Introduction

In the beginning was the Word, and the Word was with God, and the Word was God. The same was also in the beginning with God.

John's Gospel, ch 1 v 1

Despite having this text by heart I still have no idea what it means. What I do know is that the word that is translated from the Greek into English as 'word' is $\lambda o \gamma o \sigma$, which also gave us the word 'logic'. It is entirely appropriate that we use a Greek word since it was the Greeks who invented logic. They also invented the axiomatic method, in which one makes basic assumptions about a topic from which one then derives conclusions.

The most striking aspect of the development of mathematics in its explosive modern phase of the last 120-odd years has been the extension of the scope of the subject matter. By this I do not mean that mathematics has been extended to new subject areas (one thinks immediately of the way in which the social sciences have been revolutionised by the discovery that the things they study can be given numerical values), even though it has, nor do I mean that new kinds of mathematical entities have been discovered (imaginary numbers, vectors and so on), true though that is too. What I mean is that in that period there was a great increase in the variety of mathematical entities that were believed to have an independent existence.

To any of the eighteenth-century mathematicians one could have begun an exposition "Let n be an integer…" or "Let n be a real…" and they would have listened attentively, expecting to understand what was to come. If, instead, one had begun "Let f be a set of reals …" they would not. The eighteenth century had the idea of an *arbitrary integer* or an *arbitrary point* or an *arbitrary line*, but it did not have the idea of an *arbitrary real valued function*, or an *arbitrary set of reals*, or an *ar-*

bitrary set of points. During this period mathematics acquired not only the concept of an arbitrary real-valued function, but also the concepts of arbitrary set, arbitrary formula, arbitrary proof, arbitrary computation, and additionally other concepts that will not concern us here. A reader who is not happy to see a discussion begin "Let x be an arbitrary ...", where the dots are to be filled in with the name of a suite of entities (reals, integers, sets), is to a certain extent not admitting entities from that suite as being fully real in the way they admit entities whose name they will accept in place of the dots. This was put pithily by Quine: "To be is to be the value of a variable". There are arbitrary X's once you have made X's into mathematical objects.

At the start of the third millenium of the common era, mathematics still has not furnished us with the idea of an arbitrary game or arbitrary proof. However, there is a subtle difference between this shortcoming and the eighteenth century's lack of the concept of an arbitrary function. Modern logicians recognise the lack of a satisfactory formalisation of a proof or game as a shortcoming in a way in which the eighteenth century did not recognise their lack of a concept of arbitrary function.

This historical development has pedagogical significance, since most of us acquire our toolkit of mathematical concepts in roughly the same order that the western mathematical tradition did. Ontogeny recapitulates phylogeny after all, and many students find that the propensity to reason in a freewheeling way about arbitrary reals or functions or sets does not come naturally. The ontological toolkit of school mathematics is to a large extent that of the eighteenth century. I remember when studying for my A-level being nonplussed by Richard Watts-Tobin's attempt to interest me in Rolle's theorem and the intermediate value theorem. It was too general. At that stage I was interested only in specific functions with stories to them: $\Sigma_{n \in \mathbb{N}} x^{2^n}$ was one that intruiged me, as did the function $\Sigma_{n \in \mathbb{N}} x^n \cdot n!$ in Hardy's (1949), which I encountered at about that time. I did not have the idea of an arbitrary real-valued function, and so I was not interested in general theorems about them.

Although understanding cannot be commanded, it will often come forward (albeit shyly) once it becomes clear what the task is. The student who does not know how to start answering "How many subsets does a set with n elements have?" may perhaps be helped by pointing out that their difficulty is that they are less happy with the idea of an arbitrary set than with the idea of an arbitrary number. It becomes easier to make the leap of faith once one knows which leap is required.

Some of these new suites of entities were brewed in response to a need

Cambridge University Press
978-0-521-82621-1 — Logic, Induction and Sets
Thomas Forster
Excerpt
More Information

to solve certain problems, and the suites that concern us most will be those that arose in response to problems in logic. Logic exploded into life in the twentieth century with the Hilbert programme and the celebrated incompleteness theorem of Gödel. It is probably a gross simplification to connect the explosive growth in logic in the twentieth century with the Hilbert programme, but that is the way the story is always told. In his famous 1900 address Hilbert posed various challenges whose solution would perforce mean formalising more mathematics. One particularly pertinent example concerns Diophantine equations, which are equations like $x^3 + y^5 = z^2 + w^3$, where the variables range over integers. Is there a general method for finding out when such equations have solutions in the integers? If there is, of course, one exhibits it and the matter is settled. If there is not, then in order to prove this fact one has to be able to say something like: "Let $\mathcal{A}$ be an arbitrary algorithm ..." and then establish that $\mathcal{A}$ did not perform as intended. However, to do *that* one needs a concept of an algorithm as an arbitrary mathematical object, and this was not available in 1900. It turns out that there is no method of the kind that Hilbert wanted for analysing diophantine equations, and in chapter 6 we will see a formal concept of algorithm of the kind needed to demonstrate this.

This extension of mathematical notation to nonmathematical areas has not always been welcomed by mathematicians, some of whom appear to regard logic as mere notation: "If Logic is the source of a mathematician's hygiene, it is not the source of his food" is a famous sniffy aside of Bourbaki. Well, one *bon mot* deserves another: there is a remark of McCarthy's as famous among logicians as Bourbaki's is to mathematicians to the effect that, "It is reasonable to hope that the relationship between computation and mathematical logic will be as fruitful in the next century as that between analysis and physics in the last." With this at the back our minds it has to be expected that when logicians write books about logic for mathematicians they will emphasise the possible connections with topics in theoretical computer science.

The autonomy of syntax

One of the great insights of twentieth-century logic was that, in order to understand how formulæ can bear the meanings they bear, we must first strip them of all those meanings so we can see the symbols as themselves. Stripping symbols of all the meanings we have so lovingly bestowed on them over the centuries in various unsystematic ways[1] seems an extremely perverse thing to do – after all, it was only so that they could bear meaning that we invented the symbols in the first place. But we have to do it so that we can think about formulæ as (perhaps mathematical) objects in their own right, for then can we start to think about how it is possible to ascribe meanings to them in a systematic way that takes account of their internal structure. That makes it possible to prove theorems about what sort of meanings can be born by languages built out of those symbols. These theorems tend to be called *completeness theorems*, and it is only a slight exaggeration to say that logic in the middle of the twentieth century was dominated by the production of them.

It is hard to say what logic is dominated by now because no age understands itself (a very twentieth century insight!), but it does not much matter here because all the material in this book is fairly old and long-established. All the theorems in this will be older than the undergraduate reader; most of them are older than the author.

Finally, a cultural difference. Logicians tend to be much more concerned than other mathematicians about the way in which desirable propositions are proved. For most mathematicians, most of the time, it is enough that a question should be answered. Logicians are much more likely to be concerned to have proofs that use particular methods, or refrain from exploiting particular methods. This is at least in part because the connections between logic and computation make logicians prefer proofs that correspond to constructions in a way which we will see sketched later, but the reasons go back earlier than that. Logicians are more likely than other mathematicians to emphasise that ‘trivial’ does not mean ‘unimportant’. There are important trivialities, many of them in this book. The fact that something is unimportant may nevertheless itself be important. There are some theorems that it is not a kindness to the student to make seem easy. Some hard things should be seen to be hard.

[1] The reader is encourged to dip into Cajori’s *History of Mathematical Notations* to see how unsystematic these ways can be and how many dead ends there have been.

1

Definitions and notations

This chapter is designed to be read in sequence, not merely referred back to. There are even exercises in it to encourage the reader.

Things in **boldface** are usually being **defined**. Things in *italic* are being *emphasised*. Some exercises will be collected at the end of each chapter, but there are many exercises to be found in the body of the text. The intention is that they will all have been inserted at the precise stage in the exposition when they become doable.

I shall use lambda notation for functions. $\lambda x.F(x)$ is the function that, when given x, returns $F(x)$. Thus $\lambda x.x^2$ applied to 2 evaluates to 4. I shall also adhere to the universal practice of writing '$\lambda xy.(\ldots)$' for '$\lambda x.(\lambda y.(\ldots))$'. Granted, most people would write things like '$y = F(x)$' and '$y = x^2$', relying on an implicit convention that, where 'x' and 'y' are the only two variables are used, then y is the output ("ordinate") and x is the input ("abcissa"). This convention, and others like it, have served us quite well, but in the information technology age, when one increasingly wants machines to do a lot of the formula manipulations that used to be done by humans, it turns out that lambda notation and notations related to it are more useful. As it happens, there will not be much use of lambda notation in this text, and I mention it at this stage to make a cultural point as much as anything. By the same token, a word is in order at this point on the kind of horror inspired in logicians by passages like this one, picked almost at random from the literature (Ahlfors, 1953 p. 69):

Suppose that an arc γ with equation $z = z(t), \alpha \leq t \leq \beta$ is contained in a region Ω, and let f be defined and continuous in Ω. Then $w = w(t) = f(z(t))$ defines an arc ...

The linguistic conventions being exploited here can be easily followed

Cambridge University Press
978-0-521-82621-1 — Logic, Induction and Sets
Thomas Forster
Excerpt
More Information

by people brought up in them, but they defy explanation in any terms that would make this syntax machine-readable. Lambda notation is more logical. Writing '$w = \lambda t.f(z(t))$' would have been much better practice.

I write ordered pairs, triples, and so on with angle brackets: $\langle x, y\rangle$. If x is an ordered pair, then $\texttt{fst}(x)$ and $\texttt{snd}(x)$ are the first and second components of x. We will also write '$\vec{x}$' for '$x_1 \ldots x_n$'.

1.1 Structures

A set with a relation (or bundle of relations) associated with it is called a **structure**, and we use angle brackets for this too. $\langle X, R\rangle$ is the set X associated with the relation R, and $\langle X, R_1, R_2 \ldots R_n\rangle$ is X associated with the bundle of relations – $R_1 \ldots R_n$. For example, $\langle \mathbb{N}, \leq\rangle$ is the naturals as an ordered set.

The elements are "in" the structure in the sense that they are members of the underlying set – which the predicates are not. Often we will use the same letter in different fonts to denote the structure and the *domain* of the structure; thus, in "$\mathfrak{M} = \langle M, \ldots\rangle$" M is the domain of $\mathfrak{M}$. Some writers prefer the longer but more evocative locution that M is the **carrier set** of $\mathfrak{M}$, and I will follow that usage here, reserving the word '**domain**' for the set of things that appear as elements of n-tuples in R, where R is an n-place relation. We write '$dom(R)$' for short.

Many people are initially puzzled by notations like $\langle \mathbb{N}, \leq\rangle$. Why specify the ordering when it can be inferred from the underlying set? The ordering of the naturals arises from the naturals in a – natural(!) – way. But it is common and natural to have distinct structures with the same carrier set. The rationals-as-an-ordered-set, the rationals-as-a-field and the rationals-as-an-ordered-field are three distinct structures with the same carrier set. Even if you are happy with the idea of this distinction between carrier-set and structure and will not need for the moment the model-theoretic jargon I am about to introduce in the rest of this paragraph, you may find that it helps to settle your thoughts. The rationals-as-an-ordered-set and the rationals-as-an-ordered-field have the same carrier set, but different signatures (see page 48). We say that the rationals-as-an-ordered-field are an **expansion** of the rationals-as-an-ordered-set, which in turn is a **reduction** of the rationals-as-an-ordered-field. The reals-as-an-ordered-set are an **extension** of the rationals-as-an-ordered-set, and, conversely, the rationals-as-an-ordered-set are a **substructure** of the reals. Thus:

Beef up the signature to get an *expansion*.
Beef up the carrier set to get an *extension*.
Throw away some structure to get a *reduction*.
Throw away some of the carrier set to get a *substructure*.

We will need the notion of an **isomorphism** between two structures. If $\langle X, R\rangle$ and $\langle Y, S\rangle$ are two structures, they are **isomorphic** iff there is a bijection f between X and Y such that, for all $x, y \in X$, $R(x, y)$ iff $S(f(x), f(y))$.

(This dual use of angle brackets for tupling and for notating structures has just provided us with our first example of **overloading**. "Overloading"!? It is computer science-speak for "using one piece of syntax for two distinct purposes" – commonly and gleefully called "abuse of notation" by mathematicians.)

1.2 Intension and extension

Sadly the word 'extension', too, will be overloaded. We will not only have extensions of models – as just now – but extensions of theories (of which more later), and there is even **extensionality**, a property of relations. A binary relation R is extensional if $(\forall x)(\forall y)(x = y \longleftrightarrow (\forall z)(R(x, z) \longleftrightarrow R(y, z))$. Notice that a relation can be extensional without its converse (converses are defined on page 9) being extensional: think "square roots". An extensional relation on a set X corresponds to an injection from X into $\mathcal{P}(X)$, the power set of X. For us the most important example of an extensional relation will be $\in$, set membership. Two sets with the same members are the same set.

Finally, there is the intension extension distinction, an informal device but a standard one we will need at several places. We speak of **functions-in-intension** and **functions-in-extension** and in general of **relations-in-intension** and **relations-in-extension**. There are also 'intensions' and 'extensions' as nouns in their own right.

The standard illustration in the literature concerns the two properties of being *human* and being a *featherless biped* – a creature with two legs and no feathers. There is a perfectly good sense in which these concepts are the same (one can tell that this illustration dates from before the time when the West had encountered Australia with its kangaroos!), but there is another perfectly good sense in which they are different. We name these two senses by saying that 'human' and 'featherless biped' are the same property in extension but different properties in intension.

A more modern and more topical illustration is as follows. A piece of code that needs to call another function can do it in either of two ways. If the function being called is going to be called often, on a restricted range of arguments, and is hard to compute, then the obvious thing to do is compute the set of values in advance and store them in a look-up table in line in the code. On the other hand if the function to be called is not going to be called very often, and the set of arguments on which it is to be called cannot be determined in advance, and if there is an easy algorithm available to compute it, then the obvious strategy is to write code for that algorithm and call it when needed. In the first case the embedded subordinate function is represented as a function-in-extension, and in the second case as a function-in-intension. Functions-in-extension are sometimes called the **graph**s of the corresponding functions-in-intension: the graph of a function f is $\{\langle x, y\rangle : x = f(y)\}$. One cannot begin to answer exercise 1(vi) unless one realises that the question must be, "How many binary relations-*in-extension* are there on a set with n elements?" (There is no answer to "how many binary relations-in-intension ...")

I remember being disquieted – when I was a A-level student – by being shown a proof that if one integrates $\lambda x.\frac{1}{x}$ with respect to x, one gets $\lambda x.log(x)$. The proof procedes by showing that the two functions are the same function-in-extension – or at least that they are both roots of the one functional equation, and that did not satisfy me.

The intension – extension distinction is not a formal technical device, and it does not need to be conceived or used rigorously, but as a piece of mathematical slang it is very useful. One reason why it is a bit slangy is captured by an *aperçu* of Quine's: "No entity without identity". What this *obiter dictum* means is that if you wish to believe in the existence of a suite of entities – numbers, ghosts, functions-in-intension or whatever it may be – then you must have to hand a criterion that tells you when two numbers (ghosts, functions-in-intension) are the same number (ghost, etc.) and when they are different numbers (ghosts, etc). We need *identity criteria* for entities belonging to a suite before those entities can be used rigorously. And sadly, although we have a very robust criterion of identity for functions-in-extension, we do not yet have a good criterion of identity for functions-in-intension. Are the functions-in-intension $\lambda x.x+x$ and $\lambda x.2\cdot x$ two functions or one? Is a function-in-intension an algorithm? Or are algorithms even more intensional than functions-in-intension?

Finally, this slang turns up nowadays in the connection with the idea of evaluation. In recent times there has been increasingly the idea that intensions are the sort of things one *evaluates* and that the things to which they evaluate are extensions.

1.3 Notation for sets and relations

Relations in extension can be thought of as sets of ordered tuples, so we had better ensure we have to hand the elementary set-theoretic gadgetry needed.

Some people write '$\{x|F\}$' for the set of things that are F, but since I will be writing '$|x|$' for the cardinal of x, I shall stick to '$\{x : F(x)\}$'. This notation is commonly extended by moving some of the conditions expressed on the right of the colon to the left: for example, '$\{x \in \mathbb{N} : (\exists y)(x = 2 \cdot y)\}$' instead of '$\{x : x \in \mathbb{N} \wedge (\exists y)(x = 2 \cdot y)\}$'. There is a similar notation for the quantifiers: often one writes '$(\forall n \in \mathbb{N})(\ldots)$' instead of '$(\forall n)(n \in \mathbb{N} \rightarrow \ldots)$'. The reader is presumably familiar with '$\subseteq$' for subset of, but perhaps not with '$x \supseteq y$' (read 'x is a superset of y'): it means the same as $y \subseteq x$'. $\mathcal{P}(x)$ is the **power set** of x: $\{y : y \subseteq x\}$. Set difference: $x \setminus y$ is the set of things that are in x but not in y. The symmetric difference: $x \Delta y$, of x and y, is the set of things in one or the other but not both: $(x \setminus y) \cup (y \setminus x)$. (This is sometimes written '`XOR`', but we will reserve `XOR` for the corresponding propositional connective). Sumset: $\bigcup x := \{y : (\exists z)(y \in z \wedge z \in x)\}$; and intersection $\bigcap x := \{y : (\forall z)(z \in x \rightarrow y \in z)\}$. These will also be written in indexed form at times: $\bigcup_{i \in I} A_i$. The **arity** of a function or a relation is the number of arguments it is supposed to have. It is a significant but generally unremarked fact that one can do most of mathematics without ever having to consider relations of arity greater than 2. These relations are **binary**. The **composition** of two binary relations R and S, which is $\{\langle x, z\rangle : (\exists y)(\langle x, y\rangle \in R \wedge \langle y, z\rangle \in S)\}$, is notated '$R \circ S$'. $R \circ S$ is not in general the same as $S \circ R$: the sibling of your parent is probably not the parent of your sibling. (Mini exercise: how is it legally possible for them to be the same?)

$R \circ R$ is written R^2, and similarly R^n. The **inverse** or **converse** of R, written 'R^{-1}', is $\{\langle x, y\rangle : \langle y, x\rangle \in R\}$. However, do not be misled by this exponential notation into thinking that $R \circ R^{-1}$ is the identity. See exercise 1.

It is sometimes convenient to think of a binary relation as a matrix whose entries are `true` and `false`. This has an advantage, namely, that under this scheme the matrix product of the matrices for R and S is the matrix for $R \circ S$. (Take multiplication to be $\wedge$ and addition to be

$\vee$). However, in principle this is not a good habit, because it forces one to decide on an ordering of the underlying set (rows and columns have to be put down in an order after all) and so is less general than the picture of binary relations-in-extension as sets of ordered pairs. It also assumes thereby that every set can be totally ordered, and this is a nontrivial consequence of the axiom of choice, a contentious assumption of which we will see more later. However, it does give a nice picture of converses: the inverse – converse of R corresponds to the transpose of the matrix corresponding to R, and the matrix corresponding to $R \circ S$ is the product of the two matrices in the obvious way.

A relation R is **transitive** if $\forall x \forall y \forall z\ R(x,y) \wedge R(y,z) Dz \rightarrow R(x,z)$ (or, in brief, $R^2 \subseteq R$). A relation R is **symmetrical** if $\forall x \forall y (R(x,y) \longleftrightarrow R(y,x))$ or $R = R^{-1}$. Beginners often assume that symmetrical relations must be reflexive . They are wrong, as witness "rhymes with", "conflicts with", "can see the whites of the eyes of", "is married to", "is the sibling of" and so on.

An **equivalence relation** is symmetrical, transitive and reflexive. An equivalence relation $\sim$ is a **congruence relation for** an n-ary function f if, whenever $x_i \sim y_i$ for $i \leq n$, then $f(\vec{x}) \sim f(\vec{y})$. (The notation "$\vec{x}$" abbreviates a list of variables, all of the shape 'x' with different subscripts.) A cuddly familiar example is integers mod k: congruence mod k is a congruence relation for addition and multiplication of natural numbers. (It is not a congruence relation for exponentiation: something that often confuses beginners.) We will need this again in sections 3.4 (on boolean algebra) and 5.7 (on ultraproducts) and in chapter 7, on transfinite arithmetic.

I have used the adjective 'reflexive' without defining it. A binary relation on a set X is **reflexive** if it relates every member of X to itself. (A relation is **irreflexive** if it is disjoint from the identity relation: note that irreflexive is not the same as not reflexive!) That is to say, R is reflexive iff $(\forall x \in X)(\langle x, x\rangle \in R)$. Notice that this means that reflexivity is not a property of a relation, but of the structure $\langle X, R\rangle$ of which the relation is a component.

This annoying feature of reflexivity (which irreflexivity does not share) is also exhibited by **surjectivity**, which is a property not of a function but a function-with-a-range. A function is surjective if every element of the range is a value. **Totality** likewise is a property of a function-and-an-intended-domain. A function f on a set X is total if it is defined for every argument in X.

Some mathematical cultures make this explicit, saying that a function