939) NBG: the theory is rephrased as a first order theory of classes, in which the notion of a set is definable (x is a set iff $x \in a$ for some class a), and comprehension is restricted to formulas where all quantifiers are bounded to sets. Note that in NBG we may state the principle of global choice, asserting the existence of a class acting as a global choice function for all sets in the universe: $\exists C \forall x(x \neq \varnothing \to \exists! y((x, y) \in C \wedge y \in x))$.
- (Wang, 1949; Quine, 1951; Tarski, 1981; Kelley, 1955; Morse, 1965) KM: second order ZF with full comprehension. The relation of KM to ZF is analogous to the relation of full analysis to first order arithmetic: KM proves reflection for ZF, and is therefore not a conservative extension thereof.
- Higher order logic has also impacted the development of set theory itself, notably concerning the status of strong infinity axioms.
- The most important results here were inspired by the Montague-Lévy reflection principle: ϕ is true in V iff ϕ is true in V_κ for some cardinal κ.

- The principle is interiorized as an axiom schema in a very powerful extension of KM, due to Bernays(1976). Various forms of this principle imply the existence of ever larger cardinals, including measurable cardinals, which do not seem to have a clear justification on the basis of first order closure conditions.

7.3 Higher order logic in relation to computing and programing

Higher order logic's fundamental effects on computing and programming are mainly as follows:

- the very use of higher order data;
- the computational nature of natural deduction for higher order logic;
- the use of higher order logic in the meta-theory of formal systems;
- the relations between second and higher order logic and computational complexity.

Most notable is our omission of a legion of uses of higher order constructs in computer science, whose relation to logic is less direct.

References

[1] J. Van Benthem and K. Doets. *Higher-Order Logic*. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, Volume 164 of the series Synthese Library, pp. 275-329, Springer, 1983.

[2] P. G. Hinman. *Fundamentals of Mathematical Logic*. pp. 276-293, A K Peters, Ltd, 2005.

[3] D. Leivant. *Higher Order Logic*. In D. Gabbay, C. J. Hogger and J. A. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, Volume 1 of Logical Foundations, Oxford Science Publications, 1993.