I

A Bit of Logic: A User's Toolbox

This prerequisite chapter – what some authors call a "Chapter 0" – is an abridged version of Chapter I of volume 1 of my *Lectures in Logic and Set Theory*. It is offered here just in case that volume *Mathematical Logic* is not readily accessible.

Simply put, logic[†] is about *proofs* or *deductions*. From the point of view of the *user* of the subject – whose best interests we attempt to serve in this chapter – logic ought to be just a toolbox which one can employ to prove theorems, for example, in set theory, algebra, topology, theoretical computer science, etc.

The volume at hand is about an important specimen of a mathematical theory, or logical theory, namely, axiomatic set theory. Another significant example, which we do not study here, is arithmetic. Roughly speaking, a mathematical theory consists on one hand of assumptions that are specific to the subject matter – the so-called *axioms* – and on the other hand a toolbox of logical rules. One usually performs either of the following two activities with a mathematical theory: One may choose to work *within the theory*, that is, employ the tools and the axioms for the sole purpose of proving theorems. Or one can take the entire theory as an *object of study* and study it "from the outside" as it were, in order to pose and attempt to answer questions about the power of the theory (e.g., "does the theory have as theorems all the 'true' statements about the subject matter?"), its reliability (meaning whether it is free from contradictions or not), how its reliability is affected if you add new assumptions (axioms), etc.

Our development of set theory will involve both types of investigations indicated above:

(1) Primarily, we will act as *users* of logic in order to deduce "true" statements about sets (i.e., theorems of set theory) as consequences of certain

[†] We drop the qualifier "mathematical" from now on, as this is the only type of logic we are about.

"obviously true"[†] statements that we accept up front without proof, namely, the ZFC axioms.[‡] This is pretty much analogous to the behaviour of a geometer whose job is to prove theorems of, say, Euclidean geometry.

(2) We will also look at ZFC from the outside and address some issues of the type "is such and such a sentence (of set theory) provable from the axioms of ZFC and the rules of logic alone?"

It is evident that we need a *precise formulation* of set theory, that is, we must turn it into a *mathematical object* in order to make task (2), above, a meaningful mathematical activity.[§] This dictates that we develop logic itself *formally*, and subsequently set theory as a *formal theory*.

Formalism,[¶] roughly speaking, is the *abstraction* of the reasoning processes (proofs) achieved by deleting any references to the "truth content" of the component mathematical statements (formulas). What is important in formalist reasoning is solely the syntactic *form* of (mathematical) statements as well as that of the proofs (or deductions) within which these statements appear.

A formalist builds an *artificial language*, that is, an infinite – but *finitely specifiable*[#] – collection of "words" (meaning *symbol sequences*, also called *expressions*). He[||] then uses this language in order to build deductions – that is, *finite sequences* of words – in such a manner that, at each step, he writes down a word if and only if it is "certified" to be *syntactically correct* to do so. "Certification" is granted by a toolbox consisting of the very same rules of logic that we will present in this chapter.

The formalist may pretend, if he so chooses, that the words that appear in a proof are meaningless sequences of meaningless symbols. Nevertheless, such posturing cannot hide the fact that (in any purposefully designed theory) these

[†] We often quote a word or cluster of related words as a warning that the crude English meaning is not necessarily the intended meaning, or it may be ambiguous. For example, the first "true" in the sentence where this footnote originates is technical, but in a *first approximation* may be taken to mean what "true" means in English. "Obviously true" is an ambiguous term. Obvious to whom? However, the point is – to introduce another ambiguity – that "reasonable people" will accept the truth of the (ZFC) axioms.

[‡] This is an acronym reflecting the names of *Zermelo* and *Fraenkel* – the founders of this particular axiomatization – and the fact that the so-called axiom of *choice* is included.

[§] Here is an analogy: It is the precision of the rules for the game of chess that makes the notion of analyzing a chessboard configuration meaningful.

[¶] The person who practises formalism is a formalist.

[#] The finite specification is achieved by a finite collection of "rules", repeated applications of which build the words.

[||] *By definition*, "he", "his", "him" – and their derivatives – are gender-neutral in this volume.

words *codify* "true" (intuitively speaking) statements. Put bluntly, we must have something meaningful to talk about before we bother to codify it.

Therefore, a formal theory is a laboratory version (artificial replica or *simulation*, if you will) of a "real" mathematical theory of the type encountered in mathematics,[†] and formal proofs do unravel (codified versions of) "truths" beyond those embodied in the adopted axioms.

It will be reassuring for the uninitiated that it is a fact of logic that the totality of the "universally true" statements – that is, those that hold in all of mathematics and not only in specific theories – coincides with the totality of statements that we can *deduce purely formally* from some simple universally true *assumptions* such as $x = x$, without any reference to meaning or "truth" (Gödel's completeness theorem for first order logic). In short, in this case formal deducibility is as powerful as "truth". The flip side is that formal deducibility *cannot* be as powerful as "truth" when it is applied to *specific* mathematical theories such as set theory or arithmetic (Gödel's incompleteness theorem).

Formalization allows us to understand the deeper reasons that have prevented set theorists from settling important questions such as the *continuum hypothesis* – that is, the statement that there are no cardinalities between that of the set of natural numbers and that of the set of the reals. This understanding is gathered by "running diagnostics" on our laboratory replica of set theory. That is, just as an engineer evaluates a new airplane design by building and testing a model of the real thing, we can find out, with some startling successes, what are the limitations of our theory, that is, what our assumptions are incapable of logically implying.[‡] If the replica is well built,[§] we can then learn something about the behaviour of the real thing.

In the case of formal set theory and, for example, the question of our failure to resolve the continuum hypothesis, such diagnostics (the methods of Gödel and Cohen – see Chapters VI and VIII) return a simple answer: We have not included enough assumptions in (whether "real" or "formal") set theory to settle this question one way or another.

† Examples of "real" (non-formalized) theories are Euclid's geometry, topology, the theory of groups, and, of course, Cantor's "naïve" or "informal" set theory.

‡ In model theory "model" means exactly the opposite of what it means here. A model airplane abstracts the real thing. A model of a formal (i.e., abstract) theory is a "concrete" or "real" version of the abstract theory.

§ This is where it pays to choose reasonable assumptions, assumptions that are "obviously true".

But what about the interests of the reader who only wants to *practise* set theory, and who therefore may choose to skip the parts of this volume that just *talk about* set theory? Does, perchance, formalism put him into an unnecessary straitjacket?

We think not. Actually it is easier, and safer, to reason formally than to do so informally. The latter mode often mixes syntax and semantics (meaning), and there is always the danger that the "user" may assign incorrect (i.e., convenient, but not general) meanings to the symbols that he manipulates, a phenomenon that anyone who is teaching mathematics must have observed several times with some distress.

Another uncertainty one may encounter in an informal approach is this: "What can we allow to be a 'property' in mathematics?" This is an important question, for we often want to collect objects that share a common property, or we want to prove some property of the natural numbers by induction or by the least principle. But what *is* a property? Is colour a property? How about mood? It is not enough to say, "no, these are not properties", for these are just *two* frivolous examples. The question is how to describe accurately and unambiguously the *infinite variety* of properties that *are* allowed. Formalism can do just that.†

"Formalism for the user" is not a revolutionary slogan. It was advocated by Hilbert, the founder of formalism, partly as a means of – as he believed‡ – formulating mathematical theories in a manner that allows one to check them (i.e., run diagnostic tests on them) for freedom from contradiction,§ but also as the *right way* to do mathematics. By this proposal he hoped to salvage mathematics itself – which, Hilbert felt, was about to be destroyed by the Brouwer school of intuitionist thought. In a way, his program could bridge the gap between the classical and the intuitionist camps, and there is some evidence that Heyting (an influential intuitionist and contemporary of Hilbert) thought that such a *rapprochement* was possible. After all, since meaning is irrelevant to a formalist, all that he is doing (in a proof) is shuffling finite sequences of

† Well, almost. So-called cardinality considerations make it impossible to describe *all* "good" properties formally. But, practically and empirically speaking, we can define all that matter for "doing mathematics".

‡ This belief was unfounded, as Gödel's incompleteness theorems showed.

§ Hilbert's *metatheory* – that is, the "world" or "lab" outside the theory, where the replica is actually manufactured – was *finitary*. Thus – Hilbert believed – all this theory building and theory checking ought to be effected by finitary means. This was another ingredient that was consistent with peaceful coexistence with the intuitionists. And, alas, this ingredient was the one that – as some writers put it – destroyed Hilbert's program to found mathematics on his version of formalism. Gödel's incompleteness theorems showed that a finitary metatheory is not up to the task.

symbols, never having to handle or argue about infinite objects – a good thing, as far as an intuitionist is concerned.†

In support of the "formalism for the user" position we must not fail to mention Bourbaki's (1966a) monumental work, which is a formalization of a huge chunk of mathematics, including set theory, algebra, topology, and theory of integration. This work is strictly for the *user* of mathematics, not for the *metamathematician* who *studies* formal theories. Yet, it is fully formalized, true to the spirit of Hilbert, and it comes in a self-contained package, including a "Chapter 0" on formal logic.

More recently, the proposition of employing formal reasoning as a tool has been gaining support in a number of computer science undergraduate curricula, where logic and discrete mathematics are taught in a formalized setting, starting with a rigorous course in the two logical calculi (propositional and predicate), emphasizing the point of view of the user of logic (and mathematics) – hence with an attendant emphasis on *calculating* (i.e., writing and annotating formal) proofs. Pioneering works in this domain are the undergraduate text (1994) and the paper (1995) of Gries and Schneider.

You are urged to master the technique of writing formal proofs by studying how we go about it throughout this volume, especially in Chapter III.‡ You will find that writing and annotating formal proofs is a discipline very much like computer programming, so it cannot be that hard. Computer programming is taught in the first year, isn't it?§

† True, a formalist applies classical logic, while an intuitionist applies a different logic where, for example, double negation is not removable. Yet, unlike a Platonist, a formalist does not believe – or he does not have to disclose to his intuitionist friends that he might do – that infinite sets exist *in the metatheory*, as his tools are just finite symbol sequences. To appreciate the tension here, consider this anecdote: It is said that when Kronecker – the father of intuitionism – was informed of Lindemann's proof (1882) that π is transcendental, while he granted that this was an interesting result, he also dismissed it, suggesting that π – whose decimal expansion is, of course, infinite but not periodic – "does not exist" (see Wilder (1963, p. 193)). We do not propound the tenets of intuitionism here, but it is fair to state that infinite sets *are* possible in intuitionistic mathematics as this has later evolved in the hands of Brouwer and his Amsterdam school. However, such sets must be (like all sets of intuitionistic mathematics) *finitely generated* – just like our formal languages and the set of theorems (the latter provided that our axioms are too) – in a sense that may be familiar to some readers who have had a course in automata and language theory. See Wilder (1963, p. 234).

‡ Many additional paradigms of formal proofs, in the context of arithmetic, are found in Chapter II of volume 1 of these *Lectures*.

§ One must not gather the impression that formal proofs are just obscure sequences of symbol sequences akin to Morse code. Just as one does in computer programming, one also uses *comments* in formal proofs – that is, annotations (in English, Greek, or your favourite natural language) that aim to explain or justify for the benefit of the reader the various proof steps. At some point, when familiarity allows and the length of (formal) proofs becomes prohibitive, we agree to relax the proof style. Read on!

It is also fair to admit, in defense of "semantic reasoning", that meaning is an important tool for formulating conjectures, for analyzing a given proof in order to figure out what makes it tick, or indeed for *discovering* the proof, in rough outline, in the first place. For these very reasons we supplement many of our formal arguments in this volume with discussions that are based on intuitive semantics, and with several examples taken from informal mathematics.

We forewarn the reader of the inevitability with which the *informal* language of sets already intrudes in this chapter (as it indeed does in all mathematics). More importantly, some of the elementary results of Cantorian naïve set theory are needed here. Conversely, *formal* set theory needs the tools and some of the results developed here. This apparent "chicken or egg" phenomenon is often called "bootstrapping",† not to be confused with "circularity" – which it is not: Only informal set theory notation and results are needed here in order to found *formal* set theory.

This is a good place to summarize our grand plan:

First (in this chapter), we will formalize the rules of reasoning in general – as these apply to all mathematics – and develop their properties. We will skip the detailed study of the interaction between formalized rules and their *intended meaning* (semantics), as well as the study of the limitations of these formalized rules. Nevertheless, we will state without proof the relevant important results that come into play here, the *completeness* and *incompleteness* theorems (both due to Kurt Gödel).

Secondly (starting with the next chapter), once we have learnt *about* these tools of formalized reasoning – what they are and how to use them – we will next become *users* of formal logic so that we can discover important theorems of (or, as we say, develop) set theory. Of course, we will not forget to run a few diagnostics. For example, Chapter VIII is entirely on metamathematical issues.

Formal theories, and their artificial languages, are defined (built) and "tested" *within* informal mathematics (the latter also called "real" mathematics by Platonists). The first theory that we build here is general-purpose, or "pure", formal logic. We can then build mathematical *formal theories* (e.g., set theory) by just adding "impurities", namely, the appropriate special symbols and appropriate special assumptions (written in the artificial formal language).

We *describe* precisely how we construct these languages and theories using the usual abundance of mathematical notation, notions, and techniques available

† The term "bootstrapping" is suggestive of a person pulling himself up by his bootstraps. Reputedly, this technique, which is pervasive, among others, in the computer programming field – as alluded to in the term "booting" – was invented by Baron Münchhausen.

to us, augmented by the descriptive power of natural language (e.g., English, or Greek, or French, or German, or Russian), as particular circumstances or geography might dictate. This *milieu* within which we build, pursue, and study our theories – besides "real mathematics" – is also often called the *metatheory*, or more generally, *metamathematics*. The language we speak while at it, this *mélange* of mathematics and natural language, is the *metalanguage*.

I.1. First Order Languages

In the most abstract and thus simplest manner of describing it, a *formalized mathematical theory* (also, *formalized logical theory*) consists of the following sets of things: a set of basic or primitive symbols, $\mathscr{V}$, used to build *symbol sequences* (also called strings, or expressions, or words, *over* $\mathscr{V}$); a set of strings, **Wff**, over $\mathscr{V}$, called the *formulas* of the theory; and finally, a *subset* of **Wff**, **Thm**, the set of *theorems* of the theory.†

Well, this is the *extension* of a theory, that is, the explicit set of objects in it. How is a theory *given*?

In most cases of interest to the mathematician it is given by specifying $\mathscr{V}$ and two sets of simple rules, namely, formula-building rules and theorem-building rules. Rules from the first set allow us to build, or *generate*, **Wff** from $\mathscr{V}$. The rules of the second set generate **Thm** from **Wff**. In short (e.g., Bourbaki (1966b)), *a theory consists of an alphabet of primitive symbols and* rules *used to generate the "language of the theory" (meaning, essentially,* **Wff***) from these symbols, and some additional rules used to generate the theorems.* We expand on this below.

I.1.1 Remark. What is a rule? We run the danger of becoming circular or too pedantic if we overdefine this notion. Intuitively, the rules we have in mind are string manipulation rules – that is, "black boxes" (or functions) that receive string inputs and respond with string outputs. For example, a well-known theorem-building rule receives as input a formula and a variable, and it returns (essentially) the string composed of the symbol ∀, immediately followed by the variable and, in turn, immediately followed by the formula.‡ □

(1) First off, the (*first order*) *formal language*, L, where the theory is "spoken"§ is a triple ($\mathscr{V}$, **Term**, **Wff**), that is, it has three important components, each of them a set. $\mathscr{V}$ is the *alphabet* (or vocabulary) of the language. It is the

† For a less abstract, but more detailed view of theories see p. 39.
‡ This rule is usually called "generalization".
§ We will soon say what makes a language "first order".

collection of the *basic* syntactic "bricks" (symbols) that we use to form symbol sequences (or *expressions*) that are *terms* (members of **Term**) or *formulas* (members of **Wff**). We will ensure that the processes that build terms or formulas, using the basic building blocks in $\mathscr{V}$, are (intuitively) *algorithmic* ("mechanical"). Terms will formally codify objects, while formulas will formally codify statements about objects.

(2) *Reasoning* in the theory will be the process of discovering "true statements" about objects – that is, *theorems*. This discovery journey begins with certain formulas which codify statements that we take for granted (i.e., accept without proof as "basic truths"). Such formulas are the *axioms*. There are two types of axioms. *Special*, or *nonlogical*, axioms are to describe specific aspects of any theory that we might be building; they are "basic truths" in a restricted context. For example, "$x + 1 \neq 0$" is a special axiom that contributes towards the characterization of number theory over $\mathbb{N}$. This is a "basic truth" in the context of $\mathbb{N}$ but is certainly not true of the integers or the rationals – which is good, because we do not want to confuse $\mathbb{N}$ with the integers or the rationals. The other kind of axiom will be found in *all* theories. It is the kind that is "universally valid", that is, *not* a theory-specific truth but one that holds in all branches of mathematics (for example, "$x = x$" is such a universal truth). This is why this type of axiom will be called *logical*.

(3) Finally, we will need *rules* for reasoning, actually called *rules of inference*. These are rules that allow us to deduce, or derive, a true statement from other statements that we have already established as being true.† These rules will be chosen to be oblivious to meaning, being only conscious of *form*. They will apply to statement configurations of certain *recognizable forms* and will produce (derive) new statements of some corresponding recognizable forms (see Remark I.1.1).

I.1.2 Remark. We may think of axioms (of either logical or nonlogical type) as being special cases of rules, that is, rules that receive *no* input in order to produce an output. In this manner item (2) above is subsumed by item (3), thus we are faithful to our abstract definition of theory (where axioms were not mentioned).

An example, outside mathematics, of an inputless rule is the rule invoked when you type **date** on your computer keyboard. This rule receives no input, and outputs the current date on your screen. □

We next look carefully into (first order) formal languages.

† The generous use of the term "true" here is only meant to motivate. "Provable" or "deducible" formula, or "theorem", will be the technically precise terminology that we will soon define to replace the term "true statement".

There are two parts in each first order alphabet. The first, the collection of the *logical symbols*, is common to all first order languages (regardless of which theory is spoken in them). We describe this part immediately below.

Logical Symbols.

LS.1. *Object or individual variables*. An *object variable* is any one symbol out of the unending sequence $v_0, v_1, v_2, \ldots$. In practice – whether we are using logic as a tool or as an object of study – we agree to be sloppy with notation and use, generically, x, y, z, u, v, w with or without subscripts or primes as *names* of object variables.† This is just a matter of notational convenience. We allow ourselves to write, say, z instead of, say, $v_{1200000000000560000009}$. Object variables (*intuitively*) "vary over" (i.e., are allowed to take *values* that are) the objects that the theory studies (e.g., numbers, sets, atoms, lines, points, etc., as the case may be).

LS.2. *The Boolean or propositional connectives*. These are the symbols "$\neg$" and "$\vee$".‡ These are pronounced *not* and *or* respectively.

LS.3. *The existential quantifier*, that is, the symbol "$\exists$", pronounced *exists* or *for some*.

LS.4. *Brackets*, that is, "(" and ")".

LS.5. *The equality predicate*. This is the symbol "$=$", which we use to indicate that objects are "equal". It is pronounced *equals*.

The logical symbols will have a fixed interpretation. In particular, "$=$" will always be expected to mean *equals*.

The theory-specific part of the alphabet is not fixed, but varies from theory to theory. For example, in set theory we just add the nonlogical (or special) symbols, $\in$ and U. The first is a special *predicate symbol* (or just predicate) of *arity* 2; the second is a predicate symbol of arity 1.§

In number theory we adopt instead the special symbols S (intended meaning: successor, or "$+\,1$", function), $+$, $\times$, 0, $<$, and (sometimes) a symbol for the

† Conventions such as this one are essentially agreements – *effected in the metatheory* – on how to be sloppy and get away with it. They are offered in the interest of user-friendliness and readability. There are also theory-specific conventions, which may allow additional names in our informal (metamathematical) notation. Such examples, in set theory, occur in the following chapters.

‡ The quotes are *not* part of the symbol. They serve to indicate clearly, e.g., in the case of "$\vee$" here, what *is* part of the symbol and what is not (the following period is not).

§ "arity" is derived from "ary" of "un*ary*", "bin*ary*", etc. It denotes the number of arguments needed by a symbol according to the dictates of correct syntax. Function and predicate symbols need arguments.

CAMBRIDGE

Cambridge University Press
978-0-521-16848-9 - Lectures in Logic and Set Theory: Volume 2: Set Theory
George Tourlakis
Excerpt
More information

exponentiation operation (function) a^b. The first three are *function symbols* of arities 1, 2, and 2 respectively. 0 is a *constant symbol*, $<$ is a predicate of arity 2, and whatever symbol we might introduce to denote a^b would have arity 2.

The following list gives the general picture.

Nonlogical Symbols.

NLS.1. A (possibly empty) set of symbols for *constants*. We normally use the metasymbols† a, b, c, d, e, with or without primes or subscripts, to stand for constants unless we have in mind some alternative "standard" formal notation in specific theories (e.g., $\emptyset$, 0, ω).

NLS.2. A (possibly empty) set of symbols for *predicate symbols* or *relation symbols* for each possible arity $n > 0$. We normally use P, Q, R, generically, with or without primes or subscripts, to stand for predicate symbols. Note that $=$ is in the logical camp. Also note that theory-specific formal symbols are possible for predicates, e.g., $<, \in, U$.

NLS.3. Finally, a (possibly empty) set of symbols for *functions* for each possible arity $n > 0$. We normally use f, g, h, generically, with or without primes or subscripts, to stand for function symbols. Note that theory-specific formal symbols are possible for functions, e.g., $+, \times$.

I.1.3 Remark. (1) We have the option of assuming that each of the *logical* symbols that we named in **LS.1–LS.5** have no further structure and that the symbols are, ontologically, *identical to their names*, that is, they are just these exact signs drawn on paper (or on any equivalent display medium).

In this case, changing the symbols, say, $\neg$ and $\exists$ to $\sim$ and **E** respectively results in a "different" logic, but one that is, trivially, *isomorphic* to the one we are describing: Anything that we may do in, or say about, one logic trivially translates to an equivalent activity in, or utterance about, the other as long as we systematically carry out the translations of all occurrences of $\neg$ and $\exists$ to $\sim$ and **E** respectively (or vice versa).

An alternative point of view is that the symbol names are *not* the same as (identical with) the symbols they are naming. Thus, for example, "$\neg$" names the connective we pronounce **not**, by we do not know (or care) exactly what the nature of this connective is (we only care about how it behaves). Thus, the name "$\neg$" becomes just a typographical expedient and may be replaced by other names that name the same object, **not**.

This point of view gives one flexibility in, for example, deciding how the variable symbols are "implemented". It often is convenient to suppose that the

† *Meta*symbols are *informal* (i.e., outside the formal language) symbols that we use within "real" mathematics – the *meta*theory – in order to describe, as we are doing here, the formal language.