6 Cardinals

6.1 Definition of cardinals ........................................ 27
6.2 The almost-cardinal ‘Shin’ .................................... 28
6.3 The Cantor Beth function .................................... 28
6.4 Hessenberg’s theorem ........................................ 30

7 The universal ZFC theory ...................................... 33

7.1 Main result ..................................................... 33
7.2 Universality .................................................... 34
7.3 Reformulation of the ZFC axioms ............................ 36

8 Bibliography .................................................... 39
1 Introduction

We describe a context more primitive than set theory, with a considerably weaker logic. Inside this context we find relaxed set theory that, among other things, provides a universal model for the ZFC axioms.

Category theorists should find this approach comfortable: the basic ingredients of the primitive context are essentially the object and morphism primitives of category theory. Others should find it almost as easy to use as naive set theory, even with full precision. The axioms are much less complex than ZFC. In particular they do not involve first-order logic, which justifies the general mathematical practice of ignoring first-order fine print in ZFC. Finally—modulo the Quantification hypothesis—it is well-defined. This justifies the general practice of ignoring the many strikingly different implementations of ZFC.

The approach may also offer some perspective for set theorists. Starting in a more general context makes it possible to pin down exactly why the ordinal numbers (class of all sets, universe of elements, etc.) is not a set: it does not support quantification. This means the set theory does not satisfy the axioms of Kelly-Morse or von Neumann-Gödel-Bernys, and is not a Grothendieck Universe. There may be subtheories that do satisfy these axioms, though, and these conclusions depend on the Quantification axiom.

The universal model might provide a way to organize the profound work on ZFC models done in the last century. It may also have consequences for the theory of large cardinals. For example it is known ([Jech] §17) that if a model has a measurable cardinal then it has a nontrivial elementary extension. The universal model has no nontrivial extensions, so evidently it cannot have a measurable cardinal.

1.1 Some ideas

The key to this development is explicit use of logical functions: functions to \{yes, no\}. In the traditional approach these are usually implicit, and rely on the “Law of Excluded Middle”. To illustrate this, suppose we have a collection of objects, \(A\), and a sub-collection \(B \subset A\). Given \(a \in A\) we might ask: “is \(a\) in \(B\)?” Excluded-middle asserts that this question always has an answer, either ‘yes’ or ‘no’. In other words, there is a logical function on \(A\) that detects \(B\). As a result the traditional focus is on subcollections, and functions that detect them are implicit. In the general context here there are subcollections that are not logical, so more precision is required.

Other features missing in the primitive theory but necessary for mainstream mathematics are: a meaning for equality of elements in a collection; and quantification. Accordingly, we identify the subcontext in which these are available. A logical domain is a collection with a pairing \(A \times A \rightarrow \text{yes/no}\) that returns ‘yes’ if and only if the two inputs are the same (in a sense to be made precise later). These are settings for basic binary logic. Quantification also concerns identifications, but up a level. The powerset of a logical domain, denoted \(P[A]\), is the collection of all logical functions on \(A\). We say that a logical domain...
supports quantification if there is a logical function $P[A] \to \text{yes/no}$ that detects the empty (always ‘no’) function. The reason is that standard expression for this uses quantification: $(h = \emptyset) := (\forall x \in A, h[x] = \text{no})$. Conversely, if this one quantification expression is implemented, other quantification expressions over $A$ also work. ‘Supports quantification’ is equivalent to ‘$P[A]$ is a logical domain’.

In principle this pattern could continue: suppose $P[A]$ is a logical domain, and ask “is $P^2[A]$ a logical domain?” The Quantification hypothesis, assumed here, asserts that this is always true. This hypothesis is the least well established of the axioms, and its negation also seems tenable. Fortunately this uncertainty effects things only at the very largest scales, and only set- and category-theorists are likely to be interested. There is further discussion in §2.4.

Returning to the discussion, it turns out that logical domains that support quantification have all the properties desired of sets, though it takes some development (well-orders, cardinals, etc.) to see this. Accordingly we call them ‘sets’, or ‘relaxed sets’ if distinctions are necessary. Note that these sets are free-range in the sense that they are identified individually rather than caged in a model for an elaborate axiom system. As a result they are much easier to use with full precision.

1.2 Outline

Section 2 describes the primitives of descriptor theory (undefined objects, core logic, and assumed hypotheses). Primitive objects are essentially the ‘object’ and ‘morphism’ primitives of category theory. These emerged from a great deal of trial and error, with set theory as the goal: their suitability for categories is a bonus rather than a design objective. The primitive logic is weaker than standard binary logic, and in particular does not include the “law of excluded middle”. Most of it uses assertions rather than binary (yes/no) logic. The primitive hypotheses are mostly standard, including the axiom of Choice. To this we add the Quantification hypothesis mentioned above. The success of traditional set theory provides good experimental evidence that, except for Quantification, these are consistent. We hope to explore consequences of Quantification being false in a subsequent paper. In a nutshell, the development is considerably more complicated and the outcome seems less elegant.

Section 3 describes logical functions, logical domains, and quantification. It also includes a discussion of logical pairings (functions $A \times B \to \text{yes/no}$).

Section 4 gives a brief description of “relaxed” set theory. We describe the differences from usual set theory, and give some indications how it provides a good context for category theory and mainstream mathematics.

The development of well-orders is recalled in Section 5. The first real novelty appears when we consider $\mathbb{W}$, the domain with elements the isomorphism classes of well-orders. The Burali-Forti paradox, formulated in the late 1800s, shows that $\mathbb{W}$ cannot be well-ordered. It comes close: it is a logical domain with a linear order such that every bounded subdomain is well-ordered. The resolution of the paradox is that $\mathbb{W}$ does not support quantification. Therefore, we cannot
logically distinguish between functions on it. We can, for instance, show that a
subdomain that is known to be cofinal, is order-isomorphic to \( W \), but there is
no logical function that distinguishes cofinal subdomains from bounded ones.

Section 6 recalls the development of cardinals and cardinality. The outcome
is essentially the same as in classical set theory, though some of the arguments
are slightly different. There is an “almost-cardinal” upper bound for the card-
nals, namely the order-isomorphism type of \( W \). Cardinals are used to construct
the universal ZFC theory.

Section 7 gives the construction of the universal ZFC set theory. This is a
von Neumann-type cumulative hierarchy, and uses Cantor’s Beth function (§6.3)
as a template. The result is a set theory that satisfies all the Zermillo-Fraenkel-
Choice axioms and, as mentioned above, is the universal such theory.
2 Primitives

There are three types of irreducible ingredients: primitive objects, primitive logic, and primitive hypotheses. We particularly focus on the logic since it is weaker than the logic embedded in our language.

2.1 Primitive objects

Standard practice in mathematics is to define new things in terms of old, and use the definition to infer properties from properties of the old things. The old things are typically defined in terms of yet more basic things. But to get started there must be some objects that are not defined. Properties and usage of primitive objects must be specified directly, since they cannot be inferred from a definition.

The primitive objects here are essentially the “object” and “morphism” primitives of category theory. This was not a deliberate choice: set theory was the goal, but a great deal of experimentation led to category theory anyway.

Descriptors

Object descriptors, usually shortened to just “descriptor”, are indicated by the symbol ∈. Usage takes the form x ∈ A, which we read as “x is an output of the descriptor x ∈ A”, or “x is an object in A”.

The term “descriptor” is supposed 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 ∈ A” may be expected (by excluded middle) to be ‘true’ or ‘false’. Here, for previously specified x, x ∈ A is usually a usage error that invalidates arguments. However, see the assertion forms below in §2.2.

Syntax for defining descriptors takes the form “x ∈ A means ‘...’”. For example, the descriptor whose objects are themselves descriptors is defined by: A ∈ DS means “A is a descriptor.” A more standard example is the descriptor for groups. “G is a group”, or G ∈ (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: x ∈ A → x ∈ B is a morphism” means that every object x ∈ A specifies an object f[x] ∈ B.

“Specifies” can be made somewhat more precise, but it seems to work well enough in practice that we forego the complication. The logical-function overlay also imposes more discipline. Morphisms have some of the structure expected of functors: for instance morphisms A → B and B → C can be composed to get a morphism A → C.
This composition is associative, for the usual reason, but stating this requires use of the assertion form of = (see ‘Primitive logic’, below).

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

### 2.2 Primitive logic

Logic provides methods of reasoning with primitive objects and hypotheses. The core logic here is weaker than that of set theory. We describe primitive logical terms, and provide examples to illustrate usage.

In traditional set theory, logic is presented as a formal language, cf. [Manin]. The role of first-order predicate calculus in some of the axioms seems to make this formality necessary. The lack of an analogous restriction here enables us to be less formal, but deeper investigations may require more precision.

**Assertions**

An assertion is a statement that is known to be correct, usually in the sense that it follows from definitions and primitive hypotheses.

Assertions are generally indicated by an exclamation mark. We think of statements in traditional binary logic as questions, with value either ‘yes’ or ‘no’. Accordingly, these are often indicated by a question mark. The following are the main assertion statements:

1. “\( a \neq b \)” is read as “\( a \) is known to be identical to \( b \)” (see below for “identical”);
2. “\( \exists a \mid \ldots \)” is read as “it is known that there exist an element \( a \) such that (\ldots)”.
3. “\( \exists ! a \mid \ldots \)” is read as “it is known that there does not exist an element \( a \) such that (\ldots)”.
4. “\( \forall a \mid \ldots \)” is read as “it is known that for all \( a \), (\ldots) holds”.
5. “\( a \in A \)” is read as “it is known that \( a \) is an output of the descriptor \( \in A \)”.

The main functional difference from binary logic is that these cannot be usefully negated. For instance, the negation of “\( a \) is known to be identical to \( b \)” is “\( a \) is not known to be identical to \( b \)”. This has no logical force. Proofs by contradiction still work, in the sense that if is known that if “\( x \) has \( Q \)” is false, then \( x \) is known not to have \( Q \).

We also use the common notation := for “defined by”. Note \( (:=) \Rightarrow (\neq) \).
More about ‘! =’

Officially, “identical” in the definition of \( a \neq b \) means that \( a, b \) are symbols representing a single output of the descriptor.

Example: Suppose \( a \in A \) and \( b \in B \) are descriptors, and \( b_0 \in B \). Then there are projections and inclusions

\[
p : A \times B \to A, \quad \text{by} \quad (a, b) \mapsto a
\]

\[
j : A \to A \times B, \quad \text{by} \quad a \mapsto (a, b_0)
\]

Then \( p[j[a]] \neq p[(a, b_0)] \neq a \).

Sometimes there are more-explicit formulations for identity. For example, suppose \( f, g : A \to B \) are morphisms of descriptors. Then \( (f \neq g) \) is equivalent to \( (\forall a \in A, f[a] \neq g[a]) \). This formulation of “identical” makes the usual proof of associativity of composition work. Explicitly, suppose

\[
\begin{align*}
A_1 & \xrightarrow{f_1} A_2 \xrightarrow{f_2} A_3 \xrightarrow{f_3} A_4
\end{align*}
\]

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

Example: Russell’s paradox

This gives an assertion that cannot be a logical function.

Suppose \( a \in A \) is a descriptor, and consider the statement \( (a \in A) \in A \). This makes sense as an assertion, ie. “consider A such that A is known to be an object of itself”. Suppose it is implemented by a logical function defined on descriptors. In that case we could define a descriptor \( \text{NotIn} \) by: \( A \in \text{NotIn} \) means “the logical function \( # \in \# \) has value ‘no’ on A”. This leads to a contradiction because the value of \( \text{NotIn} \in \text{NotIn} \) cannot be either ‘yes’ or ‘no’. Therefore there is no such logical function.

Here is another way to look at this. Descriptors cannot recognize their outputs, so any interpretation of \( (a \in A) \in A \) other than an assertion (eg. as a defining characteristic) is a useage error. In contrast, the logic of traditional set theory implies \( A \in A \) must define a logical function on sets. The traditional conclusion from the contradiction is that the descriptor \( \text{NotIn} \) cannot be a set. Alternatively, ZFC theories are well-founded so \( A \in A \) always has value ‘no’. Negating gives that if \( \text{NotIn} \) is a set then \( \text{NotIn} \in \text{NotIn} \) must be ‘yes’. Since this is not possible, \( \text{NotIn} \) cannot be a set.

Example: von Neumann’s axiom

von Neumann’s “axiom of size” for this context would assert that a domain \( A \) is a set if and only if the class \( \mathcal{W} \) (the ordinals) does not inject into it. The binary-logic interpretation is that this would give a criterion for something to be a set. This would considerably simplify set theory because it implies many of the ZFC axioms. The axiom did not catch on because the experts of the time (eventually including von Neumann) worried that it is too powerful. These experts were
right if the Quantification hypothesis is right: we see that the axiom is true in an assertion sense, but does not give a yes/no criterion for domains to be sets. This illustrates some of the subtleties of assertion statements. (If Quantification is wrong then von Neumann’s axiom does work; we explain this in a subsequent paper.)

**Proposition.** Suppose $A$ is a logical domain.

1. There is known to be an injection $\mathbb{W} \rightarrow A$ if and only if $A$ is known not to be a set; and conversely

2. It is known there is no such injection if and only if $A$ is known to be a set.

However there is no logical function that detects which of (1) or (2) hold.

See §5.4. So knowledge, about whether or not $\mathbb{W}$ injects, translates faithfully to knowledge about whether or not something is a set. Unfortunately this knowledge is not encoded in a logical function, so cannot be used to detect sets. For the same reason, item (2) cannot be obtained by negating (1).

**Example: Subdescriptors and the set-builder notation**

The traditional set-builder notation is useful in describing general subdescriptors, but not logical ones.

First, define a subdescriptor to be an injective morphism of descriptors. A subdescriptor is logical if there is a logical function that detects its image. We make this last explicit. Suppose $b: \in B \rightarrow \in A$ is an injective morphism, and $p: A \rightarrow \text{y/n}$. Then $p$ detects $B$ if $(\in y \in A \& p[y] = \text{yes}) \iff (\exists \in x \in B \mid b[x] \neq y)$.

Now suppose $\in X$ is an object descriptor, and some of the $\in y \in A$ have a property denoted ‘$P$’ (“property” is a placeholder for some sort of assertion; see example below). The traditional set-builder notation is

$$\{ y \in A \mid y \text{ has } P \}.$$

We interpret this as a subdescriptor. The official syntax is:

$$z \in \{ y \in A \mid y \text{ has } P \} \text{ means } “z \in A \& (z \text{ has } P)”.$$

In this form we see that $y$ is a dummy variable, and it may be a bit clearer if we write it explicitly that way: $\{ \# \in A \mid \# \text{ has } P \}$. Finally, “has $P$” should be understood as an assertion, and might be better written as “# is known to have $P$”.

Example: if $f: \in B \rightarrow \in A$ is a morphism of descriptors, then the image of $f$ is

$$\text{im}[f] := \{ \# \in A \mid \exists \in x \in B \mid \# \neq f[x] \}.$$

We caution that, in contrast to traditional set theory, the set-builder notation does not define a logical function. Traditionally one could begin with $z \in A$, and expect $z \in \{ \# \in A \mid \# \text{ has } P \}$ to return ‘yes’ or ‘no’ (or, ‘true’ or ‘false’)
depending on whether or not \( z \) has \( P \). Here, however, “has \( P \)” may not be a logical function and, if the expression can be considered to have a value, it is ‘wrong’. This shows up in the logic as follows: descriptors cannot identify their outputs, so choosing \( z \in A \) first makes writing \( z \in \{ \# \in A \mid \# \text{ has } P \} \) a syntax error.

We usually use the same name to denote a logical function and the corresponding logical subdescriptor. Thus, if \( p: A \rightarrow y/n \) is a logical function, we define \( x \in p := (x \in A \& p[x] = \text{yes}) \). The right-hand side of this is essentially the traditional notation when “\( x \) has \( P \)” means “\( p[x] = \text{yes} \)”, but it retains the information that “has \( P \)” is a logical function.

2.3 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 hypotheses. More than a century of extremely heavy use indicates that most of them are completely reliable. The Quantification hypothesis is less well-tested; see below.

| Hypotheses |
|-------------|
| **Two** : There is a descriptor \( \in \in y/n \) such that \( \text{yes} \in y/n \), \( \text{no} \in y/n \) and if \( a \in y/n \) then either \( a \neq \text{yes} \) or \( a \neq \text{no} \). |
| **Choice** : Suppose \( f: \in A \rightarrow \in B \) is a morphism of object descriptors and \( f \) is known to be onto. Then there is a morphism \( g: \in B \rightarrow \in A \) so that \( g \circ f \) is (known to be) the identity. We refer to such morphisms as sections. |
| **Infinity** : The powerset of the natural numbers is a logical domain. (Equivalently, the real numbers is a logical domain, or, the natural numbers support quantification.) |
| **Quantification** : If \( A \) and \( \mathcal{P}[A] \) are logical domains, then so is \( \mathcal{P}^2[A] \). Equivalently, if \( A \) supports quantification then \( \mathcal{P}[A] \) does too. |

2.4 Discussion

About Two

The force of this hypothesis is that, unlike general descriptors, we can tell the objects apart. This is an excluded-middle conclusion 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 other uses conflict with this.
About Choice

The term "choice" comes from the idea that if a morphism is onto, then we can "choose" an element in each preimage to get a morphism \(g\). 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, or equivalently, that the definition of the real numbers using \(P[\mathbb{N}]\) is a logical domain. The Infinity hypothesis is that this is the case. This is essentially the same as the ZFC axiom "there is an infinite set".

About Quantification

Without the axiom one sees that there is a threshold \(n\) so that if \(P^n[A]\) is a domain then \(P^k[A]\) is a domain for all \(k\). Huristically, if \(P^n\) does not make \(A\) explode, then it never explodes. The quantification hypothesis is that the threshold is 1: if it is going to explode then it does so immediately. Counterexamples could only occur at very large scale, larger than any set, so are irrelevant for most practical purposes.
3 Logical domains, and quantification

We provide formal definitions and basic properties of logical domains and quantification. Only first-order quantification is used until the final subsection.

3.1 Logical domains

A descriptor \( \in A \) is a logical domain, or simply “domain”, if there is a logical function of two variables that detects equality. Explicitly, there is \( ?= : A \times A \) such that \((a ?= b) \Rightarrow (a \neq b)\). We reserve the notation ‘?’ for this use, i.e. for equality-detecting pairings on domains.

Logical domain are first-order approximations to traditional sets, so we use traditional terms. Objects in a domain are referred to as elements, and we use \( a \in D \) instead of \( a \in A \).

Domains of equivalence classes

A pre-domain consists of \((\in D, \text{eqv})\) where

1. \( \in D \) is an object descriptor,
2. \( \text{eqv}[x, y] \) is a logical pairing \( D \times D \rightarrow \{\text{y/n}\}; \)
3. \( \text{eqv} \) satisfies the standard requirements for an equivalence relation:
   - reflexive: \( \text{eqv}[x, x] = \text{yes} \);
   - symmetric: \( \text{eqv}[x, y] \Rightarrow \text{eqv}[y, x] \);
   - and transitive: \( \text{eqv}[x, y] \& \text{eqv}[y, z] \Rightarrow \text{eqv}[x, z] \).

Given this data we define a quotient descriptor by: \( h \in D/\text{eqv} \) means “\( h \) is a logical function \( D \rightarrow \{\text{y/n}\} \) and \( h \neq \text{eqv}[x, #] \) for some \( x \in D \)”. Define \( ?= : D/\text{eqv} \times D/\text{eqv} \rightarrow \{\text{y/n}\} \) by: suppose \( h \neq \text{eqv}[x, #] \) and \( g \neq \text{eqv}[y, #] \), then \( (h ?= g) := \text{eqv}[x, y] \). The hypothesis that ‘\( \text{eqv} \)’ is a (logical) equivalence relation implies that ‘?’ is a logical pairing, and does not depend on which representatives \( x, y \) are used. Similarly \( ?= \Rightarrow != \). Thus \((D/\text{eqv}, ?=)\) is a logical domain.

Finally, note that if \( ?= \Rightarrow != \), as in the definition of domain, then \( ?= \) is automatically an equivalence relation. Therefore domains are also pre-domains; exactly those for which the quotient function \( D \rightarrow D/\text{eqv} \) is a bijection.

3.2 Cantor-Bernstein theorem

This is the first of a number of classical results that we review to clarify quantification and binary-logic requirements. It is put here to emphasize it does not require quantification.

**Theorem.** (Cantor-Bernstein) Suppose \( A, B \) are logical domains and \( A \rightarrow B \rightarrow A \) are injections with logical image. Then there is a bijection \( A \simeq B \).
Proof: Composing the injections reduces the hypotheses to the following. Suppose $J: A \to A$ is an injection, $j: A \to \mathbb{y/n}$ detects the image of $J$, and $k$ is a logical function with $j \subset k \subset A$ ($k$ detects $B$). Composing with iterates of $J$ gives a sequence

$$
\cdots k \circ J^{n+1} \subset j \circ J^n \subset k \circ J^n \cdots \subset j \circ J^0 \subset k \circ J^0 \subset A.
$$

Define a new function $\hat{J}: A \to A$ by:

$$
\hat{J}[a] := \begin{cases} 
\text{if } \exists n \in \mathbb{N} \mid a \in k \circ J^n - j \circ J^n, & \text{then } a \\
\text{if not} & \text{then } J[a]
\end{cases}
$$

Then it is straightforward to see that $\hat{J}$ is a bijection $A \to k \simeq B$, as required.

The first case in the definition of $\hat{J}$ uses quantification over the natural numbers, but the Axiom of Infinity asserts that this is valid. In the second case “if not” makes sense because we can apply not[*] to a logical function.

### 3.3 Quantification

The empty function on $A$ is $\emptyset[a] := \text{no}$, for all $a \in A$.

**Definition.** A logical domain $(A, \equiv)$ supports quantification, or is a set if there is a logical function $P[A] \to \mathbb{y/n}$ that detects the empty function.

The traditional quantification notation for the empty-set detecting function is $(\forall a \in A, h[a] \equiv \text{no})$. The logic here does not imply that such expressions define logical functions. However if there is a logical function that implements the intent of this particular expression, then all expressions using quantification over $A$ will define logical functions. This is illustrated by:

**Lemma.** A domain supports quantification if and only if $P[A]$ is a domain.

Suppose $\equiv$ is a logical pairing on $P[A]$ such that $\equiv \Rightarrow \text{!}$. Then $\# \equiv \emptyset$ is a logical function that detects the empty function. Conversely, suppose $\phi: P[A] \to \mathbb{y/n}$ detects $\emptyset$, ie $(\phi[h] = \text{yes}) \leftrightarrow (h \equiv \emptyset)$. Define ‘?=' by $(h ?= g) := \phi[\# \mapsto (h[\#] = g[\#])]$. Then

$$(h ?= g) \Rightarrow (\forall \# \in A, h[\#] = g[\#]) \Rightarrow (h \equiv g).$$

**Operations on $P[A]$**

The logical operations on $\mathbb{y/n}$ induce operations on logical functions $A \to \mathbb{y/n}$, and these correspond to traditional operations on subsets. Explicitly:

1. intersection: $h_1 \cap h_2 := h_1 \& h_2$
2. union: \( h_1 \cup h_2 := h_1 \text{ or } h_2 \)

3. complement: \( D - h_1 := \text{not}[h_1] \)

We caution that these operations may not be defined for subdomains that are not logical; see the next section.

### 3.4 Subdomains

Logical functions correspond to subsets in traditional set theory: the function \( h: A \rightarrow \text{y/n} \) determines a subset by \( \{ x \in A \mid h[x] = \text{yes} \} \). In the traditional theory “excluded middle” is assumed to hold, which implies that the function can be recovered from the subset. Here, failure of excluded middle means there may be subdomains that are not logical; see ‘Example: subdescriptors’ in 2.2 for explanation and notation. However we have:

**Lemma.** Suppose \( B \subset A \) is a subdomain. If \( B \) is a set then it is logical. Conversely, if \( A \) is a set and \( B \) is logical, then \( B \) is a set.

We will eventually see that subdomains of a set domains must be logical, so “logical” can be dropped from the converse statement.

**Proof:** The traditional expression for a logical function that detects \( B \) is \( b[x] := (x \in A \& \exists y \in B \mid y = x) \). If \( B \) is a set then it supports quantification and ‘\( \exists \)’ defines a logical function. For the converse suppose \( b[#] \) detects \( B \). A logical function \( h \) on \( B \) extends to one on \( A \) by \( \hat{h}[x] := (b[x] \& h[x]) \), and the extension is empty if and only if \( h \) is empty. But since \( A \) is a set we can logically detect the empty function on it. Applying this to \( \hat{h} \) detects the empty function on \( B \).

**Functions with set support**

There is a useful blend of the above two lemmas. Suppose \( A \) is a logical domain, and define \( sP[A] \) to be the logical functions whose support is (known to be) a set.

**Lemma.** \( sP[A] \) is a logical domain, for any \( A \).

**Proof:** Suppose \( B, C \subset A \) are sets. Then \( (B \subseteq C) := (\forall b \in B \mid C[b] = \text{yes}) \) is a logical pairing (because \( B \) is a set) that detects inclusion. Define \( B ?= C \) by \( (B \subseteq C) \& (C \subseteq B) \), then \( ?= \implies != \), as required.

### 3.5 Logical pairings

A **logical pairing** is a logical function of two variables, or equivalently, a function on a product \( \lambda: A \times B \rightarrow \text{y/n} \).

This section briefly mentions basic structure, and describes detection of special pairings. In traditional set theory, topology, etc. logical pairings are usually called ‘relations’, but this term is not used here.
**Examples**

Equivalence relations, isomorphisms, orderings, and set theories are all defined in terms of pairings. The ‘?’ and ‘eqv’ functions required in the definition of domain and pre-domain are pairings.

Functions $f: A \to B$ are primitive objects here. On domains, functions determine pairings by $f[a, b] := (f[a] ?= b)$, and this puts them in a larger context where they can be logically manipulated.

If $A$ is a set then $P[A]$ is a domain and the evaluation pairing

$$\text{Ev}: P[A] \times A \to y/n,$$

is defined by $\text{Ev}[h, x] := h[x]$. Note the convention that the $P[A]$ variable comes first in the evaluation pairing. This order seems to be the most convenient, but it is not consistent with some traditional notations. For instance if we think of $h$ as a subset then $h[x]$ is traditionally denoted by ‘$x \in h$’, with the $A$ variable first.

**Standard structure**

Suppose $\lambda: A \times B \to y/n$ is a pairing of sets. Then:

1. the **domain** of $\lambda$ is the logical function on $A$ defined by: $\text{dom}[\lambda, a] := (? \exists b \in B \mid \lambda[a, b])$. The image is defined similarly.

2. The **opposite** is defined by formally changing the order of the variables: $\lambda^{\text{op}}[b, a] := \lambda[a, b]$. Note that the domain of $\lambda^{\text{op}}$ is the image of $\lambda$, and vice versa.

3. $\lambda$ is **single-valued** if $\forall a, x, y, (\lambda[a, x] & \lambda[a, y]) \implies (x = y)$

4. $\lambda$ is **injective** if the opposite is single-valued;

Note that ‘single-valued’ does not require that the domain of $\lambda$ be all of $A$. If $\text{dom}[\lambda] = A$ then we say $\lambda$ has **full domain**. Pairings corresponding to functions are automatically full-domain. The opposite notion, full-image, is traditionally called “onto”.

**Adjoints**

Suppose $\lambda$ is a pairing on $A \times B$ and $B$ is a set. Then $P[B]$ is a logical domain, and we can define the **adjoint** function $\lambda^{\text{adj}}: A \to P[B]$ by:

$$a \mapsto \lambda[a, #].$$

For example, the evaluation pairing $\text{Ev}: P[B] \times B$ has adjoint the identity function $P[B] \to P[B]$. 

15
Composition

Suppose $\alpha: A \times B \rightarrow y/n$ and $\beta: B \times C \rightarrow y/n$ are pairings of sets. The composition is defined by

$$(\alpha \ast \beta)[a,c] := (\exists b \in B \mid \alpha[a,b] \& \beta[b,c]).$$

If $\alpha, \beta$ are both single-valued then pairing-composition corresponds to composition of the associated functions, but the notations have reversed order. If we denote the function associated to a single-valued $\lambda$ by $\hat{\lambda}: A \rightarrow B$ then

$$\hat{\alpha} \ast \hat{\beta} = \hat{\beta} \circ \hat{\alpha}.$$

3.6 Detection of special pairings

We will sometimes want questions such as “is there a nice pairing on $A \times B$?” to define a logical function of $A$ and $B$. This proceeds in two steps. First, show we can detect nice pairings with a logical function $P[A \times B] \rightarrow y/n$. This usually requires quantification on $A \times B$. The second step is “is the detecting function nonempty?”, which requires quantification on $P[A \times B]$. We an example.

Suppose $A, B$ are sets.

1. There is a logical function $\text{?inj}: P[A \times B] \rightarrow y/n$ such that $\lambda$ is an injection if and only if $\text{?inj}[\lambda] = \text{yes}$;

2. “is there an injective pairing on $A \times B$?” is therefore a logical function of $A, B$.

The required function is:

$$\text{?inj}[\lambda] := (\exists a_1, a_2 \in A, b \in B, (\lambda[a_1, b] \& \lambda[a_2, b] \implies a_1 = a_2))$$
4 Disjointness and categories

Relaxed sets have all the properties usually expected of sets, though it takes the technical developments of the next two sections to show this. Here we explain one of the differences: generic disjointness. The category material describes modifications needed in the standard definitions c.f. [Mac Lane], and discusses skeleta and small categories.

4.1 Generic disjointness

Two logical domains can intersect only if they are both subdomains of another domain. In traditional set theory, sets are all subdomains of the class of all possible elements. In principle, therefore, two randomly-chosen sets could intersect. If disjointness is needed then, again in principle, repositioning may be necessary.

In actual practice most mathematicians assume sets are disjoint unless there is a reason they might intersect. Relaxed set theory justifies this assumption. A logical domain is an object descriptor. Generally there is no logical function that can determine whether the output of one descriptor is somehow the same as, or equivalent to, the output of another. Domains are therefore generally disjoint, not because equality of elements is defined and the domains are known not to share elements, but because equality of elements is not defined.

4.2 Categories

Categories, functors, etc. have very general definitions here. The ones of most use are essentially the same as in [Mac Lane], with relaxed sets substituted for the Foundations section of that work, §1.6.

Definition, category

A category \( \mathcal{A} \) consists of:

1. an object descriptor, denoted by \( \text{obj}[\mathcal{A}] \) or \( \text{obj}_{\mathcal{A}} \);
2. for every pair \( A, B \in \mathcal{A} \), an object descriptor \( \text{morph}_{\mathcal{A}}[A, B] \);
3. for every triple \( A, B, C \in \mathcal{A} \), a “composition” morphism

\[
*: \text{morph}[A, B] \times \text{morph}[B, C] \to \text{morph}[A, C]
\]

such that

4. composition is associative, and there are identity elements \( \text{id}[A] \) in \( \text{morph}[A, A] \), for every \( A \).

Note that we are using “\(*\)” for composition because the order is opposite the traditional one. If \( A \xrightarrow{f} B \) and \( B \xrightarrow{g} C \) then we can write the composition as either \( f \ast g \) or \( g \circ f \).
We say that a category is **ordinary** if the morphism-descriptors are (relaxed) sets. Traditional treatments are restricted to ordinary categories.

Functors and natural transformations in this context are straightforward translations of the traditional notions, [Mac Lane] §1.3, 1.4.

**Example: object descriptors**

The properties ascribed, in §2.1, to the primitive notions “object descriptor” and “morphism” are essentially that they form a category. To make this explicit, define $X \in OD$ to mean “$X$ is an object descriptor”. $F \in \text{morp}_OD[X,Y]$ means “$F$ is a morphism of object descriptors.” Composition is composition of morphisms.

Some care is needed in using this category due to the failure of quantification. For example, we might be able to show two morphisms are the same, or are different, but generally there are no logical functions that distinguish between morphisms.

**Example: sets**

The standard category of sets, denoted $\text{Set}$, has objects relaxed sets, and morphisms the functions $A \to B$. We enlarge this a bit. The category of sets and pairings has objects sets, and $\text{morph}[A,B] := P[A \times B]$. Composition of morphisms is defined to be the composition of pairings $(\alpha, \beta) \mapsto \alpha * \beta$ defined in §3.5.

This definition requires objects to be sets because composition of pairings requires quantification. $(\text{Set}, \text{pairings})$ has $(\text{Set}, \text{functions})$ as a subcategory: functions correspond to single-valued pairings with full domain. It follows that both of these categories are ordinary (morphism descriptors are sets).

**Example: categories and functors**

We denote by $\text{Cat}$ the category with objects categories, and morphisms the functors between categories. In the next section we will use skeleta to find ordinary subcategories of this.

**4.3 Isomorphism classes and skeleta**

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 seem not to fully justify the way it is used.
Isomorphism classes

Suppose $\mathcal{A}$ is a category, and $X, Y \in \mathcal{A}$ are objects. An isomorphism is, as usual, a morphism $i \in \text{morph}[X,Y]$ such that there exists a $j \in \text{morph}[Y,X]$ so that the compositions $i \circ j$ and $j \circ i$ 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.

If the morphism-descriptors of $\mathcal{A}$ are sets, then $\text{iso}[X,Y]$, defined by “is there an isomorphism $X \to Y$?” is a logical function of $X,Y$ (see §3.6). This is an equivalence relation on objects. According to ‘Domains of equivalence classes’ in §3.1, the equivalence classes form a logical domain. We denote this by $\text{obj}\mathcal{A}/\text{iso}$. There is a quotient morphism (of object descriptors) $q: \text{obj}\mathcal{A} \to \text{obj}\mathcal{A}/\text{iso}$.

Since functors preserve identity morphisms, they also preserve isomorphisms. A functor $F: \mathcal{A} \to \mathcal{B}$ therefore induces a function $F: \text{obj}\mathcal{A}/\text{iso} \to \text{obj}\mathcal{B}/\text{iso}$. In other words $\#/\text{iso}$ is a functor, from ordinary categories and functors, to domains and functions.

Skeleta

A category $\mathcal{S}$ is skeletal if the objects $\text{obj}\mathcal{S}$ constitute a logical domain (ie. we can distinguish between them), and isomorphic objects are equal. The isomorphism condition is equivalent to: the quotient function, from objects to isomorphism classes, is a bijection.

A skeleton of a category $\mathcal{A}$ is a skeletal category and a functor $s: \mathcal{S} \to \mathcal{A}$ that induces a bijection on isomorphism classes, 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.

The proof is routine, but we go through it because this is where the strong form of ‘Choice’ is used.

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

The next step is to extend the quotient function $q: \text{obj}(\mathcal{A}) \to \text{obj}(\mathcal{A})/\text{iso} = \text{obj}(\mathcal{S})$ to a functor. To begin, consider the descriptor with objects $(a, \theta)$, where $a$ is an object in $(\mathcal{A})$ and $\theta$ is an isomorphism $a \simeq h[q[a]]$. The forgetful morphism from this to $\text{obj}(\mathcal{A})$ 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

$$a \xrightarrow{\theta[a]} h[a] \xrightarrow{\theta_1[a]} h[a].$$
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 $f : a \to b$, $q[f]$ 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 $\mathcal{S}$ because $h$ is the identity on morphisms.

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

### 4.4 Small categories

This illustrates the use of skeleta. It is extended to “almost-small” categories in another paper. An ordinary category $\mathcal{A}$ is said to be **small** if the domain of isomorphism classes $\text{obj}\mathcal{A}/\text{iso}$ is a set.

Traditionally “small” requires the collection of all objects, not just isomorphism classes, to be a set. Relaxing the condition to isomorphism classes fits better with the way categories are typically used, particularly as regards expectations about disjointness; see the discussion above.

**Proposition.** If $\mathcal{A}, \mathcal{B}$ are small categories then the functor category $\text{Funct}[\mathcal{A}, \mathcal{B}]$ is small.

Since equivalences of categories induce isomorphisms of functor categories, the proposition reduces to the case where both $\mathcal{A}$ and $\mathcal{B}$ are skeleta. In this case functors are a subdomain of the union, over functions $f : \text{obj}\mathcal{A} \to \text{obj}\mathcal{B}$ of products: $\prod_{(a,b) \in \text{obj}\mathcal{A}} \text{morph}_\mathcal{B}[f[a], f[b]]$. But smallness implies that this is a set.
5 Well-orders

We briefly recall the properties of well orders, and show that there is a universal almost well-ordered domain. Basic properties of this domain are described. Only first-order quantification is used in this section.

5.1 Definitions

Suppose \((A, =)\) is a logical domain.

1. A linear order is a pairing \((#1 \geq #2) : A \times A \to \text{y/n}\) that is transitive; ie. any two elements are related: \((a \geq b)\) or \((b \geq a)\); and elements related both ways are the same: \((a \geq b) \& (b \geq a) \iff (a = b)\).

2. Suppose \((A, \geq)\) is a linear order. A logical subdomain (logical function) \(B \subset A\) is said to be saturated if \((a \in B) \& (a \geq b) \implies b \in B\).

3. \((A, \geq)\) is a well-order if it is a linear order, \(A\) is a set, and if \(B\) is saturated then either \(B = A\) or the complement \(A - B\) has a least element.

4. \((A, \geq)\) is an almost well-order if subdomains of the form \((x > #)\) are well-ordered.

Notes

1. “Saturated” corresponds to “transitive” in traditional set theory; see Jech [Jech], definition 2.9. We prefer to reserve the term “transitive” for pairings.

2. The definition of ‘well-order’ is slightly different from the usual one (cf. Jech [Jech] definition 2.3), but it is equivalent and slightly better for the development here. As usual, well-orders are hereditary in the sense that if \((A, \geq)\) is well-ordered, and \(B \subset A\) is a logical subdomain, then the induced order on \(B\) is a well-order. Similarly for almost well-orders. If \(B\) is nonempty then it has a unique minimal element denoted by \(\min[B]\).

3. We will see that an almost well-order fails to be a well-order if and only if the domain is not a set (fails to support quantification), §5.3. It will turn out that there is only one of these, up to order-isomorphism.

Well-ordered equivalence classes

This is a variation on the description of ‘Domains of equivalence classes’ in §3.1. We will use this construction in the definition of the universal almost well-order.

A linear pre-order consists of an object descriptor \(\in A\) and a logical pairing \(\text{geq}[#1, #2]\) defined on pairs of outputs from the object descriptor \(\). The pairing satisfies:

1. (transitive) \(\text{geq}[a, b] \& \text{geq}[b, c] \Rightarrow \text{geq}[a, c]\).
2. (reflexive) \( \geq[a,a] \); and

3. (pre-linear) for all \( a, b \in A \), either \( \geq[a,b] \) or \( \geq[b,a] \) (or both).

Given this structure, define \( \equiv[a,b] := (\geq[a,b] & \geq[b,a]) \). The quotient \( A/\equiv \) is a logical domain, and \( \geq \) induces a linear order in the ordinary sense on elements (i.e., on equivalence classes of objects). The additional conditions used to define well-orders in this context are the same as those on the element level.

5.2 Recursion

Our formulation is slightly different from the standard one (cf. [Jech], Theorem 2.15) in part because we do not find the standard one to be completely clear. This version is for well-orders. There is a version for well-founded partial orders in §7.2.

First we need a notation for restrictions. Suppose \( f : A \to B \) is a partially-defined function and \( D \subset A \) is a set. Then \( f \upharpoonright D \) is the restriction to \( \text{dom}[f] \cap D \).

Now, suppose \( (A, \geq) \) is an almost well-order, \( D \) is a domain, and \( R \) is a partially-defined function \( R : \text{pfn}[A,D] \times A \to \text{y/n} \). Here ‘\( \text{pfn} \)’ denotes partially-defined functions whose domains are sets; according to ‘Functions with set support’ in §3.4, this is a logical domain and it is reasonable to think about functions defined on it. We refer to such an \( R \) as a recursion condition.

A function \( f : A \to B \) is \( R \)-recursive if:

1. \( \text{dom}[f] \) is saturated; and

2. for every \( c \in \text{dom}[f] \), \( f[c] = R[f \upharpoonright (\# < c), c] \).

Note that this is hereditary in the sense that if \( D \subset \text{dom}[f] \) is saturated then the restriction \( f \upharpoonright D \) is also recursive.

**Proposition.** (Recursion) If \( (A, \geq), B, R \) are as above, then there is a unique maximal \( R \)-recursive (partially-defined) function \( r : A \to B \).

“Maximal” refers to domains: \( r \) is maximal if there is no recursive function with larger domain. We describe a criterion for maximality below. Note that the domain of \( r \) may not be a set (i.e., may be all of \( A \)); the recursion condition applies to restrictions to subdomains of the form \( \# < a \), and according to the definition of a ‘almost’ well-order these are all sets.

The proof is essentially the same as the classical one; we sketch it to illustrate the slightly non-standard definitions. First, if \( f, g \) are recursive and have the same domain, then they are equal. Suppose not and let \( a \) be the least element on which they differ. Minimality of \( a \) implies the restrictions to \( \# < a \) are equal, but then

\[
 f[a] = R[f \upharpoonright \# < a] = R[g \upharpoonright \# < a] = g[a]
\]

a contradiction.
The maximal $r$ has domain the union $\cup (\text{dom}[f] \mid f \text{ is } R - \text{recursive})$. If $x$ is in this union then $x \in \text{dom}[f]$ for some recursive $f$. Define $r[x] := f[x]$. Uniqueness implies this is well-defined, and is recursive.

A recursive $r$ is maximal if $\text{dom}[r] = A$ or, if $\text{dom}[r] \neq A$ and $a$ is the least element not in $\text{dom}[r]$, then $(r, a)$ is not in the domain of $R$.

**Order isomorphisms**

The first application of recursion is a key result in traditional set theory. It is extended to well-founded pairings in §7.2.

**Proposition.** Suppose $(A, \geq), (B, \geq)$ are almost well-orders. Then there is a unique maximal order-isomorphism, from a saturated subdomain of $A$ to a saturated subdomain of $B$. Maximality is characterized by either full domain or full image.

This results from the recursion condition $R[f, a] := \min[B - \text{im}[f]]$.

**5.3 Universal almost well-order**

In this section the domain $W$ is defined, and shown to be universal for almost well-orders. $W$ corresponds to the “ordinal numbers” of classical set theory, cf. Jech [Jech], §2.

**Definition of $W$**

$W$ is the quotient of an equivalence relation on a descriptor $\text{pre}W$.

1. the object descriptor is defined by: $(A, \geq) \in \text{pre}W$ means “$(A, \geq)$ is a well-ordered set”;

2. $\text{geq}[(A, \geq), (B, \geq)] := (\text{im}[r] ?= B)$, where $r : A \to B$ denotes the maximal order-isomorphism of saturated subdomains described just above.

We expand on (2). Since $A$ is a set, the image $\text{im}[r]$ is a logical function on $B$. Since $B$ is a set the comparison of logical functions $\text{im}[r] ?= B$ is a logical function. Finally, since $r$ is uniquely determined by $A, B$ and their well-orders, $\text{im}[r] ?= A$ is a logical function of $A, B$.

$(\text{pre}W, \text{geq})$ is a linear pre-order, as defined above. $(W, \geq)$ denotes the quotient domain with its induced linear order. The elements of the domain $W$ are order-isomorphism classes, and we denote the class represented by $(A, \geq)$ by $\langle A, \geq \rangle$ (angle brackets).

**Canonical embeddings**

Recall that if $(A, \geq)$ is almost well-ordered and $x \in A$ then $(x > \#) \subset A$, with the induced order, is well-ordered. It therefore determines an equivalence class $\langle (x > \#), \geq \rangle \in W$. This defines the canonical embedding $\omega : A \to W$. Explicitly, $\omega[x] = \langle (x > \#), \geq \rangle$. 

23
Theorem. (Ordinals)

1. \((\mathcal{W}, \geq)\) is almost well-ordered, but not well-ordered because it does not support quantification;

2. If \((A, \geq)\) is almost well-ordered then the canonical embedding \(\omega: A \rightarrow \mathcal{W}\) defined above is an order-isomorphism to a saturated subdomain, and \(\omega\) is uniquely determined by this property;

3. If \(A\) is a set then the image of \(\omega\) is \((\# < (A, \geq))\); if \(A\) is not a set (does not support quantification) then the image is all of \(\mathcal{W}\).

Proof of the theorem

First, the properties of maximal order-isomorphisms described in §5.2 imply that \((\mathcal{W}, \geq)\) is a linear pre-order. We next see that it is almost well-ordered. If \(a \in \mathcal{W}\) then \(a\) is an equivalence class of well-ordered domains \((A, \geq)\). As explained in “canonical embeddings”, there is an order-preserving bijection, from \((A, \geq)\) to the subdomain \((A, \geq) > \#) \subset \mathcal{W}\).

Since \(A\) is well-ordered, so is the indicated subdomain of \(\mathcal{W}\). But this is the definition of “almost” well-order.

This proves all but the second halves of (1) and (3) are verified. We take these up next.

Quantification fails in \(\mathcal{W}\)

To begin, we make explicit that quantification is the issue.

First, an almost well-order is a well-order if and only if the domain supports quantification (is a set). One direction is clear: a well-order is a set by definition. For the converse suppose \((A, \geq)\) is an almost well-order on a set, and suppose \(h\) is a saturated logical function on \(A\). Since \(A\) is a set, \(\neg[h] \neq \emptyset\) is a logical function that returns either yes, in which case \(h = A\), or no, in which case \(\exists x \mid h[x] = \text{no}\). Saturation implies that \(h[y] = \text{no}\) for \(y \geq x\), so \(h\) has domain contained in \(x > \#\). But \(x > \#\) is well-ordered, so either \(\neg[h] \neq (x > \#)\) or there is \(x > z\) so that \(h[z] = (z > \#)\). In either case \(h\) has the form required to show \(A\) is well-ordered.

Next we use a form of the Burali-Forti paradox to show that \((\mathcal{W}, \geq)\) cannot be a well-order. If it were, then by definition of \(\mathcal{W}\) there would be \(x \in \mathcal{W}\) and an isomorphism \(\omega: \mathcal{W} \cong (x > \#)\). But then, \(\omega^2\) gives an isomorphism with \((\omega[x] > \#)\). Since \(\omega[x] < x\), this contradicts the fact that isomorphism classes of well-orders correspond uniquely to elements of \(\mathcal{W}\). Thus \((\mathcal{W}, \geq)\) cannot be a well-order, and therefore does not support quantification.

The final part of the theorem is to see that if \(A\) is an almost well-order and not a set then the canonical embedding is an isomorphism to \(\mathcal{W}\). Suppose there is \(x\) not in the image. The image is saturated, so must be contained in \((\# < x)\). But this is well-ordered, so a saturated subdomain of it must be a set. This
contradicts the hypothesis that $A$ is not a set. We conclude that there cannot be any such $x$, and therefore $\omega$ is onto. It follows that it is an order-isomorphism.

This completes the proof of the theorem. \qed

5.4 Existence of well-orders

We review a crucial classical theorem, and get some information about $W$.

**Proposition.** Suppose $A$ is a logical domain.

1. If $A$ is a set then it has a well-order.
2. If $A$ is not a set then there is an injection $\mathbb{W} \to A$.
3. There is no logical function that detects which of these alternatives holds.

An version of this is used to illustrate assertion logic in ‘von Neumann’s axiom’ in §2.2.

**Proof:** Let $s\mathcal{P}[A]$ denote the logical functions on $A$ whose supports are sets, and $s\mathcal{P}[A] & (\# \neq A)$ the ones whose support is not all of $A$. Comments on the logic: if $A$ is a set then $s\mathcal{P}[A] = \mathcal{P}[A]$ is a logical domain and (because $A$ supports quantification) $\# \neq A$ is a logical function on it. If $A$ is not a set then $\mathcal{P}[A]$ is not a logical domain. The subdescriptor $s\mathcal{P}[A]$ is a logical domain, but we do not need this now. More to the point, $(\# \neq A)$ still makes sense as a logical function because it is always ‘yes’.

Define a **choice function** for $A$ to be $\text{ch}: (s\mathcal{P}[A] & (\# \neq A)) \to A$, satisfying $h[\text{ch}[h]] = \text{no}$ (ie. $\text{ch}[h]$ is in the complement of $h$). The axiom of Choice implies that any $A$ has a choice function, as follows: Define $c: (s\mathcal{P}[A] & (\# \neq A)) \times A \to \text{y/n}$ by $c[h, a] := \text{not}[h[a]]$. The projection of (the support of) $c$ to $s\mathcal{P}[A]$ is known to be onto, due to the “not all of $A$” condition. According to Choice, there is a section of this, and sections are exactly choice functions.

We now set up for recursion. Fix a choice function and define a condition $R: \text{pf}\text{n}[\mathbb{W}, A] \times A \to \text{y/n}$, where ‘$\text{pf}\text{n}[\mathbb{W}, \#]$’ denotes partially-defined functions with bounded domain, by:

$$R[f, a] := \text{ch}[\text{im}[f \upharpoonright (a > \#)]].$$

In words, $f[a]$ is the chosen element in the complement of the image of the restriction $f \upharpoonright (a > \#)$.

This is clearly a recursive condition. We conclude that partially-defined functions $r: \mathbb{W} \to A$ satisfying:

1. $\text{dom}[r] \subset \mathbb{W}$ is saturated;
2. $f[a] = R[r \upharpoonright (a > \#), a]$ holds for all $a \in \text{dom}[r]$.

form a linearly-ordered domain with a maximal element, and $r$ is maximal if and only if either $\text{im}[r] = A$ or $\text{dom}[r] = \mathbb{W}$.

If $\text{dom}[r] = \mathbb{W}$ then $r$ gives an injective function $\mathbb{W} \to A$ and $A$ cannot be a set. If $\text{dom}[r] \neq \mathbb{W}$ then $r$ gives a bijection from a saturated proper subdomain of $\mathbb{W}$ to $A$. But a saturated proper subdomain has a well-order, so $A$ must have one also. \qed
5.5 Cofinality in $\mathbb{W}$

This clarifies the structure of $\mathbb{W}$ “near infinity”. Suppose $(A, \geq)$ is almost well-ordered and $A$ does not have a maximal element. A logical function $h : A \to \{y/n\}$ is said to be **cofinal** if for every $a \in A$ there is $b \in h$ such that $b \geq a$. See Jech [Jech] §3.6. Caution: if $A$ is well-ordered then “is $h$ cofinal?” is a logical function on $P[A]$. If $A$ is not well-ordered (ie. is $\mathbb{W}$) then this is not a logical function and “for every” has to be interpreted as an assertion.

**Proposition.** A logical function $h$ on $\mathbb{W}$ is (known to be) cofinal if and only if $h$ with the induced order is (known to be) order-isomorphic to $\mathbb{W}$.

This is essentially the traditional definition of “regular cardinal”. In these terms the almost-cardinal $\beth$ is regular.

**Proof:** we show that $h$ is a set if and only if it is not cofinal. If it is not cofinal then there is $m$ larger than any element of $h$. This means $h$ is a logical function on $(m > #)$. But this is a set and a logical subdomain of a set is a set.

For the converse, define a function $p : \mathbb{W} \to h$ by $p[a] := \min[h[#] \& (a > #)]$. If $b \in h$ then (since $h$ is cofinal) the preimage $p^{-1}[b]$ is bounded and therefore a set. According to the Surjection proposition at the end of §3.4, this implies $\mathbb{W}$ is known to be a set if and only if $h$ is known to be a set. But $\mathbb{W}$ is not a set so neither is $h$. The induced order on $h$ is an almost well-order, so by the last part of the Main Theorem, it is order-isomorphic to $\mathbb{W}$. □

Global choice of well-orders

We give the global analog of the proposition above, then apply it to extend the characterization of $\mathbb{W}$.

**Lemma.** There is a morphism of object descriptors that assigns to each set a well-order on it.

Denote by WO the descriptor with objects, well-orders $(A, \geq)$, and Set the descriptor with objects sets. Forgetting the order gives a morphism $\text{WO} \to \text{Set}$. Since every set has a well-order, this morphism is known to be onto. The axiom of Choice implies this has a section. This section is the morphism required for the Lemma. □

**Corollary.** Suppose $f : A \to \mathbb{W}$ is a function with set point inverses and cofinal image. Then there is a bijection $b : \mathbb{W} \to A$ so that $f \circ b$ is nondecreasing.

The rule $a \mapsto f^{-1}[a]$ defines a morphism $\mathbb{W} \to \text{Set}$. Compose this with a morphism that provides a well-order on each set. This gives a well-order $(f^{-1}[a], \geq_a)$, for every $a$. Now define an order on $A$ by:

$$(x \geq y) := (f[x] > f[y]) \text{ or } (f[x] = f[y] \& (x \geq f[x] y)).$$

This is easily seen to be an almost well-order, and $f$ is nondecreasing with respect to this order. Finally, a cofinal subdomain of $\mathbb{W}$ is order-isomorphic to $\mathbb{W}$. By the characterization theorem above, $(A, \geq)$ must be order-isomorphic to $\mathbb{W}$. □
6 Cardinals

Cardinals have been studied for well over a century. We need a few of the most basic results, transposed into the current setting.

6.1 Definition of cardinals

Cardinal elements and well-orders

Recall that an element of $\mathcal{W}$ is an order-isomorphism class of well-orders. Bi-
jection of underlying sets induces an equivalence relation on $\mathcal{W}$; explicitly, \( \langle A, \geq \rangle \simeq \langle B, \geq \rangle \) if there is a bijection $A \to B$. An element of $\mathcal{W}$ is defined to be a cardinal element if it is the minimum of its equivalence class.

A cardinal well-order is $\langle A, \leq \rangle$ whose equivalence class in $\mathcal{W}$ is a cardinal
element.

The point is that a cardinal well-order is a specific well-order, while cardinal
elements are equivalence classes. These are closely connected, but not quite
the same, and it clarifies some discussions to have both terms. Or more to the
point, traditional texts tend not to distinguish them and some discussions are
muddled by this.

Next we define cardinality of sets. If $A$ is a set, define $\text{card}[A]$ to be the
minimum of the bijection equivalence class of $(A, \geq)$, for any well-order $\geq$ on
$A$. Since bijection-equivalence depends only on the underlying set, this is well-
defined.

Injections

Lemma. If $A,B$ are sets then there is an injection $A \to B$ if and only if
$\text{card}[B] \geq \text{card}[A]$.

Choose well-orders so $\text{card}[A]$ is the equivalence class of $(A, \geq)$ and similarly
for $\text{card}[B]$. If $\text{card}[B] \geq \text{card}[A]$ then, by definition of the ordering in $\mathcal{W}$, there
is an order-preserving injection $A \to B$. This proves the ‘if’ part of the lemma.

The ‘only if’ direction is proved by contradiction. Suppose there is an in-
jection $A \to B$ but $\text{card}[A] > \text{card}[B]$. This and the ‘if’ direction implies there
is an injection $B \to A$. If there are injections both ways the Cantor-Bernstein
theorem ([Jech], Theorem 3.2) asserts that there is a bijection $A \simeq B$. But
then $\text{card}[A] = \text{card}[B]$, contradicting the assumed cardinal inequality. Thus
$\text{card}[A] \leq \text{card}[B]$.

We note that the Cantor-Bernstein theorem requires knowing that the images
of injections involved are logical. This follows from Lemma 3.4.

Cofinality

Lemma. Cardinal elements are cofinal in $\mathcal{W}$.  

27
If not then there is $a \in \mathcal{W}$ larger than any cardinal. But Cantor’s theorem implies $\text{card}[(\# < a)] < \text{card}[\mathcal{P}[\# < a]]$. This gives a cardinal larger than $a$, a contradiction.

An immediate consequence is:

**Corollary.** A subdomain of a set is a set (ie. is logical).

Proof: Suppose $A$ is a set, and suppose $X \subset A$ is a non-logical subdomain. Non-logical implies $X$ is not a set, which in turn implies that there is an injection $\mathcal{W} \to X$. Restricting this gives an injection $(b > \#) \to A$, for any $b \in \mathcal{W}$. The image of such an injection is logical because $(b > \#)$ is a set. But since $b$ is arbitrary, we can choose it so that $\text{card}[b > \#] > \text{card}[A]$. This cardinal inequality implies there cannot be an injection $(b > \#) \to A$. This contradiction refutes the assumption that $X \subset A$ is non-logical.

### 6.2 The almost-cardinal ‘Shin’

$\mathcal{W}$ does not have a cardinality because it cannot be well-ordered, but it has many of the properties of a cardinal domain. For convenience we extend the definition.

Denote $\mathcal{W}$ with a maximal element adjoined by $\overline{\mathcal{W}}$; this is the “completion” of $\mathcal{W}$. The new maximal element is denoted by $\mathcal{W}$ (the Hebrew character ‘shin’). If $A$ is a domain then $\text{card}[A] = \mathcal{W}$ means: it is known there is a bijection $A \simeq \mathcal{W}$. This has to be used with care: $\text{card}[A] = \mathcal{W}$ does not mean that $A$ has a genuine cardinality, and it happens to be $\mathcal{W}$. This interpretation would give a logical function that detects $\mathcal{W}$, and this is impossible.

### 6.3 The Cantor Beth function

Cantor introduced several functions related to cardinals. Only one is needed here.

**Definition**

‘Beth’ ($\beth$) is the second character in the Hebrew alphabet, and was used by Cantor to denote a function $\mathcal{W} \to \mathcal{W}$ that might also be called the “iterated powerset function. $\beth$ is briefly mentioned in [Jech] §5, p. 55. This, and the associated rank function, play major roles in the construction of the ZFC set theory in §7.

**Proposition.** There is a unique function $\beth: \mathcal{W} \to \mathcal{W}$ satisfying:

1. if $a = 0$, $\beth[a] = \text{card}[N]$;
2. if $a > 0$ is not a limit, $\beth[a] = \text{card}[\mathcal{P}[\beth[a - 1]]]$; and
3. if $a$ is a limit, $\beth[a] = \sup[\beth[\# < a]]$. 

28
‘sup’ in (3) is the ‘supremum’, which is a convenient shorthand for the minimum of larger elements: \( \text{sup}[h] := \min \{ \# | (\forall a \: | \: h[a] = \text{yes}) \} \).

It is straightforward to formulate a recursive condition on partially defined functions so that conditions (1)–(3) correspond to fully recursive. Existence of a maximal such function therefore follows from recursion. The image of \( \boxdot \) is cofinal in \( \mathbb{W} \) for the same reason cardinals are, and this implies the maximal function has domain all of \( \mathbb{W} \).

We extend Beth to the almost-cardinal by \( \beth[\mathbb{W}] := \mathbb{W} \).

Limits and strong limits

Recall that \( a \in \mathbb{W} \) is a limit if \((a > \#)\) does not have a maximal element. It is a strong limit if \( a > b \) implies \( a > \text{card}[P[b > \#]] \). The almost-cardinal \( \mathbb{W} \) is a strong limit, and strong limits less than \( \mathbb{W} \) are cardinals.

**Proposition.**

1. \( b \leq \beth[b] \);

2. \( a \in \mathbb{W} \) is a strong limit if and only if \( a = \beth[x] \) for either \( x = 0 \) or \( x \) a limit and

3. if \( b \) is a limit then the image \( \beth[b > \#] \) is cofinal in \((\beth[b] > \#)\).

**Proof:** (1) is standard, and easily proved by considering the least element that fails.

For (2), suppose \( a \) is a strong limit. \( \beth \) is increasing, so \( a > \beth^{-1}[\#] \) is saturated and therefore of the form \( x > \# \). Since \( b \notin (\beth^{-1}[a > \#]) \), \( \beth[b] \geq a \).

We show \( b \) is a limit. If not then \((b > \#)\) has a maximal element, \( b - 1 \).
\( a > \beth[b - 1] \), so by definition of strong limit \( a > \beth[b - 1] > \text{card}[P[\#]] \), but the left side is the definition of \( \beth[b] \), so this contradicts the choice of \( b \). We conclude \((b > \#)\) does not have a maximal element, so \( b \) is a limit.

Next, \( \beth[b > \#] \) is bounded by \( a \), so \( \sup[\beth[b > \#]] \) is defined, is \( \leq a \), and is the definition of \( \beth[b] \). But the choice of \( b \) requires \( \beth[b] \geq a \), so \( \beth[b] = a \), as required.

For the other direction of ‘if and only if’, suppose \( b \) is a limit. We want to show that \( a = \beth[b] := \sup[\beth[b > \#]] \) is a strong limit. Suppose \( a > x \). Then there is \( b > y \) with \( \beth[y] \geq x \). Thus \( \text{card}[P[x > \#]] \leq \text{card}[P[\beth[y] > \#]] \leq \text{card}[P[x > \#]] \). Next, since \( b \) is a limit, \( b > y + 1 \). But the definition of \( \beth \) gives \( \text{card}[P[\# < \beth[y]]] = \beth[y + 1] \). Since \( \beth[y + 1] < \sup[\beth[b > \#]] = a \), we get \( a > \text{card}[P[x > \#]] \). This verifies the definition of strong limit.

The cofinality conclusion follows from the definition of ‘sup’. □

A corollary of (3) in the Proposition is that the cofinalities are the same: \( cf[\beth[b]] = cf[b] \). Therefore if \( b < \beth[b] \), or \( b \) is not regular, then \( \beth[b] \) is not regular.

Beth rank

Rank plays a central role in the construction of set theory in §7. This rank follows the \( \beth \) function closely, and is a bit different from the rank function.
defined in [Jech] §6.2: it starts with \( \text{rank} = 0 ⇔ \text{‘finite’} \), rather than \( \text{rank} = 0 ⇔ \text{‘empty’} \), and otherwise differs by +1.

The (Beth) \textbf{rank} is the function \( \mathcal{W} \to \mathcal{W} \) given by
\[
\text{rank}(x) := \min(\# | \mathcal{Z}(\#) > x)
\]

Note that if \( a \) is a limit then there are no elements of rank \( a \). The reason is that \( \mathcal{Z}(a) = \sup(\mathcal{Z}(a > \#)) \), so if \( \mathcal{Z}(a) > x \) then there is some \( a > b \) with \( \mathcal{Z}(b) > x \). These hidden values come into play in ranks of logical functions, defined next.

If \( h : \mathcal{W} \to \{y/n\} \) is a set (ie. has bounded domain) then define
\[
\text{rank}(h) := \sup(\text{rank}(\# | h(\#)))
\]

If \( h(\#) = (\# = a) \), the function that detects \( a \), then the ranks of \( a \) and \( h \) are the same. More generally, \( h[a] \Rightarrow (\text{rank}[a] ≤ \text{rank}[h]) \).

If \( h \) is cofinal in \( g \) then \( \text{rank}[h] = \text{rank}[g] \).

Finally, consider \( A = (a > \#) \). \( \text{rank}[a > \#] = \text{rank}[a] \) unless \( a = \mathcal{Z}[b] \) for some \( b \), in which case \( \text{rank}[a > \#] = b \) and \( \text{rank}[a] = b + 1 \).

### 6.4 Hessenberg’s theorem

In the classical development this is a key fact about cardinals. Here it is a key ingredient in showing ‘relaxed sets’ have the properties expected of sets. One of the consequences is that if \( A \) is an infinite set then \( \text{card}[A \times A] = \text{card}[A] \). We go through the proof to check the use of quantification, and because the traditional proof is somewhat muddled by the identification of sets and elements.

**The canonical order**

Suppose \( (A, ≥) \), is a linear order. The \textbf{canonical} order on \( A \times A \) is a partially-symmetrized version of lexicographic order.

First define the maximum function \( \max : A \times A \to A \) by \( (a, b) → \max[a, b] \).

This induces a pre-linear order on \( A \times A \), namely
\[
(a_2, b_2) > (a_1, b_1) := \max[a_2, b_2] > \max[a_1, b_1].
\]

The canonical order refines this to a linear order as follows: Fix \( c ∈ A \), then the elements of \( A \times A \) with \( \max \) equal to \( c \) have the form \( (c > \#, c) \) or \( (c, c ≥ \#) \). Each of these is given the order induced from \( A \), and pairs of the first form are defined to be smaller than pairs of the second form. The canonical order is denoted by \( ≥_{\text{can}} \).

More explicitly, \( (a_2, b_2) ≥_{\text{can}} (a_1, b_1) \) means:

\[
(\max[a_2, b_2] > \max[a_1, b_1]) \quad \text{or} \quad
((\max[a_2, b_2] = \max[a_1, b_1]) \quad \text{and} \quad ((a_2 > a_1) \quad \text{or} \quad (a_2 = a_1 \quad \text{and} \quad b_2 > b_1)).
\]

The following is straightforward:
Lemma. If \((A, \geq)\) is a well-order then the canonical order \((A \times A, \geq_{can})\) is a well-order. If \(\geq\) is an almost well-order then so is \(\geq_{can}\).

This implies that the classifying function gives an order-preserving bijection

\[
\omega: (\mathcal{W} \times \mathcal{W}, \geq_{can}) \to (\mathcal{W}, \geq).
\]

The main result is a century-old theorem of Hessenberg (see [Jech], Th. 3.5.)

Proposition. If \(c\) is an infinite cardinal then

1. \(\omega[(0, c)] = c\), or equivalently
2. \(\omega[(c > #) \times (c > #)] = (c > #)\)

Version (1) concerns images of elements, while (2) concerns images of saturated subdomains. They are equivalent because

\[(c > #) \times (c > #) = ((0, c) >_{can} (#1, #2)).\]

The first step is that \(\omega[0, c] \geq c\). Note that \((c > #) \times (c > #)\) contains a copy of \((c > #)\), so \(\text{card}[(c > #) \times (c > #)] \geq \text{card}[(c > #)] = c\). But \(\omega\) is a bijection, so the image has cardinality \(\geq c\). The image is \(\omega[0, c] > #\), so by minimality of cardinals we have \(\omega[0, c] \geq c\).

Next we work out the consequences of strict inequality in this last: suppose \(\omega[0, c] > c\). Since \(\omega\) is order-preserving, this implies there is \((0, c) >_{can} (x, y)\) with \(\omega[x, y] = c\). Denote the maximum of \((x, y)\) by \(m\), then \(m < c\). Since cardinals are limits, the successor \(m + 1\) is also less than \(c\). Turning to the subdomain form, this means \((m + 1 > #) \times (m + 1 > #)\) contains \((x, y) >_{can} (#1, #2)\), which is the preimage of \((c > #)\). Thus \(\text{card}[(m + 1 > #) \times (m + 1 > #)] \geq c\). Finally, define \(d := \text{card}[m + 1]\). This gives a cardinal \(c > d\) so that \(\text{card}[(d > #) \times (d > #)] \geq c\).

We proceed by induction on \(c\) to show this cannot happen. To begin, suppose \(c = \text{card}[\mathbb{N}]\), the first infinite cardinal. If \(\omega[0, c] \neq c\) then we get \(d\) with \(c > d\), as above. But the domain \((d > #)\) is finite, so the product with itself is finite, and it is false that the cardinality of the product is \(\geq \text{card}[\mathbb{N}]\). Therefore \(\omega[0, \text{card}[\mathbb{N}]] = \text{card}[\mathbb{N}]\).

Now suppose the proposition is false, and let \(c\) be the least cardinal for which \(\omega[0, c] > c\). We get \(c > d\) as above, with \(\text{card}[(d > #) \times (d > #)] \geq c\). But the induction hypothesis implies that \(\text{card}[(d > #) \times (d > #)] = d\), and \(d < c\). This gives a contradiction, so the proposition must be true.

Products and unions

For finite sets, the cardinality corresponds to the number of elements. So, cardinality of a disjoint union is the sum of the cardinalities, and cardinality of a product is the product. The preceding section implies that the situation is much simpler for infinite sets.
**Proposition.** Suppose $A, B$ are sets, and at least one is infinite. Then

1. $|A \times B| = \max\{|A|, |B|\}$, and

2. if $A, B \subseteq D$ then $|A \cup B| = \max\{|A|, |B|\}$. 

Proof: suppose $|A| \geq |B|$, so $\max\{|A|, |B|\} = |A|$. Then

$$|A| \leq |A \times B| \leq |A \times A| = |A|$$

The last step being the Proposition above. This gives (1).

For (2), $|A| \leq |A \cup B| \leq |A \times B| = |A|$, by (1).
7 The universal ZFC theory

This section provides an interface between the present theory and traditional
axiomatic set theory. The main result is a theory that satisfies appropriate
versions of the Zermello-Fraenkel-Choice (ZFC) axioms, and is maximal in the
sense that any other ZFC set theory injects into it.

7.1 Main result

First, a definition.

**Bounded and cofinal logical functions**

Suppose \((A, \leq)\) is almost well-ordered.

The **bounded** logical functions, or **bounded powerset**, denoted \(bP[A]\),
consists of the \(h: A \rightarrow \{y/n\}\) with \(a \in A\) such that \(h[\#] \Rightarrow \# < a\).
The **cofinal** logical functions, denoted \(cfP[A]\) are the \(h\) so that if \(a \in A\) then there is \(b \geq a\) with \(h[b] = yes\).

If \(A\) has a maximal element \(m\) then these reduce to standard things: \(bP[A]\)
consists of the \(h \in P[A]\) with \(h[m] = no\), and \(cfP[A]\) consists of the \(h\) with \(h[m] = yes\).

If \(A\) is a set then these are complementary subsets of \(P[A]\), in the sense that
there is a logical function \(bP: P[A] \rightarrow \{y/n\}\) with support the evident subdomain,
and \(cfP[A] = \text{not}[bP]\). If \(A\) is not a set (ie. \(A = \mathbb{W}\)) then they are still complementary in a sense, but are not logical. In this case ‘bounded’ should therefore be understood as ‘known to be bounded’, and ‘cofinal’ similarly.

We come to the main result. Some of the terms are defined below.

**Theorem.**

1. *(Existence)* There is a pairing \(\exists: \mathbb{W} \times \mathbb{W} \rightarrow \{y/n\}\) whose adjoint
gives a bijection \(\exists^{adj}: \mathbb{W} \rightarrow bP[\mathbb{W}]\), such that

   (a) the restriction \(N \rightarrow bP[N]\) is the canonical order isomorphism; and

   (b) if \(x \notin N\) then \(\text{rank}[\exists^{adj}[x]] = \text{rank}[x] - 1\).

2. *(Axioms)* Such pairings satisfy all the ZFC axioms, appropriately interpreted.

3. *(Universality)* Suppose \((A, \lambda)\) is an almost well-founded pairing (§7.2).
   Then there is a unique function giving a morphism of pairings \((A, \lambda) \rightarrow (\mathbb{W}, \exists)\).

Clarifications: the adjoint of ‘\(\exists\)’ takes elements of rank 1 to the cofinal
functions on \(N\). Technically, all functions on \(N\) have rank 0, but the bounded ones are accounted for by condition (1a). In (1b) recall that ranks of elements cannot be limits, so \(\text{rank}[x] - 1\) is defined. Finally, the notation ‘\(\exists\)’ is a reminder that this is the opposite of the standard ‘\(\in\)’ pairings.

Two such pairings ‘\(\exists\)’ differ by a bijection \(\mathbb{W} \rightarrow \mathbb{W}\) that is the identity on \(N\) and preserves the rank function.
The universality property is a version of Mostowski collapsing ([Jech], §6.15). It has several striking corollaries: First, since ZFC theories are well-founded, every ZFC theory whose sets are relaxed sets, canonically embeds into \((W, \succ)\). Note that the relaxed sets in a ZFC theory form a sub-ZFC theory, so this observation applies to a large part (probably all) of any ZFC theory. Second, the usual uniqueness of universal objects shows that this property characterizes \((W, \succ)\) up to (unique) equivalence.

See §7.2 below for details about well-founded pairings, and morphisms thereof.

**Proof of Existence**

We will show that for every \(a\) there is a bijection, from elements of rank \(a + 1\) to logical functions of rank \(a\). The rest of the proof is short: According to the axiom of choice we can make a simultaneous choice of bijections for all \(a\). These fit together to give a bijection \(W \rightarrow bP[W]\). This is the adjoint of a pairing with the properties with properties (1a) and (1b).

Now we see that there are bijections as claimed. Suppose \(a \in W\). There is a bijection from elements of rank \(a + 1\) to functions of rank \(a\) if these collections have the same cardinality.

The elements of rank \(a + 1\) are \(\|a\| \leq \# < \|a\| + 1\). This is the complement of \(\# < \|a\|\) in \(\# < \|a + 1\|\). Since \(\|a + 1\| = \text{card}(P[\# < \|a\|])\), we are removing a set of smaller cardinality. According to §6.4, this does not change cardinality. The cardinality of the collection of elements is therefore \(\text{card}(P[\# < \|a\|])\).

There are two cases for functions. Suppose \(a\) is not a limit. Then functions of rank \(a\) are the complement of \(P[\# < \|a - 1\|]\) in \(P[\# < \|a\|]\). Again the smaller subdomain has smaller cardinality, so removing it does not change cardinality. Cardinality of the functions is therefore \(\text{card}(P[\# < \|a\|])\), same as the elements.

Now suppose \(a\) is a limit. The functions on \(\# < \|a\|\) of rank \(a\) are the cofinal ones. These are the complement of the bounded functions: \(\text{cf}(P[\# < \|a\|]) = P[\# < \|a\|] - bP[\# < \|a\|]\). We want to show that removing the bounded functions does not change the cardinality. For this it is sufficient to show that the cardinality of the complement is at least as large as that of the collection being removed. This is so because there is an injection \(bP[\# < \|a\|] \rightarrow \text{cf}(P[\# < \|a\|])\) defined by \(h \mapsto \text{not}[h]\). The conclusion is that the cardinality of the functions is \(\text{card}(P[\# < \|a\|])\), again same as the elements. \(\square\)

**7.2 Universality**

We begin with translations and slight modifications of some standard material about well-founded pairings, see [Jech], §6. Then well-founded recursion is used to prove the theorem.

**Well-founded pairings**

Suppose \(\lambda: A \times A \rightarrow y/n\) is a logical pairing on a set.
1. $\lambda$ is a **partial order** if it is transitive and $\lambda[a, a] = \text{no}$ for all $a$. We typically denote partial orders by $a \succ b$.

2. Any pairing has a transitive closure. The closure is a partial order if the $a \not\succ a$ condition is satisfied.

3. A subset $B \subset A$ is **saturated** if $(x \in B \& \lambda[x, y]) \Rightarrow (y \in B)$, as with linear orders.

4. $\lambda$ is **well-founded** if every saturated $B \subset A$ with nonempty complement has a minimal element in this complement.

5. Define $(A, \lambda)$ to be **almost** well-founded if
   
   (a) the transitive closure of $\lambda$, denoted by $\succ$, is a partial order;
   
   (b) the restriction of $\lambda$ to a subdomain of the form $(a \succ \#)$ is well-founded.

This definition of well-founded implies that that *any* nonempty subset, not just complements of saturated ones, has a minimal element. This stronger statement is usually taken as the definition, but the version given is more efficient here. For example, a pairing is well-founded if and only if its transitive closure is. This is because the pairing and its closure have the same saturated subsets and the same minimal elements in their complements, and the definition concerns only these minimal elements.

A function $f: (A, \lambda) \rightarrow (B, \tau)$ is a morphism of pairings if \( \tau[f[a], x] \Leftrightarrow (\exists b \in A \mid (f[b] = x) \& \lambda[a, b]) \). Note that this implies the image of $f$ is $\tau$-saturated in $B$.

In general, morphisms of pairings need not be injective. It is straightforward to see:

1. if the adjoint $\lambda^{\text{adj}}: A \rightarrow \mathcal{P}[A]$ is injective then $f$ is injective; and

2. if $\tau^{\text{adj}}: B \rightarrow \mathcal{P}[B]$ is injective then $f$ is unique.

The injective-adjoint condition is traditionally called “extensional”.

**Recursion**

Suppose $(A, \succ)$ is an almost well-founded partial order. As in §5.2, a **recursion condition** is a partially-defined function $R: \text{pfn}[A, D] \times A \rightarrow \text{y/n}$. As before, ‘\text{pfn}’ denotes partially-defined functions whose domains are sets.

Exactly as in §5.2, a partially-defined function $f: A \rightarrow B$ is $R$-**recursive** if:

1. $\text{dom}[f]$ is $\succ$-saturated; and

2. for every $c \in C$, $f[c] = R[f \upharpoonright (c \succ \#), c]$.

The theorem and proof of §5.2 extend without modification.

To prove universality, suppose $(A, \lambda)$ is an almost well-founded pairing. Define $R: \text{pfn}[A, \mathbb{W}] \times A \rightarrow \mathbb{W}$ by:
1. \((f, a) \in \text{dom}[R] \) if \(\lambda[a, \#] \subset \text{dom}[f] \), and in this case

2. \(R[f, a] := (\exists^{\text{adj}})^{-1}[\lambda[a, \#] * f] \).

Note that this definition uses the injectivity of the adjoint \(\exists^{\text{adj}} : \mathcal{W} \to \mathcal{B}[\mathcal{W}] \). Unwinding definitions gives \((R[f, a] \ni x) \iff (\exists b \in A \mid \lambda[a, b] \& f[b] = x) \).

The maximal \(R\)-recursive function \(f : A \to \mathcal{W} \) has domain \(A \), and the recursion condition implies that \(f \) is a morphism of pairings \((A, \lambda) \to (\mathcal{W}, \exists) \).

### 7.3 Reformulation of the ZFC axioms

A traditional set theory is a pool (class, logical domain) of potential elements, \(\Sigma \), and a binary logical operator ‘\(\in\)’. We change notation to prefix form and, to be consistent with previous material, reverse the order: define \(N : \Sigma \times \Sigma \to \mathcal{y/n} \) by \(N[a, b] := b \in a \). A set in the theory is a logical function on \(\Sigma \) of the form \(# \mapsto N[a, \#] \), for some \(a \in \Sigma \). Assume these sets are relaxed sets, by restricting the theory if necessary. See ‘Quantification in ZFC theories’ below.

Given all this, we translate the ZFC axioms described in [Jech] Ch. 1.

**Axioms**

As above, \(\Sigma \) is a logical domain, \(N : \Sigma \times \Sigma \to \mathcal{y/n} \) is a logical pairing, and we suppose \(a \in \Sigma \).

1. (Well-founded, or Regular) every nonempty set \(N[a, \#] \) contains an \(N\)-minimal element, ie \(b \) such that \(N[b, c] \Rightarrow \text{not}[N[a, c]] \).

2. Each \(N[a, \#] \) is a relaxed set, (see note 2);

3. (Extensionality) \(N[a, \#] = N[b, \#] \) implies \(a = b \) (see note 3);

4. (Union) \(\exists (\cup a) \in \Sigma \) such that \(N[\cup a, \#] = (N* N)[a, \#] \). \(N* N \) is the composition of pairings, defined by \((N* N)[a, \#] := (\exists b \mid N[a, b] \& N[b, \#]) \).

5. (Powerset) \(\exists P[a] \in \Sigma \) such that \(N[P[a], b] = (N[b, \#] \subset N[a, \#]) \);

6. (Infinity) There is \(a \) with \(N[a, \#] \) infinite;

7. (Choice) There is a partially-defined function \(ch : \Sigma \to \Sigma \) with domain \(a \in \Sigma \mid N[a, \#] \neq \Sigma \) satisfying \(N[a, ch[a]] = \text{no} \) (see note 4);

8. (Separation) If \(P \) is an appropriate logical function on \(\Sigma \) (a “property”) then there is \(a \& P \in \Sigma \) such that \(N[a \& P, \#] = N[a, \#] \& P[\#] \) (see note 5);

9. (Replacement) Suppose \(f : \Sigma \to \Sigma \) is an appropriate function. Then the \(f \) image of a set is a set (see note 6).
Notes

1. ‘Well-founded’ is usually put near the end of axiom lists. The universality theorem indicates that it is a key ingredient, so we put it first.

2. **External** quantification requires that there be a logical function, defined on all possible functions $A \to y/n$, that detects the empty function. The **Internal** quantification of ZFC requires only that $\emptyset$ be detectable among functions of the form $N[a, \#]$. In principle there could be many fewer of these so, again in principle, there might be theories with sets that support internal quantification but not external. If so we discard such sets.

3. The domain $\Sigma$ may not support quantification, so we may not be able to identify functions $N[a, \#]$ among all possible logical functions on $\Sigma$, but we can identify them within functions of the form $N[x, \#]$ as follows. $N[a, \#] = N[b, \#]$ means: $(\forall x \in N[a, \#])(N[b, x] = \text{yes})$, and similarly $(\forall x \in N[b, \#])(N[a, x] = \text{yes})$.

4. In this formulation the choice function provides an element not in the given set, rather than (as more usual) one in the set. Note that if $\Sigma$ does not support quantification then “$N[a, \#] \neq \Sigma$” does not make good sense. In this case omit this condition: it is redundant anyway because $N[a, \#]$ is assumed to be a set, so cannot be all of $\Sigma$. This is the form used to show $\Sigma$ has a well-order, which implies any other form one might want.

5. The point in “Separation” is that $a \& P$ is a set, even if $P$ is not (ie. is not of the form $N[p, \#]$).

6. The intent of “Replacement” is that functions should take sets to sets. However, the usual definition of “function” uses set theory, so using it here would make the definition logically circular. The developers of set theory eventually fell back on an earlier view of functions as “given by formulas”; see the historical notes in [Jech]. Thus, in standard ZFC, “appropriate” in Separation and Replacement means “given by an expression in the first-order logic of sets”. In the theory here, functions (in the guise of morphisms of descriptors) are primitive objects, and need not be defined. Roughly speaking, the theory here is universal because “appropriate” is a subset of “all”.

**Verifying the axioms**

We verify that $(\mathcal{W}, \ni)$ satisfies the translated axioms.

1. ‘Well-founded’ follows from the fact that the pairing reduces rank, and rank takes values in a well-ordered domain.

2. The domains $\ni [a, \#]$ are relaxed sets because they are bounded subdomains of $\mathcal{W}$.
3. ‘Extension’ is equivalent to injectivity of the adjoint of \( \exists \), and this is a design requirement in the construction.

4. ‘Infinity’ is a primitive axiom in the system used here.

5. Similarly, ‘Choice’ is a primitive axiom.

6. For Separation, note that \( \exists [a, \#] \) is bounded, so \( \exists [a, \#] \& P \) is bounded.

7. We interpret ‘Replacement’ in a strong way, namely that any function \( \mathcal{W} \to \mathcal{W} \) should take bounded logical functions to bounded logical functions. This is included in §5.5.

This completes the ZFC axioms, and the proof of the universality theorem.
8 Bibliography

References

[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)

[Manin] Yu. I. Manin: *A Course in Mathematical Logic for Mathematicians, second edition* Springer Graduate Texts in Mathematics 53 (2010)