OBJECT DESCRIPTORS: USAGE AND FOUNDATION

FRANK QUINN

CONTENTS

1. Introduction 1
   1.1. Abstract 1
   1.2. Outline 2
2. Everyday set theory 2
   2.1. Definitions 3
   2.2. Unions 3
   2.3. Large domains 5
3. Primitives 6
   3.1. Primitive objects 6
   3.2. Assertion logic 7
   3.3. Primitive hypotheses 8
4. Categories 9
   4.1. Definition 10
   4.2. Isomorphism classes 10
   4.3. Skeleta 11
   4.4. Size of categories 11
   4.5. Functor categories 12
5. Foundations for deductive mathematics 12
   5.1. Terminology 13
   5.2. More about consistency 13
   5.3. Sketchy history 13
References 17

1. INTRODUCTION

1.1. Abstract. In Object descriptors and set theory, [Quinn], we introduced “object descriptors”, a logical environment much more general than set theory. Inside this we found a ‘relaxed’ version of set theory. Relaxed set theory turns out to be functionally the same as the set theory used in mainstream deductive mathematics for the last hundred years, so—in principle—descriptor theory gives a deeper foundation for mathematics. We also see that Zermillo-Fraenkel-Choice (ZFC) set theory fails to imply a common version of the Union axiom, so is not fully satisfactory as a foundation. The final section of this paper gives an abstract discussion of foundations and sketches their historical evolution.

Most of the paper illustrates potential uses:

Date: September 2022.
(1) The full theory will be most useful to abstract category theorists. Object descriptors are essentially an abstraction of the ‘object’ part of a category. Making these precise requires non-binary logic, which perhaps explains why categories do not fit comfortably in standard set theory.

(2) Russell’s paradox shows that the “class of all sets”, the ‘universe’ of standard set theory, is not a set. This puts it outside the perview of standard set theory. It is accessible to relaxed set theory, which may help understand large categories and some ambitious universal constructions.

(3) The set theory used in mainstream mathematics for the last hundred years works well when input data given in terms of sets, so in this context a deeper theory is not needed. Fortunately, relaxed set theory—in this context—is just as easy to use and offers helpful notation and perspective. In particular non-binary logic is not needed.

1.2. Outline. Section 2, (Everyday set theory), concerns use in contexts where the input is already given in terms of sets. This includes most of mainstream mathematics, and is simpler than the general context in that it can be handled effectively with standard binary logic. In this context the theory is almost the same as naïve set theory. We illustrate usage by deriving a standard union operation. We also see that the full version of this operation fails in most versions of ZFC.

Section 3 (Primitives) describes the primitives of descriptor theory (undefined objects, core logic, and assumed hypotheses). Much of this duplicates the corresponding section in [Quinn], and is included to give an independent introduction. Primitive objects are abstractions of the ‘object’ and ‘morphism’ primitives of category theory. The primitive logic is weaker than standard binary logic, and in particular does not include the “law of excluded middle”. The primitive hypotheses are mostly standard, including the axiom of Choice. Encouraged by the results of [Quinn], we include the Quantification hypothesis.

Section 4 shows how categories fit easily into our general context. We present some very basic standard material that is awkward in traditional treatments (isomorphism classes, skeleta, functor categories) but straightforward and natural here.

Finally, Section 5 (Foundation) gives a brief discussion of formal foundations, and suggests that the descriptor approach provides an effective one for mathematics as it is actually practiced.

2. Everyday set theory

Almost all mainstream mathematics uses starting data defined in terms of sets and functions. A group, for instance, is a set together with a binary operation satisfying certain identities. In such contexts set theory becomes routine and almost transparent because standard operations are supposed to stay inside set theory. Most mathematicians, if they think about set theory at all, use naïve set theory as a guide to these operations. Naïve set theory is well-known to be inconsistent, and people know not to say “set of all sets” but—again in contexts where data is given in terms of sets—it has been a reliable and effective guide.

In this section we illustrate the use of relaxed set theory in everyday settings by comparing union properties with naïve and ZFC set theories. Then we describe the smallest logical domain that is not a set.
2.1. Definitions.

(1) A logical function on a collection of elements is a function that has values either ‘yes’ or ‘no’.

(2) A logical domain is a collection of elements with a logical pairing $A \times A \to \text{y/n}$ that detects identity of elements. This is denoted, as usual, by ‘=’. So $a = b$ has value ‘yes’ if $a$ and $b$ are the same, and ‘no’ otherwise.

(3) The powerset of a logical domain is the collection of all logical functions on it. Powersets are denoted $P[A]$.

(4) A domain $A$ supports quantification if there is a logical function $P[A] \to \text{y/n}$ that detects the empty (always ‘no’) function.

(5) A set is a logical domain that supports quantification.

Notes.

(1) A logical function determines a subdomain, namely the subcollection on which the function has value ‘yes’. The traditional focus is on the subcollection, but here we will find it advantageous to emphasize the function. This requires some notation. For example, suppose $A$ is a domain with a partial order $\leq : A \times A \to \text{y/n}$, and $a \in A$. Then $(\# < a)$ is the function that returns ‘yes’ if $\#$ is strictly smaller than $a$. Here $\#$ is a dummy variable for elements of $A$, used to avoid having to introduce a name for the function. If it is useful to make the domain explicit, we use something closer to the traditional set-builder notation: $(\# \in A \mid \# < a)$. We also use square brackets to denote function evaluation: if $f$ is a function then $f[A]$ denotes its value on $a$. This avoids confusion with grouping, and again avoids extra notation. For instance $(\# < a)[b] = (b < a)$ replaces the traditional “$f[b]$, where $f[x] := (x < a)$”.

(2) In relaxed and naïve set theory, “functions” are primitive objects. In ZFC functions are not primitive, so must be defined. ZFC functions are formulas in the first-order logic of set theory, cf. [Jech], p. 7.

(3) In ZFC and naïve set theory, any two elements can be compared. This means they have “universes” of all possible elements, and the universe is a logical domain. In relaxed set theory, elements of different domains generally cannot be compared.

(4) The powerset of a logical domain $A$ is again a logical domain if and only if $A$ supports quantification. There are domains that do not support quantification.

(5) The Quantification Hypothesis asserts that first-order quantification implies infinite-order quantification. This is assumed here. Without this, things are more complicated and “sets” must be defined as domains that support infinite-order quantification, see [Quinn].

2.2. Unions. Suppose we have a collection of sets, indexed by a set. Denote these by $A_\#, \text{ defined for } \# \in B$. For example, if $S$ is a set of objects in a category then $(x, y) \mapsto \text{morph}[x, y]$ is a collection of sets indexed by $S \times S$. We consider the assertion that the disjoint union $\bigcup_{\# \in B} A_\#$ is a set, in naïve, relaxed, and ZFC set theories. In the category example, the union is $\bigcup_{(x, y)} \text{morph}[x, y]$, and we want this to be a set.

The disjoint union is the collection of elements of the form $(b, a)$ where $b \in B$ and $a \in A_b$. Denote this by $C$, then there is a function $C \to B$ defined by $(b, a) \mapsto b$. 
An apparently equivalent formulation is: “suppose \( p: C \to B \) is a function with \( B \) and each \( p^{-1}[b] \) sets. Then \( C \) is a set.” In this form there is a converse: if \( C \) is a set then its image, and each \( p^{-1}[b] \) are sets. This formulation may not be available in ZFC because \( p \) may not be a ZFC function.

**naïve.** In naïve set theory everything is a set. When unconstrained this can be pretty silly. When inputs are required to be sets then this has proved to be reliable, even though “sets” are not defined. In technical terms “inputs are sets” is the restriction to ‘relative comprehension’: set-builder expressions must be of the form \( \{ a \in A \mid p[a] \} \) where \( A \) is a set and \( p \) is a logical function.

**Relaxed.** Each \( p^{-1}[b] \) is assumed to be a set, so there is a pairing \( b : p^{-1}[b] \times p^{-1}[b] \to y/n \) that detects equality of elements. Then we get a pairing \( \subseteq : C \times C \to y/n \) defined by \( (b_1, a_1) \subseteq (b_2, a_2) := (b_1 \equiv b_2) \& (a_1 \equiv a_2) \). This detects equality so shows \( C \) is a logical domain.

Next we show quantification. Each \( p^{-1}[b] \) is assumed to have an empty-set detecting function \( \theta_b : \mathcal{P}[p^{-1}[b]] \to y/n \). A function \( h : \mathcal{P}[C] \to y/n \) is empty if and only if, for every \( b \in B \) the restriction to \( p^{-1}[b] \) is empty, ie. if \( \theta_b[h|p^{-1}[b]] = \text{yes} \) for every \( b \in B \). But \( b \mapsto \theta_b[h|p^{-1}[b]] \) defines a logical function on \( B \), and it is identically ‘yes’ if the negation is identically ‘no’, ie. is empty. Since \( B \) is assumed to be a set, it has a function \( \mathcal{P}[B] \to y/n \) that detects the empty function. Putting these together gives a logical function on \( \mathcal{P}[C] \) that detects the empty function.

**ZFC.** A ZFC set theory consists of a universe of possible elements \( U \), and a “element-of” function \( \in : U \times U \to y/n \). A set is a logical function \( U \to y/n \) of the form \( \{ \# \in x \} \) for some \( x \in U \). There is no ‘external’ way to identify sets so again input objects must be given to be sets in the theory. Less obvious is that input functions need to be specified to be in the theory, because ‘functions’ are defined in terms of sets. The union operation illustrates how this might be a problem.

The Axiom of Union in ZFC is ([Jech], 1.4): suppose \( Y \subset U \) is a set. Then the union \( \cup_{b \in Y}(\# \in b) \) is a set. To relate this to the situation above, note that the assumption that each \( p^{-1}[b] \) is a set means there is an element \( x_b \in U \) so that \( p^{-1}[b] = \{ \# \in x_b \} \). Therefore \( C \) is the union of sets \( \{ \# \in x_b \} \). To apply the Axiom of Union we must see that the subcollection \( \{ x_b \mid b \in B \} \) is a set.

The Axiom of Replacement ([Jech], 1.7) asserts that if \( B \) is a set and \( B \to U \) is a ZFC function then the image is a set. To apply this to the assignment \( b \mapsto x_b \) we would need to see that it is a ZFC function. Recall that this means it is given by a first-order formula. However the way the data is formulated does not include this as a hypothesis and, unfortunately, it is not always automatically true. If the ZFC theory is not a truncation of relaxed set theory, then [Quinn], 6.4, asserts that there is a ZFC set \( B \) and an injective (relaxed) function \( B \to U \) whose image is not a ZFC set. Evidently this injection cannot be a ZFC function. The Axiom of Union does not apply to its image, so the ZFC axioms do not imply the union is a set.

We understand this as follows: it is well-known that there are many implementations of the ZFC axioms, often with mutually exclusive properties. There are surely constructions, eg. with categories, whose outputs would have properties forbidden in some ZFC implementations. This means there must be some mechanism that
blocks the construction in general, and special properties of the particular implementation must be used for it to proceed. Possible failure of the general union operation provides such a blockage. Note that since unions work in relaxed set theory, this does not block constructions per se, but only the conclusion that the outputs are in the particular ZFC implementation.

See section 5 for further comments about this.

2.3. **Large domains.** A logical domain is said to be **large** if it does not support quantification. \( \mathcal{W} \) is the logical domain whose elements are equivalence classes of well-ordered sets, [Quinn], 4.3. This corresponds to the universe, or ordinal numbers, of a traditional axiomatic set theory, so it is not a set. In particular it does not support quantification, so it is not a set in any set theory. Unlike traditional axiomatic set theory we can say quite a bit about it, though this involves non-binary logic.

The almost well-order. The orders on its elements induce a linear order on \( \mathcal{W} \). This cannot be a well-order because these require quantification, but it is an **almost** well-order in the sense that the induced orders on bounded subdomains are well-orders.

**Proposition.** Suppose \( A \subset \mathcal{W} \) is a subdomain.

1. \( A \) is bounded if and only if it is a set (ie. supports quantification);
2. \( A \) is cofinal if and only if it is order-isomorphic to \( \mathcal{W} \); and
3. there is no logical function of subdomains that detects which case occurs.

In standard binary logic (1) and (2) are equivalent, being negations of each other. However, (3) shows that binary logic does not apply. It might be helpful to read (1) as “\( A \) is known to be bounded if and only if it is known to support quantification”. Trying to negate this gives only that nothing is known, which has no logical content.

We mention that subdomains of \( \mathcal{W} \) are logical in the sense that they are detected by logical functions. This means they correspond to elements of the powerset \( \text{P}[\mathcal{W}] \).

Item (3) asserts that there is no logical function on \( \text{P}[\mathcal{W}] \) that distinguishes the two cases, even though they are mutually exclusive and together give all of \( \text{P}[\mathcal{W}] \). This is essentially the same as the failure of quantification on \( \mathcal{W} \).

**Equivalence to \( \mathcal{W} \).**

**Proposition.** For a logical domain \( A \) the following are equivalent:

1. There is a bijection \( A \simeq \mathcal{W} \);
2. there is a function \( A \rightarrow \mathcal{W} \) with point preimages sets and image cofinal in \( \mathcal{W} \); and
3. there are injective (or equivalently, surjective) functions \( A \rightarrow \mathcal{W} \) and \( \mathcal{W} \rightarrow A \).

Again, these properties are not logical, so this should be read as “knowing one case applies implies knowing the others”. (3) follows from the Cantor-Bernstein theorem, see [Quinn] 3.3. Note this theorem requires the injections to have logical image. Images in \( \mathcal{W} \) are known to be logical. The image of \( \mathcal{W} \rightarrow A \) is logical because the composition \( \mathcal{W} \rightarrow A \rightarrow \mathcal{W} \) identifies it as an image in \( \mathcal{W} \). Applications, eg. to category theory, usually use case (2).
Minimality. The “minimality” of $W$ mentioned above is:

**Proposition.** If $A$ is a logical domain, then it does not support quantification if and only if there is an injection $W \rightarrow A$.

As above, both directions in this implication are assertions, and there is no logical function on domains that identifies whether or not they support quantification. Injections $W \rightarrow A$ are obtained by recursion, and this gives no information about whether or not the image is logical. Comparing with the previous proposition we see that if there is an injection whose image is not logical then $A$ cannot inject into $W$. In other words, it is strictly larger than $W$. At present we have no way to construct such a thing, nor applications for it.

3. **Primitives**

There are three types of primitive ingredients: primitive objects, primitive logic, and primitive hypotheses. Primitives are not described in terms of other things, so reliability is an experimental conclusion. This section largely duplicates the corresponding section in [Quinn].

3.1. **Primitive objects.** Deductive systems need objects that are not defined, as starting points. Behavior and usage of primitive objects are specified directly, since they cannot be inferred from a definition. Anything that behaves in the required way then qualifies as such an object.

**Descriptors.** Object descriptors, usually shortened to just “descriptors”, describe things. The things described are called ‘objects’ or ‘outputs’. Usage takes the form $x \in A$, which we read as “$x$ is an output of the descriptor $\in A$”, or “$x$ is an object in $A$”.

The term “descriptor” is intended to suggest that these describe things but, unlike the “element of” primitive in set theory, they have no logical ability to identify outputs. In more detail, if $x$ is already specified then in standard set theory the expression “$x \in A$” may be expected (by excluded middle) to be ‘true’ or ‘false’. Here, for previously specified $x$, $x \in A$ must be proved or asserted, otherwise it is a usage error that invalidates arguments.

Syntax for defining descriptors takes the form “$x \in A$ means (...) ”. For example, the descriptor whose objects are themselves descriptors is defined by:

$A \in \text{OD}$ means “$A$ is an object descriptor.”

A more standard example is the descriptor for groups. “$G$ is a group”, or $G \in (\text{groups})$ means “$G$ is a set together with a binary operation that is associative, and has a unit and inverses”.

**Morphisms.** Morphisms of descriptors are essentially the primitives behind functors of categories. “$f: \in A \rightarrow \in B$ is a morphism” means that $f$ assigns to every object $x \in A$ an object $f[x] \in B$.

“Specifies” can be made more precise, but this form seems to work well enough that we forego complications. Morphisms have some of the structure expected of functors: for instance morphisms $A \xrightarrow{f} B$ and $B \xrightarrow{g} C$ can be composed to get a morphism $A \xrightarrow{g \circ f} C$.
This composition is associative, for the usual reason, but stating this requires use of the assertion form of \(=\), see below.

The definition of ‘morphism’ is implicitly a descriptor. Given descriptors \(\in A, \in B\), the \textit{morphism descriptor} is defined by: \(f \in \text{morph}[A,B]\) means “\(f\) is a morphism \(A \to B\)”.

\textit{Discipline.} There are silly examples that satisfy the requirements to be a descriptor. Fortunately, it is not necessary to try to avoid this because logical functions provide discipline later. For instance, an early text gives “those three chaps over there; Tom, Dick, and Harry” as an example of a set. This does give an object descriptor, but not a set. To see this, note the text containing this example was written more than a hundred years ago. Suppose someone in the present time brought in a box of bones and wanted to know “is this an element of the set \(\{\text{Tom, Dick, Harry}\}\)? And if so, which one?”. We see that the author assumed, no doubt encouraged by use of the natural-language term “property”, that properties identifying elements of sets could be time-dependent or time-specific. But as used here, logical functions are time-independent, and cannot distinguish objects in the physical world. There is no logical pairing that distinguishes the elements of \(\{\text{Tom, Dick, Harry}\}\), so it isn’t actually a set.

3.2. \textit{Assertion logic.} Logic provides methods of reasoning with primitive objects and hypotheses. Traditional set theory uses \textit{binary} (ie yes/no valued) logic. The core logic here is weaker in that it is based on assertions. We describe primitive logical terms, and provide examples to illustrate usage.

There is a language problem: binary logic is deeply embedded in both our natural language and our mathematical thinking. If we say “\(A\) is \(B\)”, for instance, it is usually implicit that this is a logical function that returns ‘yes’ if it is true, and ‘no’ if it is not. In particular we can formally negate to get “\(A\) is not \(B\)” simply by interchanging ‘yes’ and ‘no’. To avoid this we use the word ‘known’ in natural-language formulations, as for example, “\(A\) is known to be \(B\)”. This is sometimes awkward, but it does prevent unwarranted negation. The formal negation of “\(A\) is known to be \(B\)” is “\(A\) is \textit{not} known to be \(B\)”, which has no logical force.

\textit{Assertions.} An \textit{assertion} is a statement that is known to be correct. \(a \in B\), for instance, is implicitly an assertion because it means “\(a\) is (known to be) an output of \(B\)”. Assertions are often indicated by ‘!’. For example, if we specify \(a \in B\), then at a later time we could say \(a! \in B\) because this is known to be correct.

For clarity, statements implemented by logical functions are often indicated by ‘?’. For example, suppose \(a \in A\) and \(A \supset B\). Then “\(a? \in B\)” means “there is a logical function \((\#? \in B)\): \(A \to y/n\), and \(a? \in B\) is the value of that function on \(a\)”. Officially the statement itself does not include an assertion about its value, but it is common to omit “is ‘yes’” when the context doesn’t make sense otherwise. For example if \(h: B \to y/n\) is a logical function on a set then the collection of elements it identifies is traditionally written \(\{a \in B \mid h[a] = yes\}\), rather than \(\{a \in B \mid h[a]\}\).

\textit{Examples.} Examples relevant to set and descriptor theory include:

(1) “\(a! \in A\)” is read as “it is known that \(a\) is an output of the descriptor \(\in A\)”.

(2) “\(a \neq b\)” is read as “\(a\) is known to be the same as \(b\)” (see below for “same as”).
(3) “∃a | ( . . . )” is read as “it is known that there exist an element a such that ( . . . ) holds”. Beware that this is not quantification in the traditional sense, because it is not a logical function of ( . . . ).

(4) “∀a . . . ” is read as “it is known that for all a, ( . . . ) holds”.

(5) Negative statements can be asserted, for example a !≠ b asserts that a is known to be different from b.

(6) !not is the denial modifier, read as “it cannot be that”. For example !not[a != b] translates as “it cannot be that a is known to be the same as b”.

We understand this to be equivalent to “a is known to not be the same as b”, and we use the corresponding notation a !≠ b. Similarly, !not[∃a | ( . . . )] is the same as !∃| ( . . . )].

We also use the common notation := for “defined by”. Note (:=) ⇒ ( != ).

More about ‘!=’. Officially, a != b means that a, b are symbols representing a single output of a descriptor.

Example: This formulation of “same as” makes the usual proof of associativity of composition work. Explicitly, suppose

\[ A_1 \xrightarrow{f_1} A_2 \xrightarrow{f_2} A_3 \xrightarrow{f_3} A_4 \]

are morphisms of descriptors. Then we can see \( f_3 \circ (f_2 \circ f_1) \neq (f_3 \circ f_2) \circ f_1 \).

The traditional set-builder notation can be used to describe images of descriptor morphisms, but does not encode whether or not the image is logical. In almost all cases of interest this is automatic.

Suppose \( f : \in \in B \rightarrow \in \in A \) is a morphism of descriptors. The image of \( f \) is the descriptor

\[ a \in \in \text{im}[f] \text{ means } !\exists b \in \in B | f[b] \neq a \]

together with the projection to \( a \in \in A \). This is almost the standard set-builder notation for images. Writing it that way, and writing the element of \( B \) explicitly as a dummy variable gives

\[ a \in \in \text{im}[f] := \{ !\exists \# \in \in B | f[\#] \neq a \} \].

Note that the use of ‘!∃’ means this expression does not define a logical function of a, and an attempt to use it that way is a usage error.

Next, suppose \( p : A \rightarrow y/n \) is a logical function. We say \( p \) detects the image of \( f \) if

\[ (y \in \in A \& p[y] = \text{yes}) \iff (!\exists \# \in \in B | f[\#] \neq y) \].

If such \( p \) exists we say “\( f \) has logical image.” In these terms, the point of the previous paragraph is that there are descriptor morphisms whose images are not logical.

Finally, a logical function \( p : \in \in A \rightarrow y/n \) defines a descriptor with an injective morphism to \( A \) whose image is detected by \( p \). We usually denote this descriptor by the same name as the logical function: define \( x \in \in p := (x \in \in A \& p[x] = \text{yes}) \).

3.3. Primitive hypotheses. Primitive hypotheses are assertions that we believe are consistent, but cannot justify by reasoning with other primitives. Instead we regard these as experimental proposals. Extremely heavy use suggests all of these, with the possible exception of Quantification, should be completely reliable.
Hypotheses.

**Two:** There is a descriptor $\in y/n$ such that $\text{yes} \in y/n$, $\text{no} \in y/n$ and if $a \in y/n$ then either $a = \text{yes}$ or $a = \text{no}$, but not both.

**Choice:** Suppose $f : \in A \to \in B$ is a morphism of object descriptors and $f$ is known to be onto. Then there is a morphism $g : \in B \to \in A$ so that $g \circ f$ is (known to be) the identity. We refer to such $g$ as **sections**.

**Infinity:** The natural numbers support quantification or, equivalently, is a set.

**Quantification:** If a domain supports quantification then so does its powerset.

Discussion: About Two: The force of this hypothesis is that, unlike general descriptors, we can tell the objects apart. This is an excluded-middle property that needs to be made explicit because we do not require the general principle.

The names ‘yes’, ‘no’ are chosen to make it easy to remember operations (‘and’, ‘or’, etc.). One might prefer ‘1’ and ‘0’ for indexing or connections to Boolean algebra. We avoid ‘true’ and ‘false’ because philosophers have attached so much baggage to them.

About Choice: The term “choice” comes from the idea that if a morphism is onto, then we can “choose” an element in each preimage. Note that in general there is no logical-function way to determine if a morphism is onto, or if the composition is the identity. These must be **assertions**, as above.

The axiom of choice in the traditional setting has strong consequences that have been extensively tested for more than a century. No contradictions have been found, and it is now generally accepted. The above form extends the well-established version to contexts without quantification. This extension has implicitly been used in category theory, again without difficulty.

About Infinity: Using primitive objects and hypotheses other than Infinity, we can construct the natural numbers $\mathbb{N}$ as a logical domain. However we cannot show that it supports quantification. The Infinity hypothesis asserts this. This is essentially the same as the ZFC axiom “there is an infinite set”.

About Quantification The ‘powerset’ of $A$ is the descriptor defined by “$f \in P[A]$ means ‘$f$ is a logical function $f : A \to y/n’”.” Details about quantification are given in §2 and [Quinn], §3.4. Here we use the characterization that $A$ supports quantification if and only if $P[A]$ is a logical domain. The Hypothesis asserts that $P[A]$ also supports quantification, and consequently $P^2[A]$ is a domain. Iterating shows that if $A$ supports quantification then it supports infinite-order quantification.

The exploration in [Quinn] shows that the Quantification Hypothesis has essentially no impact on everyday use, and we have gotten no hints that it might be false. It makes the theory cleaner and the development much easier, so we assume it here.

4. Categories

The definition illustrates the suitability of the primitive objects in this theory. To show the logic and hypotheses in action we demonstrate the existence of skeleta, and work through a formally-complete proof that if categories $A, B$ are small, then the functor category $\text{funct}[A, B]$ is also small.
4.1. **Definition.** A category \( C \) in the terminology here is:

1. an object descriptor \( \text{obj}_C \);
2. for every pair \( A, B \in \text{obj}_C \) a ‘morphism’ object descriptor \( \text{morph}[A, B] \);
3. for every triple \( A, B, C \in \text{obj}_C \) a ‘composition’ morphism \( * : \text{morph}[A, B] \times \text{morph}[B, C] \rightarrow \text{morph}[A, C] \);
4. composition is associative: for every quadruple \( A, B, C, D \in \text{obj}_C \) the two orders of composition give the same morphism

\[
\text{morph}[A, B] \times \text{morph}[B, C] \times \text{morph}[C, D] \rightarrow \text{morph}[A, D].
\]
5. and for every object \( A \in \text{obj}_C \) there is an ‘identity’ morphism \( \text{id}_A \in \text{morph}[A, A] \) so that \( f \in \text{morph}[A, B] \) implies \( \text{id}_A \ast f = f = f \ast \text{id}_B \).

The ‘\( * \)’ notation for composition reverses the usual order: \( f \ast g = g \circ f \). We use it here because the expressions are slightly easier to write and parse.

4.1.1. *The category \( \text{OD} \).* Object descriptors and their morphisms form a category. Explicitly, \( A \in \text{obj}_{\text{OD}} \) means "\( A \) is an object descriptor", and \( F \in \text{morph}_{\text{OD}}[A, B] \) means "\( F \) is a morphism of object descriptors \( A \rightarrow B \)". Composition is composition of descriptor morphisms, and associativity is explained in §3.2.

**Domain and range matching.** The composability criterion for morphisms can be stated as: \( f \in \text{morph}[A, B] \), \( g \in \text{morph}[C, D] \) and \( B \neq C \). Note \( B \neq C \) is an assertion. In the traditional formulation 'B=C' is understood to be a logical function that detects composability. Here there is no such logical function, so this version is unavailable.

**Functors etc.** Functors, and natural transformations of functors are defined as usual (see [Mac Lane], §1). Categories and functors form a category.

4.2. **Isomorphism classes.** A category is **ordinary** if the morphism descriptors are all sets. The category \( \text{OD} \) described above is not ordinary.

In this section we see that isomorphism classes of objects in an ordinary category constitute a logical domain. Morphisms do not naturally descend to isomorphism classes, but we can get unnatural ones using the axiom of Choice. The result is called a “skeleton” of the category. This is a standard construction in category theory, cf. [Mac Lane], but the set-theory foundations used previously do not fully support the way it is used.

**Isomorphism classes.** Suppose \( X, Y \in \text{A} \) are objects in a category. An **isomorphism** is, as usual, a morphism \( i \in \text{morph}[X, Y] \) such that there exists a morphism \( j \in \text{morph}[Y, X] \) so that the compositions \( i \ast j \) and \( j \ast i \) are identities. If \( i \) is an isomorphism then, again as usual, the inverse \( j \) is unique, and is also an isomorphism. Composition of isomorphisms give isomorphisms. The isomorphisms in a category thus define a subcategory, which we denote by \( \text{iso}[\text{A}] \).

If \( \text{A} \) is an ordinary category then “is there an isomorphism \( X \rightarrow Y \)” is a logical function of \( X, Y \). This gives an equivalence relation on objects, and the equivalence classes form a logical domain. We denote this by \( \text{obj}_A / \text{iso} \). There is a quotient morphism (of object descriptors) \( q : \text{obj}_A \rightarrow \text{obj}_A / \text{iso} \).

Since functors preserve identity morphisms, they also preserve isomorphisms. A functor \( F : \text{A} \rightarrow \text{B} \) therefore induces a function \( F : \text{obj}_A / \text{iso} \rightarrow \text{obj}_B / \text{iso} \). In other words \( \text{obj}_\# / \text{iso} \) is a functor, from ordinary categories and functors, to domains and functions.
4.3. **Skeleta.** A category $S$ is **skeletal** if the objects $\text{obj}S$ constitute a logical domain and isomorphic objects are equal. This is equivalent to: the quotient function, from objects to isomorphism classes, is a bijection.

A **skeleton** of a category $A$ is a skeletal category and a functor $s: S \rightarrow A$ that induces a bijection on isomorphism classes of objects, and is a bijection on morphism sets.

**Proposition.** *(Existence of skeleta)*

1. Every ordinary category has a skeleton;
2. the inclusion of a skeleton is an equivalence of categories. More precisely, if $s: S \rightarrow A$ is the inclusion of a skeleton then there is a functor $q: A \rightarrow S$ so that $q \circ s$ is the identity of $S$ and $s \circ q$ is naturally equivalent to the identity of $A$.

**Corollary.** *(Uniqueness)* If $s_n: S_n \rightarrow A$ for $n = 1, 2$ are skeleta, then there is an isomorphism of categories $T: S_1 \rightarrow S_2$ so that $s_2 \circ T$ is naturally equivalent to $s_1$.

For the Corollary, let $q_2: A \rightarrow S_2$ be the natural inverse of (2). Then $T = p_2 \circ s_1$ is the desired isomorphism.

The proof of the Proposition is routine, modulo use of the strong form of ‘Choice’. We give details to illustrate this.

For the first step note that the quotient $q: \text{obj}(A) \rightarrow \text{obj}(A)/\text{iso}$ is known to be a surjective morphism of descriptors. Choice therefore asserts that there is a section, $h: \text{obj}(A)/\text{iso} \rightarrow \text{obj}(A)$. Define a category $S$ with objects the equivalence classes $\text{obj}(A)/\text{iso}$ and morphisms $\text{morph}_S[x, y] := \text{morph}_A[h[x], h[y]]$. It should be clear that $S$ is skeletal. We get a functor $s: S \rightarrow A$ by the section $h$ on objects, and the identity on morphisms. It should be clear that this is a skeleton.

The next step is to extend the quotient function $p: \text{obj}(A) \rightarrow \text{obj}(A)/\text{iso} = \text{obj}(S)$ to a functor. To begin, consider the descriptor with objects $(a, \theta)$, where $a$ is an object in $(A)$ and $\theta$ is an isomorphism $a \simeq h[q[a]]$. The forgetful morphism from this to $\text{obj}(A)$ is onto, so again we can chose a section. Denote this by $a \mapsto (a, \theta_1[a])$. We tidy this up a bit. Define $\theta_2$ by

$$a \xrightarrow{\theta[a]} h[a] \xrightarrow{\theta(h[a])^{-1}} h[a].$$

Then $\theta_2$ has the benefit that it takes an image object $h[b]$ to $(h[b], \text{id})$. Note that we cannot get $\theta_2$ by saying “for objects in the image of $h$, define $\theta_2[h[a]] := (h[a], \text{id})$ and extend randomly to other objects”. For this to be valid we would need a logical function on $\text{obj}(A)$ that detects the image of $h$, and there is generally no reason such a function should exist.

Now extend $q$ to morphisms by: for $f: a \rightarrow b$, $q[f]$ is the composition

$$h[q[a]] \xrightarrow{\theta_2[a]^{-1}} a \xrightarrow{f} b \xrightarrow{\theta_2[b]} h[q[b]].$$

Recall that this is a morphism in $S$ because $h$ is the identity on morphisms.

The proof is completed by observing that $q$ is a functor; $q \circ h = \text{id}[S]$; and $\theta_2$ is a natural equivalence $h \circ q \simeq \text{id}[A]$. □

4.4. **Size of categories.** Traditionally, a category is “small” if the collection of all objects is a set, and “large” otherwise. We expand this terminology a bit to help track things of size $\mathcal{W}$. 
Definition. Suppose $A$ is a category. Then we say $A$ is

1. **small** if $\text{obj}[A]/\text{iso}$ is a set;
2. **large** if $\text{obj}[A]/\text{iso}$ is not a set; and
3. **almost small** if it is small or $\text{obj}[A]/\text{iso}$ is bijective with $\mathbb{W}$.

The “almost small” terminology follows the use of “almost well-ordered” in [Quinn]. The characterizations in §2 provide ways to recognize $\mathbb{W}$.

Examples. The category of sets and functions is almost small but not small: the cardinality bijection $\text{obj}_{\text{Set}}/\text{iso} \to \mathbb{W}$ is a bijection. If $k \in \mathbb{W}$ is a cardinal then the subcategory of sets of cardinality less than $k$ is small.

Skeletal categories. Suppose $K$ is ordinary and skeletal. Let $\text{morph}[K]$ denote the union of $\text{morph}_K[A, B]$ over all pairs of objects $A, B \in \text{obj}[K]$. There is a projection $\text{morph}[K] \to \text{obj}[K] \times \text{obj}[K]$ and, because $K$ is ordinary, point preimages are sets. The results about domains of such functions give:

Proposition. Suppose $K$ is skeletal, then

1. $\text{morph}[K]$ is a logical domain;
2. If $K$ is small then $\text{morph}[K]$ is a set; and
3. If $K$ is almost small but not small, then there is a bijection $\text{morph}[K] \simeq \mathbb{W}$.

4.5. Functor categories.

Proposition. Suppose $A$ is small. Then

1. if $B$ is small then so is the functor category $\text{funct}[A, B]$; and
2. if $B$ is almost small then so is the functor category

In (2) note the possibility that the functor category is small even if $B$ is large.

Functors are the objects of the functor category, and natural equivalences are the isomorphisms. Thus we want to show that natural equivalence classes of functors has the size indicated.

First, compositions with equivalences induce equivalence of functor categories. Existence of skeleta therefore reduces the problem to skeletal categories. Suppose $A, B$ are skeletal, with $A$ small. Then a functor induces a function of total morphism domains $\text{morph}[A] \to \text{morph}[B]$, and $\text{funct}[A, B] \subset \text{fn}[\text{morph}[A], \text{morph}[B]]$. If $B$ is small then both morphism domains are sets, so the domain of functors is also a set. This completes case (1).

If $\text{obj}[B]/\text{iso}$ is bijective to $\mathbb{W}$ then so is $\text{morph}[B]$, and so is the function domain $\text{fn}[\text{morph}[A], \text{morph}[B]]$. The functors are thus a subdomain of something bijective to $\mathbb{W}$ so almost small.

We caution, again, that there is no logical function that distinguishes between subdomains of $\mathbb{W}$ that are sets and those bijective to $\mathbb{W}$. This completes the goals of this section.

5. Foundations for deductive mathematics

Modern core mathematics is infinite-precision in the sense that outcomes—modulo human error—are completely reliable. Reliability is the goal; deduction from a consistent foundation is essentially the only way we have found to routinely achieve it. The foundation in general use for the last hundred years is extremely well-established to be consistent within its domain of applicability. Our contention
in this section is that the development here and in [Quinn] provides a deeper foundation from which the standard foundation can be derived, and which has useful additional features.

5.1. Terminology.

(1) A **primitive foundation** is a collection of hypotheses and methods from which other conclusions can be deduced. Note that the methods of argument (the logic) are part of the foundation. “Primitive” indicates that it is postulated rather than deduced from something else.

(2) A foundation is **consistent** if every chain of deductions that produces a contradiction can be shown to have a logical error.

(3) Informally, the **range** of a foundation is the area in which the methods have traction and are consistent.

We use “range” to indicate whether or not a topic can be reliably studied using the foundation. For instance, arguments in traditional set theory that lead to something like “the class of all sets” have left the range of traditional set theory and are at risk. In contrast, this object is in the range of relaxed set theory.

5.2. More about consistency. The operational view of consistency is “complete reliability”, in the sense that the outcome of an error-free argument will never be contradicted by the outcome of any other such argument. Moreover, the outcome will not introduce errors if used as an ingredient in further arguments. Arguments by contradiction, for example, assume something, deduce a contradiction, and conclude that the thing assumed must be false. Success is sensitively dependent on everything else used in the argument being completely reliable, and that the logic itself does not introduce errors.

The great power of consistency is that it gives an internal way to establish correctness of an outcome: determine that the argument is error-free. This determination is itself usually a human endeavor, so is subject to error. Currently, checking correctness of important proofs is a community undertaking. If a consensus is reached then it is almost never wrong. However, we can never be completely sure, so it is essential that a proof be available for further checking. In other words, a “mathematical result” is a conclusion **together with a proof**. A conclusion asserted without providing a proof does not qualify.

Gödel has shown that foundations generally cannot prove their own consistency. However, it is straightforward to test consistency: make lots of long, delicate deductions, and verify that any contradictions are accounted for by logical errors. In other words, consistency is an experimental result. The current foundation has been thoroughly tested, and consistency within its range is extremely well-established.

5.3. Sketchy history. We give an extremely brief overview of the evolution of foundations, focusing on features relevant to the issues here. For general details I have found the essays in ‘The Stanford Encyclopedia of Philosophy’, for example [Bagaria] and [Ferreirós], to be concise and helpful.

Prehistory. The earliest versions of foundations were algorithms used for arithmetic. Methods used to implement them in specific cases were the associated logic. Consistency made these valuable, and essential for large-scale commerce and government.
Ancient Greece. The first relatively explicit use of foundations was in the geometry and number theory of ancient Greece. The primary importance seems to have been as a metaphor for order in the natural world; consistency coming from logic and innate understanding, rather than from cultural bias or religious or philosophical doctrine. We describe some consequences.

Unlike arithmetic, and much of the mathematics of the last four centuries, it was not valued for its “real-world” applications. This can be seen as a quantifier issue: for metaphors it is sufficient that the theory have some striking and non-obvious outcomes. Practical use requires outcomes for every problem in appropriate settings. For example, clever tricks with ancient geometry can describe the area between a line and a chord of a circle in a few special cases. Calculus gives a description of every such area in terms of transcendental functions, and this description shows that very few cases can be solved with ancient techniques. But many people find the classical tricks more meaningful than routine calculus.

The desire to separate mathematics from outside bias led to a key feature of mathematics: arguments must be validated or rejected using the rules established at the onset. Opinions of priests, sages, or other authority figures should not be accepted as “proof”. Similarly, descriptions of objects, such as ‘point’, ‘line’ and ‘angle’ relied on intuitions from common experience or perception to avoid distortion by philosophical, religious, etc. bias.

The range of this approach is pretty much the development as presented. Plato and others tried, unsuccessfully, to link physical things to “idealizations” that could be manipulated with the same sort of logic. Aristotle and others developed a version of physics, again relying on intuition and physical experience. Unfortunately, and in contrast to geometry, our physical intuitions do not match reality very well, and this physics fails very simple consistency checks. But recall the goal was not application, but meaningful contributions to a world view. More generally, humans seem to not expect consistency, either internal or with reality, of the things we find meaningful. Indeed, pointing out inconsistency tends to get a hostile reaction. People also resent that consistent and effective ideas are rarely immediately meaningful.

Europe in the 1600s. The next significant shift came in the 1600s when Galileo, Kepler, and others showed that mathematics could be immediately powerful in the world, not just a metaphor. We expand on this.

A useful description of deductive mathematics is “development of consequences of consistent foundations”. This cannot apply directly to anything in the physical world, so reliability is used indirectly. People develop mathematical models of physical situations. Then get whatever completely reliable information mathematics can offer about the model. Then conclude either that it says something about the physical world, or a better model is needed.

After this time mathematical development became largely driven by the needs of physical models. Methodology evolved rapidly: fractions finally replaced ratios, and compact algebraic notation replaced earlier discursive formulations. By the time Newton, Leibniz, and their contemporaries were active, the power of application-oriented mathematics was firmly established.

Infinitesimals were part of the foundation of this period. They worked quite well for a time, but eventually became problematic and had to be replaced by limits. We explain how this fits into the analysis here. Consider the truncated polynomial ring $\mathbb{R}[\delta]/\delta^2 = 0$. The presumption was that a function $f$ of a real variable extended to
this ring, and \( f[x + y\delta] = f[x] + g[x]y\delta \) where \( g \) is the derivative of \( f \). This is true if \( f \) is analytic: simply plug the extended variables into the power series. In the terminology above, piecewise analytic functions are the area of applicability of this foundation. But limits of analytic functions need not even be piecewise continuous, and continuous limits can be nowhere differentiable. When limits became important (e.g. Fourier and Laplace series) infinitesimals became untenable. The foundation had to evolve from one in which differentiability was assumed, to one in which criteria for differentiability could be derived.

In that period it was generally felt that real numbers should be understood through intuitions from physical experience. Essentially, these intuitions were included in their foundation. This became problematic as goals became more subtle. People without sufficiently accurate intuitions had unreliable outcomes, and were reduced to appeals to authority (“Gauss did it so it must be ok”) or were shut out of the game. We can see this as pushing the edges of the range of a foundation depending on intuitions.

Late 1800s. The idea that mathematics should have a foundation, and specifically a foundation in set theory, began to have traction in the late 1800s; cf. [Ferreirós]. Dedekind improved problems with real-number intuitions by rigorously describing the reals in terms of natural numbers. Roughly speaking, ‘intuition about the reals’ was replaced by ‘intuitions about natural numbers’. It turned out that at deeper levels our natural-numbers intuitions are still insufficient, and common descriptions lacked logical force. This lead Peano and others to describe natural numbers in terms of primitive versions of set theory. By 1900 foundation-oriented mathematicians were having good success with what we now know as naïve set theory. This development was not universally welcomed, but we skip the drama.

Coming from another direction, Frege, Cantor, and others clarified set theory and showed that it had its own substance, also using variations on naïve set theory.

Early 1900s. In 1902 Russell publicized his paradox showing naïve set theory to be inconsistent. There were many different reactions to this, but some commonalities. First, “comprehension” had to be constrained. Suppose \( A \) is a set, then the subcollection identified by the set-builder construction \( \{ x \in A \mid P[x] = \text{yes} \} \) is also a set. Here \( P \) is a “property”, or more precisely, a certain type of logical function (we return to this below). The new constraint is the requirement that \( A \) should already be known to be a set. Note this formulation does not provide a way to define sets from more primitive data. This forces a change of focus, from what sets “are” to how they are supposed to behave.

Another commonality was recognition that work inside the set context reduces to properties of subsets. A function \( A \to B \), for instance, is characterized by its graph as a subset of \( A \times B \). From this point we follow the main threads in set theory and mainstream mathematics.

Set theorists developed axioms for the behavior of sets. They used the logic developed in the late 1800s, which includes an equality operator that can compare any two elements. This makes the collection of all possible elements a logical domain in the terminology used here. This provides a “universe” of elements, say \( U \), and the sets of the theory are logical functions on \( U \). Next, in an elegant move, \( U \) is also used to parameterize sets: there is a logical pairing \( \in: U \times U \to y/n \), and sets
are logical functions of the form $(\# \in x)$ for some $x \in U$. In the ZF axiom system the set theory is completely specified by $U$ and $\in$.

The final issue concerns the “properties” that can be used in the restricted set-builder construction. This was muddled by use of the natural-language term “property”. For example, natural-language properties are time-specific, or at least time-dependent. But standard binary logic concerns logical functions with no provision for time dependence. Suppose $P$ is a logical function on a domain that contains $A$. Then $\{x \in A \mid P[x] = \text{yes}\}$ is the intersection of $A$ with $P$. The axiom is that if $P$ corresponds to a “property” and $A$ is a set, then the intersection is required to be a set. The issue is that we need enough “properties” to produce enough sets to transact the business of set theory. Experimentation revealed that logical functions obtained by first-order logic from the standard set operations suffice. Accordingly, the Separation axiom in the Zermello-Frankel axiom system requires “properties” to include these first-order logical functions.

We note that this definition of “property” addressed a problem in set theory, not in mathematics. There was not then, nor is there now, any hint that first-order logic is needed in mainstream practice. The Union problem described in §2 indicates that it is actually problematic. There have been first-order side excursions, notably the non-standard analysis of Robinson [Robinson] and the tame topology of Van den Dries [V.d.Dries], but these are not central and have not had much impact.

We turn to the mainstream deductive community. By 1900 quite a few mathematicians had worked with naïve set theory and found it to be effective and reliable. The response to Russell’s paradox was to add the caution “don’t say ‘set of all sets’”. This did not effect actual practice since no-one had found a reason to do this. Similarly, the restriction of “comprehension”, requiring $A$ in $\{x \in A \mid P[x] = \text{yes}\}$ to be a set, had no impact because people did this anyway. The mainstream continued to use naïve set theory with these constraints. This is not as silly as it might sound. Using the perspective developed in §3 it is easy to formalize the theory. The main point is to introduce “functions” as primitive objects, subject to the requirement that a function $f : A \to B$ associate an element of $B$ to each element of $A$. Essentially, the intuitive notion is relabeled as a primitive. This version still uses standard logic. It includes a universal sense of equality, so still assumes a universe of possible elements. It also requires quantification, so the structure of the universe is still outside the range of the theory.

By the 1920s much of the younger generation had embraced the set-theory foundation approach and full rigor, and the subject bloomed. The main set-theory concern of the period was that the axiom of Choice seemed too good to be true.

Late 1900s. By this time the axiom of Choice was accepted and included in standard formulations of both communities. Zermello-Fraenkel-Choice became the “gold standard” in the set-theory community, and naïve-with-choice (with the above constraints) the implicit standard for the mainstream. Apart from that the two communities have diverged.

In 1963 Paul Cohen introduced Forcing as a way to construct new models of set theory from old, frequently with very different properties. Profound and extensive development followed. From the perspective here, this exploited the wiggle room in the Separation axiom, between functions coming from first-order logic, and all logical functions. But naïve set theory implicitly uses all logical functions, so the consequences of forcing are irrelevant to the mainstream. Set theory has developed
in other directions, eg. large cardinal axioms but, again so far, these have not found application in the mainstream. Ultralimits have had significant use in geometry and group theory. These depend on existence of points in the Stone-Čech remainder (a.k.a. nonprincipal ultrafilters) of the natural numbers, but do not require anything more sophisticated. In short, the interests of the two communities have almost no overlap. We are unable to see any missed opportunities, though.

Toward the end of the century category theorists, particularly those investigating higher-order structures, were having trouble working within the constraints of set theory. There were proposals to use category theory as a foundation, but these seemed to implicitly assume a background set theory.

Finally, the twenty-first century is not yet history and we will not comment on it.

References

[Bagaria] Joan Bagaria, *Set Theory*, The Stanford Encyclopedia of Philosophy (Winter 2021 Edition), Edward N. Zalta (ed.), URL = [https://plato.stanford.edu/archives/win2021/entries/set-theory/].

[V.d.Dries] L.A. Van den Dries, and L. Van den Dries *Tame topology and o-minimal structures*, Cambridge university press (Vol. 248) 1998.

[Fiore] José Ferreirós, *The Early Development of Set Theory*, The Stanford Encyclopedia of Philosophy (Spring 2022 Edition), Edward N. Zalta (ed.), URL = <https://plato.stanford.edu/archives/spr2022/entries/settheory-early/>.

[Jech] Thomas J. Jech: *Set theory – 3rd Millennium ed, rev. and expanded*, Springer monographs in mathematics (2002) ISBN 3-540-44085-2

[Mac Lane] Saunders Mac Lane: *Categories for the Working Mathematician* Springer Graduate Texts in Mathematics 5 (1971)

[Quinn] Frank Quinn *Object descriptors and set theory*

[Robinson] Abraham Robinson *Non-standard analysis* Princeton University Press, 2016.