Logic

Edwin Park

2023

Contents

1 First-Order Logic **1**
1.1 Completeness Theorem 2
1.2 Incompleteness Theorems 3

1 First-Order Logic

Mathematics may be generalised as the study of structures and their properties; number theory studies properties of the natural numbers, group theory on groups, set theory on sets etc. Starting from properties of the objects we are given with (axioms), further results are deduced (theorems). This process may be formalised with a formal language, say $\mathcal{L}$, whose purpose is to formalise proofs. That is, given a set of sentences Σ in $\mathcal{L}$ which we take as axioms encoding our desired structure (groups, natural numbers, etc), there should be some algebra/arithmetic on $\mathcal{L}$ allowing us to produce correct statements about such structures starting from Σ.

In general, any first-order language includes the more "fundamental" components of: *parantheses*, *connectives* ($\wedge$, $\vee$, $\neg$, $\rightarrow$, etc.), *quantifiers* ($\forall$, $\exists$) and *variables* (an infinite set $\{v_1, v_2, \dots\}$). (These are sometimes called the "logical symbols".) Along with these are the so-called "non-logical" symbols; a selection of certain *constant*, *function* and *relation* symbols. The difference is that logical symbols always mean the same thing, while non-logical symbols may mean different things depending on one's interpretation (more on this shortly.)

For example, suppose we wanted to construct a language to study number theory. The structure in question then is the natural numbers, which we may incorporate into the language as an infinite set of constant symbols, $\{0, 1, 2, \dots\}$. But, of course, $\mathbb{N}$ by itself is meaningless without the arithmetic operations $+$ or $\leq$; without such operators, $\mathbb{N}$ would trivially be isomorphic to any countably infinite set. We introduce $+$ as a function symbol and $\leq$ as a relation symbol. Now, it is crucial to realise that the symbol 423 we introduced, or the symbol $+$, do not represent the number $423 \in \mathbb{N}$ or the operation of addition. They are just symbols in a language, which are given meaning only under a certain interpretation – that is, only once we are given a dictionary telling us what those symbols actually denote. For example, we may think that the string of five letters 'moron' means 'an idiot', but in Welsh it actually means 'carrot'. In fact, 'moron' doesn't mean anything until we specify what language we are using. As such, our constant, function and relation symbols are meaningless (until we give an interpretation). In fact, to differentiate our constant symbols to the usual numbers in $\mathbb{N}$, we will write our constant symbols with an accent: $\{\overline{0}, \overline{1}, \overline{2}, \dots\}$.

One may ask, what are the purposes of each of the non-logical symbols? The constant symbols and function symbols are used to represent objects in the structure we are working with. If the structure we are working is the integers modulo 4, our constant symbols may just be the set $\{\overline{0}, \overline{1}, \overline{2}, \overline{3}\}$. Function symbols allow us to represent objects in the structure more conveniently and succintly; for example, instead of having the infinite set $\{\overline{0}, \overline{1}, \overline{2}, \dots\}$ as constant symbols with the intent of representing $\mathbb{N}$, we may instead just have $\{\overline{0}\}$ and a function symbol S to represent $\mathbb{N}$; for example, $SSS\overline{0}$ represents 3 and so on.

Relation symbols are used to represent true-or-false statements about the structure of our structure, such as $\overline{2} \leq \overline{23}$. Without relation symbols, we wouldn't be able to make any statements about objects in our structure!

Now, we shall give an interpretation to our language $\mathcal{L}$. This is simple – we let each numeral $\overline{n}$ correspond to $n \in \mathbb{N}$, $+$ correspond to the function of addition, and $\leq$ to the less-than-or-equal-to relation on $\mathbb{N}$.

Now that we have developed the basic symbolism behind our language, we now want to define a calculus/algebra to work with and manipulate sentences – that is, we want to define a *deductive system*. A deductive system is a calculus purely operating within the language $\mathcal{L}$, with no dependence on any structure.

paragraph or two on deductive system structure

How strong is such deductive system? Firstly, the soundness theorem asserts the validity/correctness/soundness of our deductive system; that $\Sigma \vdash \phi \Rightarrow \Sigma \models \phi$. This means that any conclusion ϕ reached from a set of sentences Σ using our calculus is true in any structure, given Σ. The completeness theorem states that if a sentence ϕ is implied by a set of sentences Σ for all possible structures (i.e. $(\wedge\Sigma) \Rightarrow \phi$ is true for any structure), then there exists a proof for it in our deductive system; that $\Sigma \vdash \phi \Rightarrow \Sigma \models \phi$. (This is the converse of soundeness.) The incompleteness theorems, on the other hand, establish the limits of our deductive system in proving statements that are true in one specific structure. We will see that there are statements about the natural numbers that are true, but unprovable in our deductive system, and that such unprovable-but-true statements exist for any choice of axioms. Note that the completeness theorem is concerned with statements that follow from axioms, while the incompleteness theorem is concerned with statements that are true in the structure of the natural numbers, whether or not they follow from our axioms. Thus, the completeness theorem demonstrates the power of our deductive system with regards to axioms and their consequences, while the incompleteness theorem demonstrates the incompleteness an axiomatic system in exploring true statements in a structure.

In fact, the second incompleteness theorem gives a more concrete example of an unprovable statement (in certain deductive systems) – that is, the statement that the deductive system itsellf is consistent.

That is to say, any statement whose truth hinges on the axioms

infinitude of primes in logic?

1.1 Completeness Theorem

Definition 1.1.1. A set of $\mathcal{L}$-formulas Σ is *consistent* if a contradiction cannot be deduced from it.

Theorem 1.1.1 (Completeness). $\Sigma \models \phi \Rightarrow \Sigma \vdash \phi$.

Proof. First, we note that this is equivalent to showing $\Sigma \models \bot \Rightarrow \Sigma \vdash \bot$. Assume $\Sigma \models \phi$. Then $\Sigma \cup \neg\phi \models \bot \Rightarrow \Sigma \cup \neg\phi \vdash \bot$. By the deduction theorem, $\Sigma \cup \neg\phi \vdash \bot \Rightarrow \Sigma \vdash \neg\phi \rightarrow \bot$, and so by contradiction, $\Sigma \vdash \phi$. Taking the contrapositive, the completeness theorem is equivalent to asserting that consistency implies the existence of a model for a set of sentences (from a previous lemma, free variables can be removed).

A naive first attempt is to define a structure $\mathfrak{A}$ mirroring exactly the given language and the sentences in Σ. For example, given a language $\mathcal{L}$ with two constant symbols c_0 and c_1, no function symbols and a single unary relation R, we may have

$$\Sigma = \{R(c_0), \neg R(c_1)\}.$$

Then, to mirror this, we may construct a structure $\mathfrak{A}$ with $|\mathfrak{A}| = \{c_0, c_1\}$ and $R^{\mathfrak{A}} = \{c_0\}$[1]. But quickly such constructions break down, for example when we have

$$\Sigma = \{c_0 = c_1\}.$$

Under the interpretation of $\mathfrak{A}$ this becomes

$$c_0^{\mathfrak{A}} = c_1^{\mathfrak{A}},$$

which is false as the two are distinct elemnts in the universe $|\mathfrak{A}|$. To account for this, we can redefine the elements of the universe as equivalence classes of terms, rather than terms themselves; let $x \sim y$ iff $(x = y) \in \Sigma$. Then $|\mathfrak{A}|$ is the set of all equivalence classes of terms in $\mathcal{L}$. So in our previous example we would have $|\mathfrak{A}| = \{[c_0]\}$.

Functions may be integrated into our structure naturally; let $f^{\mathfrak{A}}([t_1], \ldots, [t_n]) = [ft_1 \ldots t_2]$.

Another problem arises when dealing with existential quantifiers. Suppose we have

$$\Sigma = \{\exists x R(x), \neg R(c_0), \neg R(c_1)\}.$$

Clearly a model exists for Σ but our construction fails to satisfy the first sentence from the set. We can blame this on the lack of symbols in $\mathcal{L}$, and solve the problem by introducing a third element $[c_2]$, and have $R^{\mathfrak{A}} = \{[c_2]\}$. However, adding arbitrary symbols as such may introduce problems; while they won't be directly referenced in Σ as they aren't in $\mathcal{L}$, they may be called upon by a quantifier, forcing us to define how the new symbol behaves with other relations and functions. For example, take

$$\Sigma = \{\exists x R(x), \neg R(c_0), \neg R(c_1), \forall x R_2(f(x))\}.$$

(we have updated $\mathcal{L}$ to include another unary relation R_2, and a unary function f.) Now, upon introducing c_2, we have to make sure that $R_2(f(c_2))$ holds. For each new symbol added we will have to check

[1] Note that c_0 and c_1 in context of the structure $\mathfrak{A}$ aren't constants nor solidly defined objects, but merely symbols. (wait rly?)

all quantifier sentences in Σ and extend the relations and functions appropriately so as to ensure they are still satisfied.

Suppose we now want to prove that our structure models Σ. A natural way is to use induction on the complexity of the sentence. Let q_n be the statement that $\sigma_n \in \Sigma \Rightarrow \mathfrak{A} \models \sigma_n$, where σ_n is a sentence of complexity n (recall that the complexity of a formula is the height of its parsing tree). We want to show q_0 and $q_n \Rightarrow q_{n+1}$. q_0 follows by construction; any expressions of the form $R(x_1, \ldots, x_n)$ in Σ are true under interpretation by $\mathfrak{A}$ by definition of $\mathfrak{A}$. The trouble is with the inductive step, $q_n \Rightarrow q_{n+1}$. Say we want to show that $\neg\sigma_n \in \Sigma$ implies $\mathfrak{A} \models \neg\sigma_n$. From the assumption we know that σ_n can't be in Σ by consistency, but we can't really go further as q_n is not bidirectional; even if $\sigma_n \notin \Sigma$, that says nothing about whether $\mathfrak{A}$ models σ_n.

The genius of Henkin's proof the completeness theorem is to make q_n bidirectional i.e. to have $\sigma_n \in \Sigma \iff \mathfrak{A} \models \sigma_n$. To do so, we need to extend Σ so that every sentence satisfied by $|\mathfrak{A}|$ is a member of it. We may simply enumerate and run through every $\mathcal{L}$-sentence, check whether it is satisfied by $\mathfrak{A}$, and add it to Σ if it is.

We can now proceed by induction. Assume q_n.

$$\begin{aligned} \neg\sigma_n \in \Sigma &\iff \sigma_n \notin \Sigma \\ &\iff \mathfrak{A} \not\models \sigma_n \\ &\iff \mathfrak{A} \models \neg\sigma_n \end{aligned}$$

$$\begin{aligned} (\sigma_n \wedge \omega_n) \in \Sigma &\iff (\sigma_n \in \Sigma) \wedge (\omega_n \in \Sigma) \\ &\iff (\mathfrak{A} \models \sigma_n) \wedge (\mathfrak{A} \models \omega_n) \\ &\iff \mathfrak{A} \models (\sigma_n \wedge \omega_n) \end{aligned}$$

$$\begin{aligned} \exists\, x,\ \phi_n \in \Sigma &\iff \exists\, c \in |\mathfrak{A}|,\ \mathfrak{A} \models \phi_n[x \to c] \\ &\iff \exists\, c \in |\mathfrak{A}|,\ \mathfrak{A} \models \exists\, x,\ \phi_n. \end{aligned}$$

Thus, $\mathfrak{A}$ models Σ. (God, this proof is sketchy af) □

1.2 Incompleteness Theorems

Definition 1.2.1. A set of non-logical axioms is *complete* if it is able to prove or refute any sentence.

Remark 1.2.1. Completeness of axioms do not follow from our completeness theorem (which was proving completeness about our deductive system, not our axioms). While it does follow that $\Sigma \not\models \phi \iff \Sigma \not\vdash \phi$, we note that $\Sigma \not\vdash \phi \not\Rightarrow \Sigma \vdash \neg\phi$; the unprovability of ϕ does not improve the provability of $\neg\phi$.

Definition 1.2.2. The *theory* of a structure, say $\mathfrak{N}$, denoted $Th(\mathfrak{N})$, is the set of all true sentences in $\mathfrak{N}$. A set of axioms is said to be an *axiomatisation* of a theory if it proves every sentence in the theory.

The key to showing Gödel's incompleteness theorem is to establish a method to generate self-referencing sentences. Self-referencing statements are prone to paradoxes such as Russell's paradox (Does the set which contains every set A such that $A \notin A$, include itself?) or the liar's paradaox. (This statement is false.)