Crash Course:
Philosophy of Logic and Second-order Logic

Ted Sider
Higher-order
metaphysics

1. Introduction

2. Importance of syntax to logic

Logic is largely about logical implication—truth preservation in virtue of logical form. So statements of logical rules make reference to logical form, e.g.:

$$\frac{A \text{ and } B}{A}$$

Why isn't this an instance of the rule?:

> Daisy and Luke are siblings
>
> Therefore, Daisy

Because A and B must be *sentences*. So: logical rules use syntactic concepts (e.g., "sentence") and make syntactic assumptions (e.g., 'and' connects sentences).

Good logic depends on good assumptions about syntax. Famous example: Aristotle assumed that sentences have subject-predicate form, with a single subject and single predicate. So the best he could do with:

> Someone respects everyone
> Therefore, everyone is respected by someone (or other)

is:

> Some Fs are Gs
> Therefore, every F is an H

where F: 'is a thing', G: 'respects-everyone', H: 'is-respected-by-someone'. Not a valid syllogism.

Frege, on the other hand, allowed multi-place predicates, and introduced quantifiers, variables, and sentential connectives, thus enabling:

> $\exists x \forall y Rxy$
> $\forall y \exists x Rxy$

2.1 Syntax in formal languages

Syntax is about which strings of symbols "make sense", or are "well-formed".

In modern logic, we invent formal languages with rigorously stipulated syntax.

Typical definition of what counts as a grammatical, or "well-formed" formula:

1. "$Rt_1 \dots t_n$" is a formula, for any n-place predicate R and any n terms (i.e., names or variables), $t_1, \dots, t_n$
2. If A is a formula, so is "$\sim A$"
3. If A and B are formulas, so are "$(A \wedge B)$", "$(A \vee B)$", "$(A \rightarrow B)$", and "$(A \leftrightarrow B)$".
4. If A is a formula then so are "$\forall v A$" and "$\exists v A$", where v is any variable
5. There are no formulas other than those that can be shown to be formulas using rules 1–4

(In addition to "formula", this uses the syntactic concepts of "predicate" and "term".)

3. First- versus second-order logic

3.1 Syntax

Second-order logic allows variables in predicate position:

$\forall x Gx$	$\exists x \exists y Bxy$	(well-formed in both)
$\exists F Fa$	$\forall R(Rab \rightarrow Rba)$	(well-formed *only* in second-order logic)

This is implemented by distinguishing *individual variables* $(x, y, z \dots)$ from *predicate variables*: $F, X, R, \dots$ (which can be 1-place, 2-place, etc.), and changing the first clause in the definition of a formula:

1′. For any n-place predicate *or predicate variable* R and any n terms (i.e., names or *individual* variables), $t_1, \ldots, t_n$, "$Rt_1 \ldots t_n$" is a formula.

Gloss of some second-order sentences:

$\exists F Fa$: "a has some property"

$\forall R(Rab \rightarrow Rba)$: "$b$ bears every relation to a that a bears to b"

But distinguish the first (for example) from:

$$\exists x(Px \wedge Hax)$$

where P: 'is a property' and H: 'has' (i.e., instantiates). This is a *first*-order symbolization of "a has some property".

3.2 Formal logic and logical consequence

In mathematical logic you stipulate the meanings of concepts like "formula in language L" and "true in all interpretations".

But it's a philosophical question whether these stipulatively defined concepts are good models of intuitive notions like *logical truth* and *logical implication*.

3.3 Semantics

There are two main approaches to formally modeling notions like logical truth and logical consequence, the semantic and the proof-theoretic approach.

Semantic approach

1. Define *interpretations*—mathematically precise "pictures" of logical possibilities
2. Define the notion of a sentence's being *true in* a given interpretation
3. Use these notions to define metalogical concepts. E.g., a sentence is *valid* iff it is true in all interpretations; a set of sentences Γ *semantically implies* a sentence S iff S is true in every interpretation in which every member of Γ is true.

Intepretation
(same for both first- and second-order logic)

A nonempty set, D (the "domain"), plus an assignment of an appropriate denotation based on D to every nonlogical expression in the language. Names are assigned members of D; one-place predicates are assigned sets of members of D; two-place predicates are assigned sets of ordered pairs of D; and so on.

Definition of truth
(this part for both first- and second-order logic)

i) A sentence Fa is true in an interpretation I if and only if the denotation of the name a is a member of the denotation of the predicate F; a sentence Rab is true if and only if the ordered pair of the denotation of the name a and the denotation of the name b is a member of the denotation of the predicate R; etc.

ii) A sentence $\sim A$ is true in I if and only if A is not true in I; a sentence $A \wedge B$ is true in I if and only if A is true in I and B is true in I; etc.

iii) A sentence $\forall v A$ is true in I if and only if A is true of every member of D; a sentence $\exists v A$ is true in I if and only if A is true of some member of D.

Definition of truth, continued
(clause added for second-order logic)

iv) Where R is an n-place predicate variable, a sentence $\forall R A$ is true in I if and only if A is true of every set of n-tuples of members of D; a sentence $\exists R A$ is true in I if and only if A is true of some set of n-tuples of members of D.

3.4 Proof theory

Second main approach to modeling logical truth and logical implication:

Proof-theoretic approach (Hilbert-style)

- Choose axioms (certain chosen formulas)
- Choose rules (certain chosen relations over formulas)
- Formula S *follows via rule* R from other formulas $S_1, \ldots, S_n$ iff the formulas $S_1, \ldots, S_n, S$ stand in R.
- A *proof of* A *from* Γ is a finite series of formulas, the last of which is A, in which each formula is either i) a member of Γ, ii) an axiom, or iii) follows from earlier lines in the series by a rule
- Γ *proves* A iff there exists some proof of A from Γ
- A is a *theorem* iff there exists some proof of A from the empty set $\varnothing$

The idea is to choose axioms that are obviously logical truths, and rules that are obviously logical implications. Here is one proof-system for first-order logic:

Axioms:

$$A \to (B \to A)$$
$$(A \to (B \to C)) \to ((A \to B) \to (A \to C))$$
$$(\sim A \to \sim B) \to (B \to A)$$
$$\forall v A \to A_{v \mapsto t} \qquad \text{(universal instantiation)}$$

Rules:

$$\frac{A \qquad A \to B}{B} \quad \text{(modus ponens)} \qquad\qquad \frac{A \to B}{A \to \forall v B} \quad \text{(universal generalization)}$$

($A_{v \mapsto t}$: the result of replacing free vs in A with ts.)

These "axioms" are actually axiom *schemas*. An axiom if what you get by replacing the As, Bs, etc. with any formulas. So there are infinitely many axioms.

To get proof systems for second-order logic, we add new axioms and/or rules.

New axiom schemas

$$\forall X A \to A_{X \mapsto F} \qquad \text{(second-order universal instantiation)}$$
$$\exists X \forall v_1 \dots \forall v_n (X v_1 \dots v_n \leftrightarrow A) \qquad \text{(Comprehension)}$$

(X is any n-place predicate variable, F is any n-place predicate, and A is a schematic variable for any formula, which cannot have X free.)

Comprehension basically says that to any formula A there is a corresponding property or relation. For example, let A be this formula:

$$Hx \land \exists y(Cy \land Oxy)$$

"x is a hipster who owns at least one chicken"

Then this is an instance of Comprehension:

$$\exists X \forall x(Xx \leftrightarrow Hx \land \exists y(Cy \land Oxy))$$

"some property is had by exactly the hipsters who own at least one chicken"

3.5 Metalogic

There are dramatic metalogical differences between first- and second-order logic.

3.5.1 Completeness

Gödel proved in 1929 that:

Completeness If a first-order formula is valid then it is a theorem

It's relatively straightforward to prove the converse:

Soundness If a first-order formula is a theorem then it is valid

But for second-order logic, there is *no* sound axiomatic system that is also complete. (Caveat: we require that in axiomatic systems, there is a mechanical procedure for telling what counts as rules or axioms.)

3.5.2 Compactness

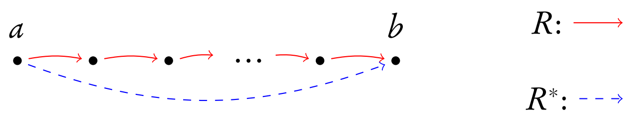
A set of formulas, Γ, is *satisfiable* if and only if there is some interpretation in which every member of Γ is true. The following holds for first-order logic:

Compactness If every finite subset of Γ is satisfiable, Γ is satisfiable.

Let's illustrate why this is important. The *ancestral* of a two-place predicate R is a two-place predicate R^*, such that:

R^*ab iff: Rab, or
Rax and Rxb, for some x, or
Rax and Rxy and Ryb, for some x and y, or ...

That is: a is an R-ancestor of b iff there is some "finite R-chain" from a to b:



There is no way to define R^* using first-order predicate logic:

Suppose for reductio that some sentence, R^*ab, of predicate logic says that a is an R-ancestor of b. Let Γ be this infinite set of sentences:

$$\Gamma = \{R^*ab, A_1, A_2, A_3, \dots\}$$

where the sentences $A_1, A_2, A_3, \dots$ are the following:

A_1: $\sim Rab$ ("There is no one-link R-chain")
A_2: $\sim \exists x(Rax \wedge Rxb)$ ("There is no two-link R-chain")
A_3: $\sim \exists x \exists y(Rax \wedge Rxy \wedge Ryb)$ ("There is no three-link R-chain")
etc.

<u>Γ is unsatisfiable</u>: in any interpretation in which R^*ab is true, there is some finite R-chain from a to b, and so one of the A_is is false.

Every finite subset of Γ is satisfiable: any finite subset of the A_is merely rules out finite chains between a and b up to some particular length (depending on the "highest" A_i in the finite subset), and R^*ab can still be true if there is a chain longer than that between a and b.

This contradicts Compactness. Therefore R^*ab can't exist.

Thus compactness tells us that the language of first-order logic is expressively weak in a certain way. (For similar reasons you can't express in first-order logic the idea that there are only finitely many things.)

But in second-order logic, compactness does *not* hold, and you *can* define the ancestral of a predicate:

$$R^*ab \leftrightarrow \forall F\Big(\big(\forall x(Rax \to Fx) \land \forall x \forall y((Fx \land Rxy) \to Fy)\big) \to Fb\Big)$$

"a is an R-ancestor of b if and only if for every property, F: IF i) everything that a bears R to is an F, and ii) whenever an F bears R to something, that something is also an F, THEN b is an F"

(i.e.,)

"a is an R-ancestor of b if and only if b has every property that i) is had by every "R-child" of a, and ii) is "closed under" R"

Second-order logic is in a sense more expressively powerful than first-order logic.

3.5.3 Clarifying differences in expressive power

But doesn't this first-order sentence define the ancestral of R?:

$$R^*ab \leftrightarrow \forall z\Big(\big(\forall x(Rax \to x \in z) \land \forall x \forall y((x \in z \land Rxy) \to y \in z)\big) \to b \in z\Big)$$

"b is a member of every set, z, that contains every R-child of x and is closed under R"

Answer: what is true is that no first-order sentence correctly defines R^* in *every interpretation*. A little more exactly:

> There is no first-order sentence R^*ab, such that in any interpretation I, R^*ab is true in I if and only if, where r is the set of ordered pairs that is denoted by R in I, there is a finite chain of members of the domain of I, pairwise connected by r, leading from the denotation of a to the denotation of b.

The set-theoretic first-order sentence doesn't define R^* correctly in interpretations in which, e.g., '$\in$' means something that has nothing to do with sets.

Moral: it's hard-wired into the semantics for second-order logic that the second-order quantifier $\forall F$ ranges over subsets of the domain, and that second-order predications Fx express set membership.

3.6 Metamathematics

3.6.1 Skolem's paradox

Call a *model* of a set of sentences an interpretation in which every sentence in the set is true. In first-order logic, the following holds:

Löwenheim-Skolem theorem If a set of sentences, Γ, has a model, it has a model whose domain is at most countably infinite.

So (e.g.), any consistent first-order axioms for set theory can be made true in an interpretation whose domain has no more elements than the natural numbers—despite the fact that you can prove in set theory that there exist sets that are larger than the set of natural numbers.

Moral: sets of first-order sentences in a sense can't pin down their intended interpretation.

The Löwenheim-Skolem theorem doesn't hold for second-order logic.

3.6.2 Nonstandard models of arithmetic

A similar moral holds for arithmetic.

First-order language of arithmetic: the first-order language with symbols 0, $'$, $+$, and $\cdot$.

Second-order language of arithmetic: the second-order language with those symbols.

Standard interpretation: the interpretation whose domain is the set of natural numbers, and in which '0' denotes the number 0, '$'$' denotes the successor (or add-one) function, '$+$' denotes the addition function, and '$\cdot$' denotes the multiplication function.

Is there a set of sentences in the first-order language of arithmetic whose *only* model is the standard interpretation?

No: you could "permute" elements in the standard interpretation.

Better question: what about a set, all of whose models are *isomorphic* to the standard interpretation?

Still No: even if you include in the set all sentences true in the standard interpretation, there will still be weird "nonstandard models", looking like this:

$$\underbrace{0, 1, 2, \ldots}_{\text{standard numbers}} \qquad \ldots \qquad \underbrace{\ldots a_{-1}, a_0, a_1, \ldots}_{\text{some nonstandard numbers}} \qquad \ldots \qquad \underbrace{\ldots b_{-1}, b_0, b_1, \ldots}_{\text{more nonstandard numbers}} \quad \ldots$$

This is another example of the expressive weakness of first-order logic—another failure to force interpretations to "look right" (as with Skolem's paradox).

But in second-order logic it's different. There is a second-order sentence all of whose models are isomorphic to the standard interpretation.

3.6.3 Schematic and nonschematic axiomatizations

Project: write down axioms for arithmetic. We'll want at least these:

$$
\begin{aligned}
&\forall x \forall y (x' = y' \to x = y)\\
&\forall x\, \mathsf{0} \neq x'\\
&\forall x (x \neq \mathsf{0} \to \exists y\, x = y')\\
&\forall x\, x + \mathsf{0} = x\\
&\forall x \forall y\, x + y' = (x + y)'\\
&\forall x\, x \cdot \mathsf{0} = \mathsf{0}\\
&\forall x \forall y\, x \cdot y' = (x \cdot y) + x
\end{aligned}
$$

But we still need something about *induction*, which says roughly:

> "if $\mathsf{0}$ has a property, and if whenever a number has that property, so does the next number, then every number has the property"

How we articulate this exactly depends on whether we're using the first- or second-order language of arithmetic.

$$\forall F\Big(\big(F\mathsf{0} \wedge \forall x(Fx \to Fx')\big) \to \forall x Fx\Big)$$

(second order induction *principle*)

$$\big(A(\mathsf{0}) \wedge \forall x(A(x) \to A(x'))\big) \to \forall x A(x)$$ (first order induction *schema*)

The schema only says that induction works for the definable properties, so to speak, whereas the second-order principle says it works for *all* properties. And really, one might argue, it's this latter thing that we want to say.