

The convention in this thesis will be to say **ZF** when talking about Zermelo-Fraenkel set theory *without* the axiom of choice. When talking about the axiom of choice on its own we will say **AC**, and when talking about Zermelo-Fraenkel set theory together with the axiom of choice we use **ZFC**.

We will use the convention of including 0 at the beginning of the natural numbers \mathbb{N} , i.e. $\mathbb{N} = \{0, 1, 2 \dots\}$. This is a *natural* choice, since we then can use \mathbb{N} to mean the set described by the [Axiom of Infinity](#).

o.I First-order Logic and Classes

o.2 Zermelo-Fraenkel Axioms of Set Theory

We assume that the reader has some familiarity with axiomatic set theory, but for convenience and consistency we restate some of the necessary basics here. For a more thorough introduction of the topic, see [?, 4.3-4.5], alternatively [?, 1.1] gives a more technical overview. The formulation of the axioms below is based on both textbooks.

o.2.1 Axiom of Extensionality

$$\forall x \forall y (x = y \iff \forall z (z \in x \iff z \in y))$$

Two sets are equal if and only if they contain the same elements.[?, 4.3, p.76]

o.2.2 Axiom of Pairs

$$\forall x \forall y \exists z \forall w (w \in z \iff (w = x \vee w = y))$$

For any two sets, there is a set whose elements are precisely these sets.

We define an ordered pair $\langle x, y \rangle$ to be the set $\{\{x\}, \{x, y\}\}$. Further, ordered n -tuples are defined recursively as $\langle x_1, x_2, x_3, \dots, x_n \rangle = \langle x_1, \langle x_2, x_3, \dots, x_n \rangle \rangle$. [?, 4.3, pp.76, 79-80]

Ordered pairs satisfy the property that for any sets x, y, u, v , if $\langle x, y \rangle = \langle u, v \rangle$, then $x = u$ and $y = v$. [?, Theorem 4.2, p.79]

o.2.3 Axiom Schema of Separation

Let $\phi(z, p)$ be a formula in first order logic with a free variable z . Then

$$\forall x \forall p \exists y \forall z (z \in y \iff (z \in x \wedge \phi(z, p))) . \quad (o.2.1)$$

For any sets x and p there is a unique set consisting of all z in x for which $\phi(z, p)$ holds. This is an axiom schema, meaning an infinite collection of axioms, since (o.2.1) is a separate axiom for every formula $\phi(z, p)$. [?, pp.5-6]

0.2.4 Axiom of the Empty Set

$$\exists x \forall y y \notin x$$

There is a set with no elements. We call this set $\emptyset = \{\}$. [?]

The Empty Set Axiom is not strictly required, the existence of the empty set also arises from the [Axiom Schema of Separation](#). Since we can define the empty class $\emptyset = \{u \mid u \neq u\}$, the empty set is also a set. However this follows from \emptyset being a subset of all sets and hence only under the assumption that at least one other set exists. The existence of that set, in turn, follows from the [Axiom of Infinity](#). [?, p.6]

0.2.5 Axiom of Power Sets

$$\forall x \exists y \forall z (z \in y \iff z \subseteq x)$$

For any set x there is a set, denoted by $\mathcal{P}(x)$ and called the power set of x , consisting of all subsets of x .

0.2.6 Union Axiom

$$\forall x \exists y \forall z (z \in y \iff \exists w (z \in w \wedge w \in x))$$

For any set x there is a set, denoted by $\bigcup x$, which is the union of all the elements of x .

0.2.7 Axiom of Infinity

$$\exists x (\emptyset \in x \wedge \forall y (y \in x \implies y \cup \{y\} \in x))$$

There is an inductive set.

0.2.8 Axiom of Replacement

$$\forall x \exists y \forall y' (y' \in y \iff \exists x' (x' \in x \wedge \phi(x', y'))),$$

where $\phi(s, t)$ is a formula such that

$$\forall s \exists t (\phi(s, t) \wedge \forall t' (\phi(s, t') \implies t' = t)).$$

If $\phi(s, t)$ is a class function, then when its domain is restricted to a set x , the resulting images form a set y .

0.2.9 Axiom of Foundation

$$\forall x \exists y (y \in x \wedge x \cap y = \emptyset)$$

Every set contains an \in -minimal element, we call this being *well-founded*. [?, p.92] This also means there exist no infinitely descending chains of sets, such as $x_0 \ni x_1 \ni x_2 \ni \dots$. [?, Theorem 4.3, p.95]

o.3 The Axiom of Choice