1 Second- and Higher-Order Logic

Second-order logic is a form of logic in which the rules of statement formation permit quantification over relations and properties, in addition to the quantification over individuals allowed in first-order logic. Thus, for example, the statement

Napoleon won every battle he fought before Waterloo
is first-order, while the statement
Napoleon had all the properties of a great general
is second-order.

In *higher-order logic*, the idea underlying second-order logic is extended to embrace quantification over higher-order entities such as properties of properties and relations between relations.

1.1 The Syntax and Semantics of Second-Order Logic

The vocabulary on which second-order logic[1] is based is an extension of that of first-order logic. A typical vocabulary for first-order logic – a *first-order vocabulary* – consists of the following symbols:

- *(Individual) constant symbols a, b, c, …*
- *(Individual) variables x, y, z, …*
- *Relation symbols P, Q, …*. Each relation symbol is assigned a natural number $n \geq 1$ called its *multiplicity.* A relation symbol of multiplicity n will be called *n-ary.* A 1-ary relation symbol is called a *predicate symbol*
- *Function symbols f, g, h, …*. Each such symbol is assigned a number $n \geq 1$ called its *multiplicity.* An operation symbol of multiplicity n will be said to be *n-ary*. A 1- or 2-ary function symbol is called *unary* or *binary*
- *Logical operators* $\wedge$, $\vee$, $\neg$, $\rightarrow$, and $\leftrightarrow$
- *Quantifiers* $\exists$ and $\forall$
- *Equality symbol* =
- *Punctuation symbols* (,) and [,]

Relation symbols are also called *second-level constants.*

The *terms* of our logical vocabulary are now defined as follows:

(i) Any variable or name standing alone is a term.
(ii) If f is an n-ary operation symbol and $t_1, \ldots, t_n$ are n terms, then $f\, t_1 \ldots, t_n$ is a term.
(iii) Nothing is a term unless it follows from **(i)** and **(ii)** that it is so.

[1] Second-order logic first appears explicitly in Frege's *Begriffsschrift* (1879).

Formulas in a first-order vocabulary are defined as follows:

1. The following are formulas: **(i)** an n-ary relation symbol followed by n terms and **(ii)** any expression of the form $s = t$, where s and t are terms. These are known as *atomic* formulas.
2. If φ and ψ are formulas,[2] so are ¬φ, φ ∧ ψ φ ∨ ψ φ → ψ, and φ ↔ ψ.
3. If φ is a formula and x a variable, then both ∃x φ and ∀x φ are formulas. In each of these formulas, the (occurrence of the) variable is said to be *bound.* A(n) (occurrence of a) variable in a formula that is not bound is called *free.*
4. Nothing counts as a formula unless its being so follows from clauses **1** to **3**.

A *sentence* is a formula in which each variable is bound in the aforementioned sense. We write $s \neq t$ as an abbreviation for $\neg(s = t)$.

Clause **3** encapsulates the 'first-order' nature of a first-order vocabulary, for it licenses quantification over individual variables, which may be considered *first-level* entities. A *second-order vocabulary*, or a *vocabulary for second-order logic*, is an extension of a first-order vocabulary that also licenses quantification over *second-level* entities such as relations. To be precise, a second-order vocabulary is obtained by adding to a first-order vocabulary a collection of symbols $X, Y, Z, \ldots$ called *relations* or *second-level variables*.[3] Each relation variable is assigned a natural number $n \geq 1$ called its *multiplicity.* A relation variable of multiplicity n will be called *n-ary,* and a 1-ary relation variable is called a *predicate variable.* The formulas of a second-order vocabulary are defined by expanding clause **1(i)** to admit as a formula any n-ary relation variable followed by n terms and expanding clause **3** to admit as formulas $(\exists X)\varphi$ and $(\forall X)\varphi$ for any formula φ and relation variable X. Free and bound variables and sentences are defined as in the first-order case.

A second-order vocabulary may be considered as a *many-sorted* first-order vocabulary. In a many-sorted first-order vocabulary, one is given a collection of entities called *sorts*, and each variable or constant is assigned a particular sort. Quantification is then restricted to each sort. Thus, in a second-order vocabulary, the collection for each $n \geq 1$ of n-ary relation variables may be taken to constitute a separate sort and second-order quantification as first-order quantification over the relevant sort.

We proceed to describe the *semantics* of second-order logic. Roughly speaking, this is an extension of the semantics of first-order logic in which the relation variables are interpreted as *set-theoretic relations* over the domain of interpretation of the individual variables. In particular, predicate variables are interpreted as *subsets* of the domain of interpretation.

[2] We shall use lower-case Greek letters φ, ψ, α, β, γ to denote arbitrary formulas.

[3] A second-order vocabulary can also contain *function variables.* We shall not consider this possibility here.

We recall the semantics of first-order logic. This is based on the concept of a *structure.* Given a first-order vocabulary **L**, let us suppose for specificity that the individual variables of **L** are enumerated as a list $v_0, v_1, \ldots, v_n, \ldots$, that the relation symbols, function symbols, and individual constants of **L** are presented as indexed lists $(P_i, i \in I)$, $(f_j j \in J)$, $(c_k, k \in K)$, respectively, and that for each $i \in I$, $j \in J$, the multiplicities of P_i, f_j are the natural numbers $n(i)$, $m(j)$, respectively. Then an **L**-*structure* (also called an *interpretation* of **L**) is a triple

$$\mathfrak{A} = (A, \{R_i : i \in I\}, \{g_j : j \in J\}, \{e_k : k \in K\}),$$

where A, the *domain* or *universe* of $\mathfrak{A}$, is a *non-empty* set; $\{R_i : i \in I\}$ is an indexed family of relations on A, where, for each $i \in I$, R_i is $n(i)$-ary;[4] $\{g_j : j \in J\}$, is an indexed family of operations[5] on A, where each g_j is $m(j)$-ary; and $\{e_k : k \in k\}$ is an indexed set of elements of A – the *designated elements* of $\mathfrak{A}$. We call R_i, g_j, e_k the *denotations* of P_i, f_j, c_k, respectively, in $\mathfrak{A}$.

Now let $\mathbf{a} = (a_0, a_1, \ldots)$ be a countable sequence of elements of A (such a sequence will be referred to as henceforth as an A-*sequence*). For any term t, we define its *interpretation* $t^{(\mathfrak{A},a)}$ *in* $(\mathfrak{A}, a)$ as follows:

(i) $c_k^{(\mathfrak{A},a)} = e_k^6$
(ii) $v_n^{(\mathfrak{A},a)} = a_n$
(iii) For $j \in J$, terms $t_1, \ldots, t_{m(j)}$, $f_j t_1 \ldots t_{m(j)}^{(\mathfrak{A},a)} = g_j\left(t_1^{(\mathfrak{A},a)}, \ldots, t_{m(j)}^{(\mathfrak{A},a)}\right)$

For a natural number n and $b \in A$, we define

$$[n|b]\boldsymbol{a} = (a_0, a_1, \ldots, a_{n-1}, b, a_{n+1}, \ldots).$$

For a formula φ, we define the relation ***a*** *satisfies* φ *in* $\mathfrak{A}$,

$$\mathfrak{A} \vDash_a \varphi,$$

as follows:

1) for terms t and u,

$$\mathfrak{A} \vDash_a t = u \Leftrightarrow t^{(\mathfrak{A},a)} = u^{(\mathfrak{A},a)}$$

2) for terms $t_1, \ldots, t_{n(i)}$,

$$\mathfrak{A} \vDash_a P_i t_1 \ldots t_{n(i)} \Leftrightarrow R_i\left(t_1^{(\mathfrak{A},a)}, \ldots, t_{n(i)}^{(\mathfrak{A},a)}\right)$$

3) $\mathfrak{A} \vDash_a \neg\varphi \Leftrightarrow$ not $\mathfrak{A} \vDash_a \varphi$
4) $\mathfrak{A} \vDash_a \varphi \wedge \psi \Leftrightarrow \mathfrak{A} \vDash_a \varphi$ and $\mathfrak{A} \vDash_a \psi$

[4] When $n(i) = 1$, R_i is a *property* defined on A, which may be identified with a *subset* of A.
[5] An *n-ary operation* on a set A is a function $A^n = A \times \ldots \times A \to A$.
[6] Thus, the interpretation of c_k in ($\mathfrak{A}$, a) is just its denotation in $\mathfrak{A}$.

Cambridge University Press
978-1-108-98690-8 — Higher-Order Logic and Type Theory
John L. Bell
Excerpt
More Information

5) $\mathfrak{A} \vDash_a \varphi \vee \psi \Leftrightarrow \mathfrak{A} \vDash_a \varphi$ or $\mathfrak{A} \vDash_a \psi$
6) $\mathfrak{A} \vDash_a \varphi \rightarrow \psi \Leftrightarrow$ if $\mathfrak{A} \vDash_a \varphi$, then $\mathfrak{A} \vDash_a \psi$
7) $\mathfrak{A} \vDash_a \varphi \leftrightarrow \psi \Leftrightarrow \mathfrak{A} \vDash_a \varphi$ if and only if $\mathfrak{A} \vDash_a \psi$
8) $\mathfrak{A} \vDash_a \exists v_n \varphi \Leftrightarrow$ for some $b \in A$, $\mathfrak{A} \vDash_{[n|\,b]\alpha} \varphi$
9) $\mathfrak{A} \vDash_a \forall v_n \varphi \Leftrightarrow$ for all $b \in A$, $\mathfrak{A} \vDash_{[n|\,b]\alpha} \varphi$

A formula φ is *true* in $\mathfrak{A}$ if $\mathfrak{A} \vDash_a \varphi$ for *every* A-sequence ***a*** and *satisfiable* in $\mathfrak{A}$ if $\mathfrak{A} \vDash_a \varphi$ for *some* A-sequence ***a***. For sentences, satisfiability and truth in a structure coincide. If a sentence σ is true in $\mathfrak{A}$, we write $\mathfrak{A} \vDash \sigma$ and call $\mathfrak{A}$ a *model* of σ. If $\mathfrak{A} \vDash \neg\sigma$, we say that σ is *false* in $\mathfrak{A}$. If Σ is a set of sentences of **L,** $\mathfrak{A}$ is a *model of* Σ, written $\mathfrak{A} \vDash \Sigma$, if each member of Σ is true in $\mathfrak{A}$. The sentence σ is a *(first-order) logical consequence of* Σ, written $\Sigma \vDash \sigma$, if σ is true in every model of Σ. σ is (logically) *valid* if $\varnothing \vDash \sigma$, that is, if σ is true in *every* interpretation of **L**.

Now the semantics of first-order logic is readily extended to second-order logic. Given a second-order vocabulary **L**′ extending a first-order vocabulary **L**, we suppose that for each $n \geq 1$, the n-ary relation variables of **L**′ are enumerated as a list $V_0^{(n)},\ V_1^{(n)},\ \ldots$. If A is a set, an L'***-sequence of relations on*** A is a double sequence $\mathbf{R} = \left(R_m^{(n)} : m = 0,\ 1, \ldots\ n = 1,\ 2, \ldots\right)$ of relations on A such that for each n, $R_m^{(n)}$ is n-ary.

For a natural number n and an n-ary relation Q on A, we define $[m|Q]\boldsymbol{R}$ to be the result of replacing $R_m^{(n)}$ by Q in ***R.***

If $\mathfrak{A}$ is an **L**-structure and ***R*** an **L**′-sequence of relations on A, we extend the notion of satisfaction to **L**′-formulas by means of the rules (in which, for simplicity, we have suppressed reference to the A-sequence ***a***):

10) $\mathfrak{A} \vDash_{\boldsymbol{R}} \exists V_m^{(n)} \varphi \Leftrightarrow$ for some n-ary relation Q on A, $\mathfrak{A} \vDash_{[m|\,Q]\boldsymbol{R}} \varphi$
11) $\mathfrak{A} \vDash_{\boldsymbol{R}} \forall V_m(n) \varphi \Leftrightarrow$ for any n-ary relation Q on A, $\mathfrak{A} \vDash_{[m|\,Q]\boldsymbol{R}} \varphi$

Clauses 10 and 11 constitute the core of the idea of a *second-order interpretation.* Clause 11 in particular asserts that in a second-order interpretation, a universal second-order quantifier '∀X' is understood to mean 'for *all* relations or subsets X of the domain'.

The notions of truth, satisfiability model, and logical consequence then extend automatically to second-order sentences.

1.2 The Expressive Power of Second-Order Logic: Second-Order Arithmetic

Second-order logic has vastly more expressive power than first-order logic. For example, students of mathematical logic soon come to learn that the property of having a finite domain is not expressible in first-order terms, that is, there is no set Σ of first-order sentences such that the models of Σ are precisely the structures with finite domains. By contrast, the property of having a finite domain can be expressed by the single second-order sentence as follows:

Fin $\forall X[[\forall x\exists! yX(x,y) \wedge \forall x\forall y\forall z[X(x,y) \wedge X(z,y) \rightarrow x = z] \rightarrow \forall y\exists xX(x,y)]]$,

where X is a binary relation variable and $\exists! y\varphi(y)$ is an abbreviation for the sentence $\exists y\forall x[\varphi(x) \leftrightarrow x = y]$, which expresses 'there is a unique y such that $\varphi(y)$'. Thus, **Fin** says that 'any binary relation which is the graph of an injective function of the domain into itself is surjective'. And this holds if and only if the domain is finite.

Second-Order Arithmetic. Mathematical concepts are often presented by means of *postulates* (sometimes called *axioms*) formulated as sentences of first- or second-order logic. In writing such sentences, it is customary to place binary operation symbols between arguments rather than in front of them: thus, for example, one writes $x + y$ instead of $+xy$.

The logical vocabulary for *arithmetic* includes a unary function symbol s, two binary function symbols $+$ and $\times$, and an individual constant 0. The *standard interpretation* $\mathfrak{N}$ of this vocabulary is the structure based on the familiar *natural number system*, specified as follows:

domain of $\mathfrak{N}$: the set $N = \{0, 1, 2, \ldots\}$ of natural numbers
denotation in $\mathfrak{N}$ of s: the (immediate) successor operation $\bullet + 1$ on N
denotations in $\mathfrak{N}$ of $+$ and $\times$: the usual operations of addition and multiplication on N
denotation in $\mathfrak{N}$ of *0*: the natural number zero

The domain and successor operation of the standard interpretation may be represented by the following *diagram*:

$$(*) \qquad 0 \rightarrow \mathbf{1} \rightarrow 2 \rightarrow 3 \rightarrow \ldots$$

in which each arrow proceeds from an element to its successor.

The postulates for *basic first-order arithmetic* (**BFOA)** are the following:

B1 $\forall x\forall y(x \neq y \rightarrow sx \neq sy)$
B2 $\forall x \;\; 0 \neq sx$
B3 $\forall x(x \neq 0 \rightarrow \exists y(x = sy))$
B4 $\forall x \;\; x + 0 = x$
B5 $\forall x\forall y \; x + sy = s(x + y)$
B6 $\forall x \;\; x \times 0 = 0$
B7 $\forall x\forall y \;\; x \times sy = (x \times y) + x.$

Each of these postulates is true in $\mathfrak{N}$. The first three express familiar facts about the successor operation:

B1 *Distinct natural numbers have distinct successors.*
B2 *Zero is the successor of no natural number.*
B3 *Every non-zero natural number is a successor.*

The next two postulates tell us how to add in this notation:

B4 *Adding 0 has no effect.*
B5 $(x + y) + 1 = x + (y + 1)$.

In this notation, each *numeral* 1, 2, 3, 4, … is represented by a string of s's of the appropriate length followed by 0, as in

$$1 = s0 \ 2 = ss0 \ 3 = sss0 \ 4 = ssss0, \ldots.$$

BFOA has a property known as *incompleteness*, which means that there are certain sentences true in the standard interpretation $\mathfrak{B}$ that are *not (first-order) logical consequences* of **BFOA** or are, simply, *independent* of **BFOA**. For instance, although each of the sentences

$$0 \neq s0, s0 \neq ss0, ss0 \neq sss0, \ \ldots$$

is a logical consequence of **BFOA**, the corresponding generalization

(a) $\forall x \ x \neq sx$

is *not.* Similarly, *none* of the following generalizations is a logical consequence of (**BFOA**), though each of their particular instances is:

(b) $\forall x \quad 0 + x = x$
(c) $\forall x \forall y \forall z \quad x + (y + z) = (x + y) + z$
(d) $\forall x \forall y \quad x + y = y + x$
(e) $\forall x \quad 0 \times x = 0$
(f) $\forall x \forall y \quad sx \times y = (x \times y) + y$
(g) $\forall x \forall y \quad x \times y = y \times x$

To establish the independence of (a)–(g) from the postulates of **BFOA**, we must supply a model of **BFOA**, that is, an interpretation in which **B1–B7** are true, but in which (a)–(g) are *false.* It is not difficult to check that the following interpretation $\mathfrak{B}$ does the job:

domain of $\mathfrak{B}$: the set of natural numbers together with two additional distinct objects \$ and @ *denotation of s*: indicated by the following diagram, in which each arrow leads from a member of the domain to its successor:

(**) $0 \to 1 \to 2 \to 3 \to \ldots$ \$ @

denotations of + and ×: these are as usual when the arguments are both natural numbers. When one or both arguments is/are \$ or @, the values are given by the following tables, in which n is any natural number and $n^>$ is any non-zero natural number:

+	*n*	\$	@
N		@	\$
\$	\$	@	\$
@	@	@	\$

×	0	$n^>$	\$	@
n	0	\$	@	
\$	0	@	@	@
@	0	\$	\$	\$

The incompleteness of **BFOA** basic implies that it also fails to be *categorical.* A set of postulates is said to be *categorical* if all of its models are *isomorphic* (Greek *iso:* 'same' and *morphe:* 'form') in the sense that the *same* diagram serves for all of them, apart from the relabelling of nodes.[7] The non-categoricity of basic arithmetic can be seen immediately from the fact that the standard interpretation $\mathfrak{N}$ is not isomorphic to the interpretation $\mathfrak{B}$ defined previously. For no relabelling of nodes can ever convert $\mathfrak{N}$'s diagram (*) into $\mathfrak{B}$'s diagram (**), since the latter contains loops and the former doesn't.

The incompleteness of **BFOA** is a kind of deductive weakness: certain arithmetical sentences that one would expect to be logical consequences of it follow turn out not to be. This weakness can be overcome by adding to it a *second-order* postulate known as the *Principle of Mathematical Induction.* Informally, this is the rule of arithmetic that states:

> *for any property* **P** *of natural numbers, if* 0 *has the property* **P** *and if, for any number x, that x* + 1 *has the property* **P** *follows from the assumption that x has the property* **P**, *then every number has the property* **P**.

This can be expressed by the second-order sentence

Ind $\forall P[[P0 \wedge \forall x(Px \to Psx)] \to \forall x Px]$,

where P is a predicate variable.

Basic second-order arithmetic (BSOA) is obtained by adding **Ind** to **BFOA.**

We have seen that **BFOA** has models that differ in essential respects from the standard interpretation $\mathfrak{N}$. But its second-order extension **BSOA** does not suffer from this defect. In fact, **BSOA** is categorical; all of its models are

[7] To be precise, an *isomorphism* between two **L**-structures $\mathfrak{A} = (A, \{R_i : j \in I\}, \{g_i : j \in J\}, \{e_k : k \in K\})$ and $\mathfrak{A}' = (A', \{R_i' : j \in I\}, \{g_i' : j \in J\}, \{e'_k : k \in K\})$ is a bijective function $F : A \to A'$ satisfying, for each $i \in I$, $j \in J$, $k \in K$, the conditions $F(e_k) = e'_k$, $F[Ri] = R'I$, for any $a_1, \ldots, a_{m(j)} \in A, g_j'(Fa_1, \ldots, Fa_{m(j)}) = F(g_j(a_1, \ldots, a_{m(j)}))$.

isomorphic to $\mathfrak{N}$. The first, and crucial, step in demonstrating this is to establish what we shall call the 'exhaustion principle'.

Exhaustion principle. *Models of* **Ind** *are exactly those interpretations in which (the interpretations of the terms on) the list 0, s0, ss0, sss0, ... exhausts the whole domain of the interpretation.*

To see this, suppose that $\mathfrak{A}$ is a model of **Ind**. Let $M = \{0^{\mathfrak{A}}, (s0)^{\mathfrak{A}}, (ss0)^{\mathfrak{A}}, \ldots\}$ and let P be a predicate symbol such that $P^{\mathfrak{A}} = M$. (We may assume without loss of generality that such exists by simply adding a new predicate symbol to **L** and interpreting it as M.) Then $P0$ and $\forall x(Px \to P_{Sx})$ are both true in $\mathfrak{A}$. Since $\mathfrak{A} \models$ **Ind**, it follows that $\forall xPx$ is true in $\mathfrak{A}$. But the truth of this means precisely that $A = M$.

Conversely, suppose that the domain A of an interpretation $\mathfrak{A}$ coincides with $\{0^{\mathfrak{A}}, (s0)^{\mathfrak{A}}, (ss0)^{\mathfrak{A}}, \ldots\}$. Let **P** be any property defined on A, and let P be a predicate symbol such that $P^{\mathfrak{A}} =$ **P**. (Again, we can always add a new predicate symbol to **L** and interpret it as U.) Now assume that $P0$ and $\forall x(Px \to Psx)$ are both true in $\mathfrak{A}$. We claim that $\forall xPx$ is also true in $\mathfrak{A}$. If not, then some element of A fails to satisfy $P^{\mathfrak{A}}$. Since $P0$ is true in $\mathfrak{A}$, this element cannot be $0^{\mathfrak{A}}$ and so is of the form $\left(s^n 0\right)^{A}$ for some $n \geq 1$ (here, $s^n 0$ is 0 preceded by n s's). Let n be the least number such that $\left(s^n 0\right)^{\mathfrak{A}}$ fails to satisfy $P^{\mathfrak{A}}$. Then $n \geq 1$ and $Ps^{n-1}0$ is true in $\mathfrak{A}$. Since $\forall x(Px \to Psx)$ is true in $\mathfrak{A}$, it follows that $Ps^{n-1}0 \to \mathrm{P}s^{n}0$ is true in $\mathfrak{A}$, and hence, $Ps^n 0$ is true in $\mathfrak{A}$. This contradicts the choice of n, and it follows that $\forall xPx$ must have been true in $\mathfrak{A}$ after all. Accordingly,

$$P0 \wedge \forall x(Px \to Psx) \to \forall xPx$$

is true in $\mathfrak{A}$; since the interpretation of P was an arbitrary property defined on A, we conclude that the **Ind** is true in $\mathfrak{A}$.

Second-order successor arithmetic (**SOSA**) is defined to be the weakened version of **BSOA** whose postulates are **B1**, **B2**, and **Ind.** We show that **SOSA** *is categorical; each of its models being isomorphic to* $\mathfrak{N}$.

To prove this, let $\mathfrak{A}$ be a model of **SOSA**. Then, by the exhaustion principle, the domain of $\mathfrak{A}$ consists of the interpretations of the terms on the list

$$(*) \quad 0,\ \mathrm{s}0,\ \mathrm{ss}0,\ \mathrm{sss}0, \ldots$$

The truth of **B2** in $\mathfrak{A}$ implies that the sentences $0 \neq \mathrm{s}0,\ 0 \neq \mathrm{ss}0\ , 0 \neq \mathrm{sss}0$ are all true in $\mathfrak{A}$. It now follows from the truth of **B1** in $\mathfrak{A}$ that distinct members of the list (*) receive distinct interpretations in $\mathfrak{A}$. (For if not, then, for example, $sss0 = \mathrm{sssss}0$ would be true in $\mathfrak{A}$, and three applications of **B1** would show $0 = \mathrm{ss}0$ to be true in $\mathfrak{A}$, contradicting what we have already established.) It follows that the diagram of $\mathfrak{A}$ looks like:

$0^{\mathfrak{A}} \rightarrow 1^{\mathfrak{A}} \rightarrow 2^{\mathfrak{A}} \rightarrow 3^{\mathfrak{A}} \rightarrow \ldots .$

Clearly, this diagram can be relabelled so as to convert it into the diagram of the standard interpretation $\mathfrak{N}$, namely,

$0 \rightarrow \mathbf{1} \rightarrow 2 \rightarrow 3 \rightarrow \ \ldots .$

Therefore, $\mathfrak{A}$ and $\mathfrak{N}$ are isomorphic.

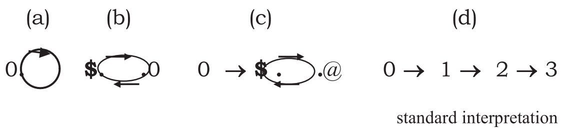
By complicating this argument, it can be shown that each model of **BSOA** is isomorphic to $\mathfrak{N}$ so that **BSOA** is also categorical.

The categoricity of **BSOA** means that – unlike **BFOA** – it furnishes a *complete characterization* of the natural number system in the following sense:

> *For any sentence* σ *in the vocabulary of* **BSOA**, σ *is a logical consequence of* **BSOA** *if and only if* σ *is true in the standard interpretation* $\mathfrak{N}$.

To prove this, we observe that if σ is a logical consequence of **BSOA**, it must be true in every model of it, and so in particular, it must be true in $\mathfrak{N}$. Conversely, suppose σ is true in $\mathfrak{N}$, and let $\mathfrak{A}$ be any model of **BSOA**. Since **BSOA** is categorical, $\mathfrak{A}$ is isomorphic to $\mathfrak{N}$, so since σ is true in $\mathfrak{N}$, it must also be true in $\mathfrak{A}$. Therefore, σ is a logical consequence of **BSOA.**

We finally note that **Ind** has several non-isomorphic models, which shows that taken *by itself*, it is not categorical. These models are based on the following four diagrams, in which the denotations of *s* and *0* are displayed: as usual, each arrow goes from an element to its 'successor'.



It is evident that no one of these diagrams can be converted into another by relabelling nodes since they all contain different numbers of nodes: 1, 2, 3, ∞, respectively. The interpretations are therefore non-isomorphic.

Note that **B1** is false in interpretation (c), and **B2** is false in both (a) and (b).

1.3 The Limitations of Second-Order Logic

While second-order logic has great expressive power, that very expressive power is the source of certain limitations, which we now describe.

One of the most useful metatheorems of first-order logic is the *compactness theorem.* This states that if each finite subset of a set Σ of first-order sentences has a model, then so does Σ. But the fact that the property of having a finite domain is expressible in second-order logic leads to the failure of the compactness theorem.

This can be seen as follows. For each $n \geq 1$ write σ_n for the first-order sentence

$$\exists v_1 \exists v_2 \ldots \exists v_n (v_1 \neq v_2 \wedge \ldots \wedge v_1 \neq v_n \wedge v_2 \neq v_3 \wedge \ldots \wedge v_2 \neq v_n \wedge \ldots \wedge v_{n-1} \neq v_n)$$

(σ_n says that the domain has at least n elements). Now, let Σ be the set of sentences $\{$ **Fin**, $\sigma_1, \sigma_2, \ldots \}$ where **Fin** is the second-order sentence expressing finitude formulated earlier. Then, clearly, each finite subset of Σ has a model, but Σ itself does not. Accordingly, the compactness theorem fails for second-order logic.

The most important metatheoretical feature of first-order logic is that the *semantic* notion $\vDash$ of first-order logical consequence can be recast in a purely *syntactic* form. This can be done by furnishing first-order logic[8] with a *derivability apparatus,* thus turning into a *deductive system.* There are various ways of doing this. Here, we describe a method based on the idea of a *formal derivation* or *proof*, resting in its turn on a body of formal *axioms* and *rules of inference.*

The *axioms* and *rules of inference* for classical first-order logic in **L** are specified as follows. As *axioms,* we take all formulas of the form:

(i) $\alpha \to (\beta \to \alpha)$

(ii) $[\alpha \to (\beta \to \gamma) \to [(\alpha \to \beta) \to (\alpha \to \gamma)]$

(iii) $\alpha \to (\beta \to \alpha \wedge \beta)$

(iv) $\alpha \wedge \beta \to \alpha \qquad \alpha \wedge \beta \to \beta$

(v) $\alpha \to \alpha \vee \beta \qquad \beta \to \alpha \vee \beta$

(vi) $(\alpha \to \gamma) \to [(\beta \to \gamma) \to (\alpha \vee \beta \to \gamma)]$

(vii) $(\alpha \to \beta) \to [(\alpha \to \neg\beta) \to \neg\alpha]$

(viii) $\neg\alpha \to (\alpha \to \beta)$

(ix) $(\alpha \leftrightarrow \beta) \to [(\alpha \to \beta) \wedge (\beta \to \alpha)] \qquad [(\alpha \to \beta) \wedge (\beta \to \alpha)] \to (\alpha \leftrightarrow \beta)$

(x) $\neg\neg\alpha \to \alpha$

(xi) $\alpha(t) \to \exists x\alpha(x) \;\; \forall x\alpha(x) \to \alpha(t)$ (x free in α and t free for x in α[9])

(xii) $x = x$

(xiii) $\alpha(x) \wedge x = y \to \alpha(y)$

As *rules of inference,* we take

Modus ponens $\dfrac{\alpha, \alpha \to \beta}{\beta}$,

Quantifier Rules $\dfrac{\beta \to \alpha(x)}{\beta \to \forall x\alpha(x)} \qquad \dfrac{\alpha(x) \to \beta}{\exists x\alpha(x) \to \beta}$ (x not free in β).

[8] We emphasize that here we are concerned with *classical* logic. In Section 3, *intuitionistic* logic will enter the picture.

[9] A term t is said to be *free for x* in a formula α if no variable occurring in t becomes bound when t is substituted for x in α.