

Zorn's lemma Let (P, \leq) be a partially ordered set in which every well-ordered subset $X \subseteq P$ has an upper bound, i.e., there is $a \in P$ such that $x \leq a$ for all $x \in X$.

Then there is a maximal element in P .

Proof Suppose not. Then, if $X \subseteq P$ is well-ordered and $a \in P$ is an upper bound for X , there is some $b \in P$ s.t. $a < b$. Hence, $x < b$ for all $x \in X$.

By the axiom of choice, there is a function $c : P(P) \setminus \{\emptyset\} \rightarrow P$ that to every non-empty subset $Y \subseteq P$ picks an element $c(Y) \in Y$.

Let f be the function that to every well-ordered subset $X \subseteq P$ associates

$$f(X) := c \left(\{b \in P \mid x < b \text{ for all } x \in X\} \right).$$

So $f(X)$ is a strict upper bound for X .

A set $X \subseteq P$ is good if

- (i) X is well-ordered, and
- (ii) for all $x_0 \in X$, $x_0 = f(\{y \in X \mid y < x_0\})$.

Also, if $X \subseteq Y \subseteq P$ are subsets, we say that X is downwards closed in Y if for $x \in X$, $y \in Y$, if $y \leq x$ then $y \in X$.

Claim If $X, Y \subseteq P$ are good, then either X is a downwards closed subset of Y or Y is a downwards closed subset of X .

To see this, let W be the union of all subsets of $X \cap Y$ that are downwards closed in both X and in Y . Then W is clearly the largest subset of $X \cap Y$ that is downwards closed in both X and Y .

Also, being a subset of a well-ordered set, W is itself well-ordered.

But if $W \not\subseteq X$ and $W \not\subseteq Y$, we have $f(W) \in X$ and $f(W) \in Y$, whereby $W \cup \{f(W)\} \subseteq X \cap Y$ is a strictly larger set, downwards closed in both X and in Y . This contradicts the maximality of W , and hence either $X = W \subseteq Y$ and X is downwards closed in Y or $Y = W \subseteq X$ and Y is downwards closed in X .

By the claim it follows that if V is the union of all good subsets of P , then V is good and hence the largest good set, which contradicts that also $V \cup \{f(V)\}$ is good. This contradiction proves the theorem. \square