Formalisation of CW-complexes

Hannah Scholz

Born 21st July, 2003 in Bonn, Germany

22nd August, 2024

Bachelor's Thesis Mathematics

Advisor: Prof. Dr. Floris van Doorn

Second Advisor: Prof. Dr. Philipp Hieronymi

Mathematisches Institut

Mathematisch-Naturwissenschaftliche Fakultät der

Rheinischen Friedrich-Wilhelms-Universität Bonn

Contents

Introduction

Theorem provers are used to formally verify proofs using strict logical frameworks in digital systems. They can help ensure that every detail of a proof is indeed correct and their libraries provide correct and connected accounts of mathematical theories with complete proofs.

The programming language and proof assistant Lean is among these theorem provers. Its extensive mathematical library *mathlib*, of which the development is largely community driven, has made it, among other reasons, a popular theorem prover both for students contributing small amounts of work as mathematical side projects and also for scientists who specialise in formalisation to manage ambitious projects with many contributors. Mathlib itself could be considered one such project. This thesis aims to contribute to and build upon this enormous amount of previous work by formalising CW-complexes in Lean, a concept that is not yet part of mathlib.

Lean itself was primarily developed by Leonardo de Moura, who co-founded the Lean focused research organisation that has taken on the development for five years in 2023 [FRO24]. The latest version is called Lean 4. More about the technical details of Lean 4 can be found in [MU21].

The accompanying library mathlib is available on GitHub at `https://github.com/leanprover-community/mathlib4`. This repository has just over 300 different contributors and multiple new pull requests every day that get approved or rejected by the 28 maintainers. While mathlib is largely focused on providing a cohesive system of foundational mathematical theories, there have been a multiple large formalisations of advanced mathematical content based on mathlib, which also contributed to the library along the way.

Here are two examples: In the *Liquid Tensor Experiment*, given to the Lean community by Peter Scholze as a challenge, Johan Commelin, Adam Topaz and other contributors formalised a theorem by Peter Scholze and Dustin Clausen from condensed mathematics [Com22]. In addition, Floris van Doorn, Patrick Massot and Oliver Nash have formalised the existence of sphere eversions, a concept from differential topology, showing that geometric areas of mathematics can also be successfully formalised in Lean [DMN23].

There are also several large-scale ongoing projects of which we again present two examples: Floris van Doorn is currently leading a formalisation of a generalisation of Carleson's theorem, a theorem from fourier analysis, by Christoph Thiele and his collaborators [Bec+24]. Additionally, there is a project led by Kevin Buzzard that aims to reduce the renowned Fermat's Last Theorem to mathematical facts already known by mathematicians in the 1980s, a starting point similar to that of Andrew Wiles and Richard Taylor, who first proved this theorem in 1995 [Buz24].

As mentioned above, one important concept that is currently missing in mathlib is CW-complexes. They were first invented by Whitehead in 1949 in [Whi18] to state and prove the famous Whitehead theorem, which says that a continuous map between CW-complexes that induces isomorphisms on all homotopy groups is a homotopy equivalence. CW-complexes are especially useful when doing calculations, for example, of singular homology and cohomology. One reason is that their skeletal structure allows one to use induction. Since we are interested in providing a basic theory of CW-complexes, we will not focus on applications but instead on basic properties. An introduction to CW-complexes and their applications can be found in [LW69].

Our mathematical discussion will mostly be based on [Hat01]. In chapter 1 we will discuss CW-complexes from a purely mathematical perspective. Chapter 2 gives a short introduction to some aspects of Lean that will be useful to understand the formalisation of most of the content of chapter 1 which we will cover in chapter 3. Note that the focus of this thesis is the formalisation of CW-complexes. The accompanying code can be found at `https://github.com/scholzhannah/CWComplexes`. Throughout this thesis we will link to code either from our formalisation or from mathlib. These links will be mark with this symbol : .

1 The mathematics of CW-complexes

1.1 Definition and basic properties of a CW-complex

The modern definition of a CW-complex is the following:

Definition 1.1.1. Let X be a topological space. A *CW-complex* on X is a filtration $X_0 \subseteq X_1 \subseteq X_2 \subseteq \dots$ such that

(i) For every $n \geq 0$ there is a pushout of topological spaces

$$\begin{tikzcd} \coprod_{i \in I_n} S_i^{n-1} \arrow[r, "\coprod_{i \in I_n} q_i^n"] \arrow[d, hook, "\coprod_{i \in I_n} j_i"'] & X_{n-1} \arrow[d] \\ \coprod_{i \in I_n} D_i^n \arrow[r, "\coprod_{i \in I_n} Q_i^n"'] & X_n \end{tikzcd}$$

where I_n is any indexing set and $j_i \colon S_i^{n-1} \to D_i^n$ is the usual inclusion for every $i \in I_n$.

(ii) We have $X = \bigcup_{n \geq 0} X_n$.

(iii) X has weak topology, i.e. $A \subseteq X$ is open $\iff$ $A \cap X_n$ is open in X_n for every n.

X_n is called the *n-skeleton*. An element $e^n \in \pi_0(X_n \setminus X_{n-1})$ is called an *(open) n-cell*. Q_i^n is called a *characteristic map*.

In this thesis we will however focus on the historical definition of CW-complexes first presented by Whitehead, which can be found in [Whi18].

Definition 1.1.2. Let X be a Hausdorff space. A *CW-complex* on X consists of a family of indexing sets $(I_n)_{n \in \mathbb{N}}$ and a family of maps $(Q_i^n \colon D_i^n \to X)_{n \geq 0, i \in I_n}$ s.t.

(i) $Q_i^n|_{\mathrm{int}(D_i^n)} : \mathrm{int}(D_i^n) \to Q_i^n(\mathrm{int}(D_i^n))$ is a homeomorphism. We call $e_i^n := Q_i^n(\mathrm{int}(D_i^n))$ an *(open) n-cell* (or a cell of dimension n) and $\overline{e}_i^n := Q_i^n(D_i^n)$ a *closed n-cell* .

(ii) For all $n, m \in \mathbb{N}$, $i \in I_n$ and $j \in I_m$ where $(n, i) \neq (m, j)$ the cells e_i^n and e_j^m are disjoint.

(iii) For each $n \in \mathbb{N}$, $i \in I_n$, $Q_i^n(\partial D_i^n)$ is contained in the union of a finite number of closed cells of dimension less than n.

(iv) $A \subseteq X$ is closed iff $Q_i^n(D_i^n) \cap A$ is closed for all $n \in \mathbb{N}$ and $i \in I_n$.

(v) $\bigcup_{n\geq 0}\bigcup_{i\in I_n} Q_i^n(D_i^n) = X$.

We call Q_i^n a *characteristic map* and $\partial e_i^n := Q_i^n(\partial D_i^n)$ the *frontier of the n-cell* for any i and n . Additionally we define $X_n := \bigcup_{m<n+1}\bigcup_{i\in I_m} \overline{e}_i^m$ and call it the *n-skeleton* of X for $-1 \leq n \leq \infty$.

For the rest of the section let X be a CW-complex.

Remark 1.1.3. Property (iii) in the above definition is called *closure finiteness*. Property (iv) is called *weak topology*. Whitehead named CW-complexes *closure finite complexes with weak topology* after these two properties [Whi18].

These two different notions are equivalent:

Proposition 1.1.4. *Definition 1.1.1 and 1.1.2 are equivalent.*

The proof to this proposition is long, tedious and not relevant to this thesis, so we will skip it here. It can be found as the proof of Proposition A.2. in [Hat01]. From here on, the term CW-Complex will always refer to the older Definition 1.1.2. As such, keep in mind that throughout this thesis any CW-complex will, by definition, be assumed to be Hausdorff.

Remark 1.1.5. The name *open n-cell* and the notation ∂e_i^n can be confusing as an open n-cell is not necessarily open and ∂e_i^n is not necessarily the boundary of $\overline{e}_i^n$.

But at least the notion of a closed n-cell makes sense:

Lemma 1.1.6. *$\overline{e}_i^n$ is compact and closed for every $n \in \mathbb{N}$ and $i \in I_n$. Similarly ∂e_i^n is compact and closed for every $n \in \mathbb{N}$ and $i \in I_n$.*

Proof. D_i^n is compact. Therefore its image $Q_i^n(D_i^n) = \overline{e}_i^n$ is compact as well. In a Hausdorff space any compact set is closed. Thus $\overline{e}_i^n$ is closed. The proof for ∂e_i^n works in the same way. □

And the following is also true:

Lemma 1.1.7. *$\overline{e_i^n} = \overline{e}_i^n$ for every $n \in \mathbb{N}$ and $i \in I_n$.*

Proof. Since $e_i^n \subseteq \overline{e}_i^n$ and $\overline{e}_i^n$ is closed by the lemma above, the left inclusion is trivial. So let us show now that $\overline{e}_i^n \subseteq \overline{e_i^n}$. This statement can be rewritten as $Q_i^n\left(\overline{D_i^n}\right) \subseteq \overline{Q_i^n(D_i^n)}$. It is generally true for any continuous map that the closure of the image is contained in the image of the closure. Thus we are done. □

Now let us define what it means for a CW-complex to be finite:

Definition 1.1.8. Let X be a CW-complex. We call X *of finite type* if there are only finitely many cells in each dimension, i.e. if I_n is finite for all $n \in \mathbb{N}$. X is said to be *finite dimensional* if there is an $n \in \mathbb{N}$ such that $X = X_n$. Finally, X is called *finite* if it is of finite type and finite dimensional.

If we already know that the CW-complex we want to construct will be finite or of finite type, we can relax some of the conditions:

Remark 1.1.9.

(i) For a CW-complex of finite type, condition (iii) in Definition 1.1.2 follows from the following: For each $n \in \mathbb{N}$, $i \in I_n$ $Q_i^n(\partial D_i^n)$ is contained in $\bigcup_{m \leq n-1} \bigcup_{i \in I_m} e_i^m$.

(ii) Additionally, for a finite CW-complex, condition (iv) in Definition 1.1.2 follows from the other conditions.

Proof. Let us begin with statement (i). Take $n \in \mathbb{N}$ and $i \in I_n$. We need to show that $Q_i^n(\partial D_i^n)$ is contained in a finite number of cells of a lower dimension. But by assumption, we have $Q_i^n(\partial D_i^n) \subseteq \bigcup_{m \leq n-1} \bigcup_{i \in I_m} e_i^m$ which in this case consists of finitely many cells. Now we can move on to statement (ii). We need to prove condition (iv) of Definition 1.1.2, i.e.

$$A \subseteq X \text{ is closed } \iff \overline{e}_i^n \cap A \text{ is closed for all } n \in \mathbb{N} \text{ and } i \in I_n.$$

For the forward direction, notice that $\overline{e}_i^n \cap A$ is just the intersection of two closed sets by assumption and Lemma 1.1.6. As such it is closed. For the backward direction, take an $A \subseteq X$ such that $\overline{e}_i^n$ is closed for all $n \in \mathbb{N}$ and $i \in I_n$. We need to show that A is closed. But using condition (v) of Definition 1.1.2 we get

$$A = A \cap \bigcup_{n \geq 0} \bigcup_{i \in I_n} \overline{e}_i^n = \bigcup_{n \geq 0} \bigcup_{i \in I_n} (A \cap \overline{e}_i^n)$$

which by assumption is a finite union of closed sets, making A closed. □

We can also think about the n-skeletons as being made up of open cells:

Lemma 1.1.10. *$X_n = \bigcup_{m<n+1} \bigcup_{i \in I_m} e_i^m$ for every $-1 \leq n \leq \infty$.*

Proof. We show this by induction over $-1 \leq n \leq \infty$. For the base case, assume that $n = -1$. Then we get $X_{-1} = \bigcup_{m<0} \bigcup_{i \in I_m} \overline{e}_i^m = \varnothing = \bigcup_{m<0} \bigcup_{i \in I_m} e_i^m$.

For the induction step, assume that that the statement is true for n. We now show that it also holds for $n+1$.

$$\begin{aligned}
X_{n+1} &= \bigcup_{m<n+2} \bigcup_{i \in I_m} \overline{e}_i^m \\
&= \bigcup_{i \in I_{n+1}} \overline{e}_i^{n+1} \cup \bigcup_{m<n+1} \bigcup_{i \in I_m} \overline{e}_i^m \\
&= \bigcup_{i \in I_{n+1}} \overline{e}_i^{n+1} \cup X_n \\
&\overset{(1)}{=} \bigcup_{i \in I_{n+1}} \overline{e}_i^{n+1} \cup \bigcup_{m<n+1} \bigcup_{i \in I_m} e_i^m \\
&= \bigcup_{i \in I_{n+1}} e_i^{n+1} \cup \bigcup_{i \in I_{n+1}} \partial e_i^{n+1} \cup \bigcup_{m<n+1} \bigcup_{i \in I_m} e_i^m \\
&\overset{(2)}{=} \bigcup_{i \in I_{n+1}} e_i^{n+1} \cup \bigcup_{m<n+1} \bigcup_{i \in I_m} e_i^m \\
&= \bigcup_{m<n+2} \bigcup_{i \in I_m} e_i^m
\end{aligned}$$

Note that (1) holds by induction and (2) holds by closure finiteness (property (iii) in Definition 1.1.2).

Now we can move on to the case $n = \infty$.

$$\begin{aligned} X_\infty &= \bigcup_{m<\infty+1} \bigcup_{i\in I_m} \overline{e}_i^m \\ &= \bigcup_{m<\infty+1} \bigcup_{l<m+1} \bigcup_{i\in I_l} \overline{e}_i^l \\ &= \bigcup_{m<\infty+1} X_m \\ &\overset{(1)}{=} \bigcup_{m<\infty+1} \bigcup_{l<m+1} \bigcup_{i\in I_l} e_i^l \\ &= \bigcup_{m<\infty+1} \bigcup_{i\in I_m} e_i^m \end{aligned}$$

Where (1) holds by induction. $\square$

This also enables us to write X as a union of open cells:

Corollary 1.1.11. $X = \bigcup_{n\geq 0} \bigcup_{i\in I_n} e_i^n$.

When we want to show that a set $A \subseteq X$ is closed, the weak topology (property (iv) in 1.1.2) lets us reduce that question to an individual cell. It is then often convenient to perform strong induction over the dimension of that cell. To this end, we want to prove a lemma that makes this repeated process easier. We first need the following:

Lemma 1.1.12. *Let $A \subseteq X$ be a set and n a natural number. Assume that for every $m \leq n$ and $j \in I_m$ the intersection $A \cap \overline{e}_j^m$ is closed. Then $A \cap \partial e_j^{n+1}$ is closed for every $j \in I_{n+1}$.*

Proof. By closure finiteness of X (property (iii) in 1.1.2), there is a set E of cells of dimension lower than $n+1$ such that $\partial e_j^{n+1} \subseteq \bigcup_{e\in E} \overline{e}$. This gives us

$$A \cap \partial e_j^{n+1} = A \cap \bigcup_{e\in E} \overline{e} \cap \partial e_j^{n+1} = \bigcup_{e\in E} (A \cap \overline{e}) \cap \partial e_j^{n+1}.$$

$\bigcup_{e\in E}(A \cap \overline{e})$ is closed as a finite union of sets that are closed by assumption and ∂e_j^{n+1} is closed by Lemma 1.1.6. Therefore, the intersection is also closed. $\square$

Now we can move on to the lemma that we actually want. We can think of this lemma as being a weaker condition than the weak topology i.e. property (iv) in 1.1.2.

Lemma 1.1.13. *Let $A \subseteq X$ be a set such that for every $n > 0$ and $j \in I_n$ either $A \cap e_j^n$ or $A \cap \overline{e}_j^n$ is closed. Then A is closed.*

Proof. Since X has weak topology, it is enough to show that $A \cap \overline{e}_j^n$ is closed for every $n \in \mathbb{N}$ and $i \in I_n$. We show this by strong induction over n. For the base case $n = 0$, notice that $\overline{e}_j^0$ is a singleton, and the intersection with a singleton is either that singleton or empty. As such, the intersection is closed in both cases.

For the induction step, assume that for every $m \leq n$ the statement already holds. We now need to show it for $n+1$. By assumption either $A \cap e_j^{n+1}$ or $A \cap \overline{e}_j^{n+1}$ is closed. The second case is just what we wanted to show.

In the first case, we can use that $A \cap \overline{e}_j^{n+1} = (A \cap \partial e_j^{n+1}) \cup (A \cap e_j^{n+1})$. The left part of the union is closed by Lemma 1.1.12 applied to the induction hypothesis. The right part of the union is closed by the assumption of our case. The union is therefore also closed. ☐

We can use the lemma we just proved to show that the n-skeletons are closed:

Lemma 1.1.14. *X_n is closed for every $n \in \mathbb{N}$.*

Proof. By the previous Lemma 1.1.13, it is enough to show that for every $m \in \mathbb{N}$ and $j \in I_m$ either $X_n \cap e_j^m$ or $X_n \cap \overline{e}_j^m$ is closed. We differentiate two cases based on whether $m < n+1$. First assume that $m < n+1$ holds. Then by the definition of X_n we get $X_n \cap \overline{e}_j^m = (\bigcup_{m<n+1} \bigcup_{i \in I_m} \overline{e}_i^m) \cap \overline{e}_j^m = \overline{e}_j^m$ which is closed by Lemma 1.1.6. Now assume that it does not hold. Then by Lemma 1.1.10 we get $X_n \cap e_j^m = (\bigcup_{m<n+1} \bigcup_{i \in I_m} e_i^m) \cap e_j^m = \varnothing$ where the last equality holds because the open cells are pairwise disjoint by property (ii) in Definition 1.1.2. The empty set is trivially closed. ☐

Another fact that can be quite helpful is a version of closure finiteness using open cells:

Lemma 1.1.15. *For each $n \in \mathbb{N}$ and $i \in I_n$, ∂e_i^n is contained in the union of a finite number of open cells of dimension less than n.*

Proof. We show this by strong induction on n. For the base case $n = 0$ notice that ∂e_i^0 is empty.

For the induction step, assume that the statement holds for all $m \leq n$. We need to show that it also holds for $n+1$. By closure finiteness there is a finite set E of cells of dimension less than $n+1$ such that $\partial e_i^{n+1} \subseteq \bigcup_{e \in E} \overline{e}$. If we can show that for every $e \in E$ there is a finite set E_e of cells of dimension less than $n+1$ such that $\overline{e} \subseteq \bigcup_{e' \in E_e} e'$, we would then be done since $\partial e_i^{n+1} \subseteq \bigcup_{e \in E} \overline{e} \subseteq \bigcup_{e \in E} \bigcup_{e' \in E_e} e'$.

So take $e \in E$. By the induction hypothesis there is a finite set E'_e of cells of a lesser dimension than that of e such that $\partial e \subseteq \bigcup_{e' \in E'_e} e'$. This gives us $\overline{e} = \partial e \cup e \subseteq (\bigcup_{e' \in E'_e} e') \cup e$, which finishes the proof. ☐

Let us now look at some more ways to show that sets in X are closed.

Lemma 1.1.16. *$A \subseteq X$ is closed iff $A \cap X_n$ is closed for every $n \in \mathbb{N}$.*

Proof. The forward direction follows directly from Lemma 1.1.14. For the backward direction, take $A \subseteq X$ such that $A \cap X_n$ is closed for every n. Since X has weak topology we need to show that $A \cap \overline{e}_i^n$ is closed for every $n \in \mathbb{N}$ and $i \in I_n$. But $A \cap \overline{e}_i^n = A \cap X_n \cap \overline{e}_i^n$ which is closed by assumption and Lemma 1.1.6. ☐

When we use this lemma with induction, we might want the following for the induction step:

Lemma 1.1.17. *Let $A \subseteq X$. $A \cap X_{n+1}$ is closed iff $A \cap X_n$ and $A \cap \overline{e}_j^{n+1}$ are closed for every $j \in I_{n+1}$.*

Proof. For the forward direction notice that $A \cap X_n = A \cap X_{n+1} \cap X_n$, which is closed by assumption and Lemma 1.1.14, and $A \cap \overline{e}_j^{n+1} = A \cap X_{n+1} \cap \overline{e}_j^{n+1}$, which is closed by assumption and Lemma 1.1.6. For the backwards direction we apply Lemma 1.1.13. We now need to show that for every $m \in \mathbb{N}$ and $j \in I_m$ either $A \cap e_j^m$ or $A \cap \overline{e}_j^m$ is closed. We differentiate three different cases. First let us look at the case $m \leq n$. Then

$$A \cap X_{n+1} \cap \overline{e}_j^m = A \cap \overline{e}_j^m = A \cap X_n \cap \overline{e}_j^m$$

which is closed by assumption and Lemma 1.1.6. Now we consider $m = n + 1$. Then $A \cap X_{n+1} \cap \overline{e}_j^{n+1} = A \cap \overline{e}_j^{n+1}$, which is closed by assumption. Lastly, we show the claim for $m > n + 1$. Here we get $A \cap X_{n+1} \cap e_j^m = A \cap (\bigcup_{l<n+1} \bigcup_{i \in I_l} e_j^l) \cap e_j^m = \varnothing$, where we used Lemma 1.1.10 and the fact that different open cells are disjoint (property (ii) in Definition 1.1.2). The empty set is obviously closed. □

With that we can write a new strong induction principle for showing that sets in a CW-complex are closed:

Lemma 1.1.18. *Let $A \subseteq X$ be a set such that for all $n \in \mathbb{N}$, if for all $m \leq n$ the intersection $A \cap X_m$ is closed, then for all $j \in I_{n+1}$ the intersection $A \cap \overline{e}_j^{n+1}$ is closed. Then A is closed.*

Proof. By Lemma 1.1.16 it is enough to show that for all $n \in \mathbb{N}$ the set $A \cap X_n$ is closed. We do strong induction over n starting at -1. For the base case notice that $X_{-1} = \varnothing$. Now for the induction step, assume that $A \cap X_m$ is closed for all $m \leq n$. We need to show that $A \cap X_{n+1}$ is closed as well. By the previous lemma, it is enough to show that $A \cap X_n$ and $A \cap \overline{e}_j^{n+1}$ are closed for all $j \in I_{n+1}$. But the first one is closed by induction hypothesis and the second one is closed by our assumption applied to the induction hypothesis. □

We can now use all these new techniques to show some important properties of CW-complexes:

Lemma 1.1.19. *X_0 is discrete.*

Proof. We want to show that every set $A \subseteq X_0$ is closed in X_0. It is enough if A is closed in X. We apply Lemma 1.1.13. Take $n > 0$ and $i \in I_n$. We show that $A \cap e_i^n$ is closed. But using Lemma 1.1.10 and that different open cells are disjoint, we have $A \cap e_i^n = A \cap X_0 \cap e_i^n = A \cap (\bigcup_{m<1} \bigcup_{j \in I_m} e_j^m) \cap e_i^n = \varnothing$, which is closed. □

The proof of the following lemma is based on the proof of Proposition A.1. in [Hat01].

Lemma 1.1.20. *For every compact set $C \subseteq X$ the set of all open cells e_i^n such that $e_i^n \cap C \neq \varnothing$ is finite.*

Proof. Assume towards a contradiction that the set $S := \{n \in \mathbb{N}, i \in I_n \mid e_i^n \cap C \neq \varnothing\}$ is infinite. For every pair $(n, i) \in S$ pick a point $p_{n,i} \in e_i^n \cap C$. Since the open cells are pairwise disjoint, we know that the set $P := \{p_{n,i} \mid (n, i) \in S\}$ is also infinite. We will now show that P is discrete and compact. Then P must be finite, which is a contradiction. For both compactness and discreteness we will need that every set $A \subseteq P$ is closed in X.

So let $A \subseteq P$. We apply Lemma 1.1.18. Assuming that for all $m \leq n$ the intersection $A \cap X_m$ is closed, we need to show that $A \cap \overline{e}_j^{n+1}$ is closed for every $j \in I_{n+1}$. Since

$A \cap \overline{e}_j^{n+1} = (A \cap \partial e_j^{n+1}) \cup (A \cap e_j^{n+1})$ and $A \cap \partial e_j^{n+1} = A \cap X_n \cap \partial e_j^{n+1}$ is closed by Lemma 1.1.6 and the assumption, it is enough to show that $A \cap e_j^{n+1}$ is closed. If the intersection $A \cap e_j^{n+1}$ is empty, then we are done. So assume that there is an $x \in A \cap e_j^{n+1}$. Since $x \in A \subseteq P$ there is $(m, i) \in S$ such that $p_{m,i} = x$. But the open cells of X are pairwise disjoint, so it must be that $(m, i) = (n + 1, j)$ and therefore $p_{n+1,j} = x$. Thus $A \cap e_j^{n+1} = \{p_{n+1,j}\}$, which is closed since every singleton in a Hausdorff space is closed.

This directly gives us that the subspace topology on P is discrete. For compactness, notice that, by what we just did, P is closed and as a closed subset of the compact set C it is also compact. This is a contradiction to the fact that P is infinite as explained above. □

This lemma helps us prove the following characterisation of finite CW-complexes:

Lemma 1.1.21. *X is a finite CW-complex iff X compact.*

Proof. For the forward direction we know that $X = \bigcup_{n\in\mathbb{N}} \bigcup_{i\in I_n} \overline{e}_i^n$, which, by assumption and Lemma 1.1.6, is compact as a finite union of compact sets.

The backward direction follows from Lemma 1.1.20 and Corollary 1.1.11. □

1.2 Constructions

In this section we will discuss how to get new CW-complexes from existing ones. We can start with some easy ones.

1.2.1 Skeletons as CW-complexes

The n-skeletons of a CW-complex X are again CW-complexes:

Lemma 1.2.1. *Let $-1 \leq n \leq \infty$. Then X_n is a CW-complex together with the cells $J_m \coloneqq I_m$ for $m < n + 1$ and $J_m = \varnothing$ otherwise.*

Proof. We need to verify the five conditions of Definition 1.1.2. Conditions (i), (ii) and (iii) follow directly from X fulfilling these conditions and condition (v) is given by the definition of the n-skeleton. Thus we only need to worry about condition (iv), i.e. that X_n has weak topology. It follows easily from Lemma 1.1.6 that for a set $A \subseteq X_n$ that is closed in X_n the intersection $A \cap \overline{e}_i^m$ is closed in X_n for every $m \in \mathbb{N}$ and $i \in J_m$. We can therefore directly consider the other direction. Let $A \subseteq X_n$ be a set such that for every $m \in \mathbb{N}$ and $i \in J_m$ the intersection $A \cap \overline{e}_i^m$ is closed in X_n. We need to show that A is closed in X_n. It suffices to show that A is closed in X. By Lemma 1.1.13 we need to prove that for every $m \in \mathbb{N}$ and $i \in I_m$ either $A \cap \overline{e}_i^m$ or $A \cap e_i^m$ is closed. Let us start with the case $i \in J_m$. By assumption, $A \cap \overline{e}_i^m$ is closed in X_n. The definition of the subspace topology tells us that there exists a closed set $C \subseteq X$ such that $C \cap X_n = A \cap \overline{e}_i^m$. But since X_n is closed by Lemma 1.1.14, that means that $A \cap \overline{e}_i^m$ is also closed in X. So we are done for this case. For the case $i \notin J_m$ notice that, by Lemma 1.1.10, we get $A \cap e_i^m \subseteq X_n \cap e_i^m = (\bigcup_{l<n+1} \bigcup_{j\in I_l} e_j^l) \cap e_i^m = \varnothing$ since different open cells of X are disjoint. The empty set is obviously closed. □

1.2.2 Disjoint union of CW-complexes

Additionally, we can get a CW-complex by taking the disjoint union of two CW-complexes:

Lemma 1.2.2. *Let X and Y be two CW-complexes with indexing sets $(I_{1,n})_{n\in\mathbb{N}}$ and $(I_{2,n})_{n\in\mathbb{N}}$. Then $X \amalg Y$ is a CW-complex with indexing sets $J_n \coloneqq I_{1,n} \cup I_{2,n}$.*

Proof. We need to show that this construction satisfies the conditions of Definition 1.1.2. Conditions (i), (ii), (iii) and (v) follow directly from X and Y fulfilling these conditions. So we again only need to focus on condition (iv), i.e. the weak topology. The forward direction follows in the same way as in a lot of the other proofs. For the backwards direction take $A \subseteq X \amalg Y$ such that $A \cap \overline{e}_i^n$ is closed in $X \amalg Y$ for every $n \in \mathbb{N}$ and $i \in J_n$. We need to show that A is closed in $X \amalg Y$. By the definition of the disjoint union topology, this is equivalent to $A \cap X$ being closed in X and $A \cap Y$ being closed in Y. We will show this for X. By the weak topology, it is enough to show that $A \cap X \cap \overline{e}_i^n$ is closed in X for every $n \in \mathbb{N}$ and $i \in I_{1,n}$. But we have $A \cap X \cap \overline{e}_i^n = (A \cap \overline{e}_i^n) \cap X$, which is closed in X by assumption and the definition of the disjoint union topology. □

1.2.3 Image of a homeomorphism

Homeomorphisms respect the CW-complex structure:

Lemma 1.2.3. *Let X and Y be topological spaces and $f\colon X \to Y$ a homeomorphism. If X is a CW-complex with indexing sets $(I_n)_{n\in\mathbb{N}}$ and characteristic maps $(Q_i^n)_{n\in\mathbb{N},i\in I_n}$, then Y is a CW-complex with the same indexing sets and characteristic maps $(f \circ Q_i^n)_{n\in\mathbb{N},i\in I_n}$.*

Proof. Properties (ii), (iii) and (v) of Definition 1.1.2 follow easily from the fact that f is a bijection. Property (i) holds since we compose the characteristic maps with a homeomorphism. Let us lastly look at property (iv), i.e. the weak topology. We get:

$$\begin{aligned} A \subseteq Y \text{ is closed} &\iff f^{-1}(A) \subseteq X \text{is closed} \\ &\iff Q_i^n(D_i^n) \cap f^{-1}(A) = f^{-1}((f \circ Q_i^n)(D_i^n) \cap A) \subseteq X \text{ is closed} \\ &\iff (f \circ Q_i^n)(D_i^n) \cap A \subseteq Y \text{ is closed.} \end{aligned}$$

□

1.2.4 Subcomplexes

One important way to get a new CW-complex from an existing one is to consider subcomplexes, which we will discuss in this section.

Let X be a CW-complex. A subcomplex of X is defined as follows:

Definition 1.2.4. A subcomplex of X is a set $E \subseteq X$ together with a set $J_n \subseteq I_n$ for every $n \in \mathbb{N}$ such that:

(i) E is closed.

(ii) $\bigcup_{n\in\mathbb{N}} \bigcup_{i\in J_n} e_i^n = E$.

Note that here we want E to be the union of the open cells instead of the union of the closed cells as in Definition 1.1.2. But we can prove the other version easily:

Lemma 1.2.5. *Let $E \subseteq X$ be a subcomplex. Then $\bigcup_{n\in\mathbb{N}} \bigcup_{i\in J_n} \overline{e}_i^n = E$.*

Proof. Let $n \in \mathbb{N}$ and $i \in J_n$. It is enough to show that $\overline{e}_i^n \subseteq E$. By Lemma 1.1.7 $\overline{e}_i^n = \overline{e_i^n}$. Since E is closed by property (i) $\overline{e}_i^n \subseteq E$ is equivalent to $e_i^n \subseteq E$ which is true by property (ii). ☐

Example 1.2.6. We have already proven that every n-skeleton is a subcomplex with Lemma 1.1.14 and Lemma 1.1.10. This section therefore provides us with an alternative way to show that n-skeletons are CW-complexes.

Here are some alternative ways to define subcomplexes. These are taken from chapter 7.4 in [Jän01]. The proof that these three notions are equivalent can be found in there. We will just show the implication that is useful to us.

Lemma 1.2.7. *Let $E \subseteq X$ and $J_n \subseteq I_n$ for $n \in \mathbb{N}$ be such that*

(i) For every $n \in \mathbb{N}$ and $i \in I_n$ we have $\overline{e}_i^n \subseteq E$.

(ii) $\bigcup_{n\in\mathbb{N}} \bigcup_{i\in J_n} e_i^n = E$.

Then E is a subcomplex of X.

Proof. Property (ii) in Definition 1.2.4 is clear immediately. So we only need to show that E is closed. We apply Lemma 1.1.13 which means we only need to show that for every $n \in \mathbb{N}$ and $i \in I_n$ either $E \cap \overline{e}_i^n$ or $E \cap e_i^n$ is closed. So let $n \in \mathbb{N}$ and $i \in I_n$. We differentiate the cases $i \in J_n$ and $i \notin J_n$. For the first one notice that by property (i) E can be expressed as a union of closed cells: $E = \bigcup_{m\in\mathbb{N}} \bigcup_{j\in J_n} e_j^m \subseteq \bigcup_{m\in\mathbb{N}} \bigcup_{j\in J_n} \overline{e}_j^m \subseteq E$. This gives us $E \cap \overline{e}_i^n = \overline{e}_i^n$, which is closed by Lemma 1.1.6. Now for the case $i \notin J_n$, the disjointness of the open cells of X gives us that $E \cap e_i^n = (\bigcup_{m\in\mathbb{N}} \bigcup_{j\in J_n} e_j^m) \cap e_i^n = \varnothing$, which is obviously closed. ☐

And here is a third way to express the property of being a subcomplex:

Lemma 1.2.8. *Let $E \subseteq X$ and $J_n \subseteq I_n$ for $n \in \mathbb{N}$ be such that*

(i) E is a CW-complex with respect to the cells determined by X and J_n.

(ii) $\bigcup_{n\in\mathbb{N}} \bigcup_{i\in J_n} e_i^n = E$.

Then E is a subcomplex of X.

Proof. We will show that this satisfies the properties of the construction above in Lemma 1.2.7. Property (ii) is again immediate. Property (i) combined with Definition 1.1.2 of a CW-complex immediately gives us property (i) of Lemma 1.2.7. ☐

Now we can show that a subcomplex is indeed again a CW-complex:

Lemma 1.2.9. *Let $E \subseteq X$ together with $J_n \subseteq I_n$ for every $n \in \mathbb{N}$ be a subcomplex of the CW-complex X. Then E is again a CW-complex with respect to the cells determined by J_n and X.*

Proof. We show this by verifying the properties in the Definition 1.1.2 of a CW-complex. Properties (i) and (ii) are immediate and we already covered property (v) in Lemma 1.2.5.

Let us consider property (iii) i.e. closure finiteness. So let $n \in \mathbb{N}$ and $i \in J_n$. By closure finiteness of X we know that there is a finite set $E \subseteq \bigcup_{m<n} I_n$ such that $\partial e_i^n \subseteq \bigcup_{e \in E} e$. We define $E' \coloneqq \{e_j^m \in E \mid j \in J_m\}$. We want to show that $\partial e_i^n \subseteq \bigcup_{e \in E'} e$. Take $x \in \partial e_i^n$. Since $\partial e_i^n \subseteq \bigcup_{e \in E} e$, there is an $e_j^m \in E$ such that $x \in e_j^m$. It is obviously enough to show that $j \in J_m$. By Lemma 1.2.5 we know that $x \in \partial e_i^n \subseteq \overline{e}_i^n \subseteq E$. But since $E = \bigcup_{m' \in \mathbb{N}} \bigcup_{j' \in J_{m'}} e_{j'}^{m'}$ there is $m' \in \mathbb{N}$ and $j' \in J_{m'}$ such that $x \in e_{j'}^{m'}$. We know that the open cells of X are disjoint which gives us $(m, j) = (m', j')$. That directly implies $j \in J_m$, which we wanted to show.

Lastly we need to show property (iv), i.e. that E has weak topology. Like in a lot of our other proofs, $A \subseteq E$ being closed implies that $A \cap \overline{e}_i^n$ is closed for every $n \in \mathbb{N}$ and $i \in J_n$. So now take $A \subseteq E$ such that $A \cap \overline{e}_i^n$ is closed in E for every $n \in \mathbb{N}$ and $i \in J_n$. We need to show that A is closed in E. It is enough to show that A is closed in X. We apply Lemma 1.1.13 which means we only need to show that for every $n \in \mathbb{N}$ and $j \in I_n$ either $A \cap \overline{e}_j^n$ or $A \cap e_j^n$ is closed. We look at two cases. Firstly consider $j \in J_n$. Then $A \cap \overline{e}_i^n$ is closed in E by assumption. By the definition of the subspace topology this means that there exists a closed set $B \subseteq X$ such that $A \cap \overline{e}_i^n = E \cap B$. But since E is closed by assumption (i) of Definition 1.2.4 of a subcomplex, that means that $A \cap \overline{e}_i^n$ is the intersection of two closed sets in X making it also closed. Now let us cover the case $j \notin J_n$. This gives us $A \cap e_j^n \subseteq E \cap e_j^n = (\bigcup_{m \in \mathbb{N}} \bigcup_{i \in J_m} e_i^m) \cap e_j^n = \varnothing$, where the last equality holds since the open cells of X are pairwise disjoint. Thus $A \cap e_j^n = \varnothing$, which is obviously closed. $\square$

Now let us look at some properties of subcomplexes:

Lemma 1.2.10. *A union of subcomplexes $(E_i)_{i \in \iota}$ of X with indexing sets $(I_{i,n})_{i \in \iota, n \in \mathbb{N}}$ is again a subcomplex of X with the indexing set $\bigcup_{i \in \iota} I_{i,n}$ for every $n \in \mathbb{N}$.*

Proof. We show that this construction satisfies the assumptions of Lemma 1.2.7. Property (ii) follows easily from that the fact that each of the subcomplexes E_i is the union of its open cells. So let us look at property (i). Take $n \in \mathbb{N}$ and $j \in \bigcup_{i \in \iota} I_{i,n}$. Then there is a $i \in \iota$ such that $j \in I_{i,n}$. With Lemma 1.2.5 we get $\overline{e}_j^n \subseteq \bigcup_{n \in \mathbb{N}} \bigcup_{j \in I_{i,n}} \overline{e}_j^n = E_i \subseteq \bigcup_{i \in \iota} E_i$, which means we are done. $\square$

Remark 1.2.11. We say a subcomplex is finite, when it is finite as a CW-complex. It is easy to see that taking a finite union of finite subcomplexes of X yields again a finite subcomplex of X.

Here are two examples of finite subcomplexes that we will need:

Example 1.2.12.

(i) Let $i \in I_0$. Then $\overline{e}_i^0$ is a finite subcomplex of X with the indexing sets $J_0 = \{i\}$ and $J_n = \varnothing$ for $n > 0$.

(ii) Let E together with the indexing sets $(J_n)_{n \in \mathbb{N}}$ be a finite subcomplex of X and $n \in \mathbb{N}$ and $i \in I_n$ such that ∂e_i^n is included in a union of cells of E of dimension less than n. Then $E \cup e_i^n$ together with $J_n' = J_n \cup \{i\}$ and $J_m' = J_m$ for $m \neq n$ is a finite subcomplex of X.

We will omit the proofs of these examples as they are quite direct to see.
This helps us get the following lemma:

Lemma 1.2.13. *Let $n \in \mathbb{N}$ and $i \in I_n$. Then there is a finite subcomplex of X such that i is among its cells.*

Proof. We show this by strong induction over n. The base case $n = 0$ is directly given by the first example in 1.2.12. For the induction step assume that the statement is true for all $m \leq n$. We now need to show that it then also holds for $n + 1$. By closure finiteness of X there is a finite set F of cells of X with dimension less than $n+1$ such that $\partial e_i^{n+1} \subseteq \bigcup_{e \in F} \overline{e}$. By induction each cell $e \in F$ is part of a finite subcomplex E_e of X. By Lemma 1.2.10 and Remark 1.2.11, $\bigcup_{e \in F} E_e$ is again a finite subcomplex of X. The second example in Example 1.2.12 now allows us to attach the cell e_i^{n+1} to this subcomplex yielding a finite subcomplex with e_i^{n+1} among its cells. □

Corollary 1.2.14. *Every finite set of cells of X is contained in a finite subcomplex of X.*

Proof. Let F be the set of finite cells. By the above Lemma 1.2.13 each cell $e \in F$ is contained in a finite subcomplex E_e. By Lemma 1.2.10 and Remark 1.2.11 $\bigcup_{e \in F} E_e$ is again a finite subcomplex of X and we obviously have $\bigcup_{e \in F} e \subseteq \bigcup_{e \in F} E_e$. □

Corollary 1.2.15. *Let $C \subseteq X$ be compact. Then C is contained in a finite subcomplex of X.*

Proof. We know from Lemma 1.1.20 and property (v) in Definition 1.1.2 that C is contained in a finite union of cells of X. And now the above corollary tells us that these finite cells, and therefore C, is contained in a finite subcomplex of X. □

1.2.5 Product of CW-complexes

In this subsection we will talk about the product of CW-complexes.

Counterexample

We will first show that the product of two CW-complexes is not necessarily a CW-complex with respect to the natural cell structure.

Remark 1.2.16. The statement that we would want but is unfortunately false is the following:

Let X, Y be CW-complexes with families of characteristic maps $(Q_i^n \colon D_i^n \to X)_{n \in \mathbb{N}, i \in I_n}$ and $(P_j^m \colon D_j^m \to Y)_{m \in \mathbb{N}, j \in J_n}$. Then we would want to get a CW-structure on $X \times Y$ with characteristic maps $(Q_i^n \times P_j^m \colon D_i^n \times D_j^m \to X \times Y)_{n,m \in \mathbb{N}, i \in I_n, j \in J_m}$. The indexing sets K_l are given by $K_l = \bigcup_{n+m=l} I_n \times J_m$ for every $l \in \mathbb{N}$.

We will discuss a counterexample first presented by Dowker in [Dow52].
We firstly define the two relevant spaces:

Definition 1.2.17. Let $X = \bigvee_{i \in \iota} A_i$ where A_i is the unit interval for every $i \in \iota$ and ι is the set of all infinite sequences in $\mathbb{N}$. X has a 0-cell at the base point of the wedge sum, which we will label 0_X and assume to be the 0 of all of the intervals. The rest of the 0-cells are the 1's of the intervals. The 1-cells are the interiors of the intervals.

Lemma 1.2.18. *X together with the described cell-structure is a CW-complex.*

Proof. Firstly note that the wedge sum is defined to be $\bigvee_{i \in \iota} A_i := \coprod_{i \in \iota} A_i / \sim$ where $\sim$ is the equivalence relation identifying all the 0's of the intervals. It is easy to see from the definition that the wedge sum of Hausdorff spaces is again a Hausdorff space. We now need to verify the five conditions of Definition 1.1.2. They are all relatively self-evident except for condition (iv), which says that X needs to have weak topology. The forward direction follows in the same way as always. For the backward direction take a set $C \subseteq X$ such that $C \cap \overline{e}_i^n$ is closed for all the closed cells of X. Note that the only relevant information this gives us is that $C \cap A_i$ is closed in X for every $i \in \iota$. We need to show that C is closed in X. By the quotient topology, C is closed in X if its preimage $q^{-1}(C)$ under the quotient map $q\colon \coprod_{i \in \iota} A_i \to \coprod_{i \in \iota} A_i / \sim$ is closed in the disjoint union. But by the disjoint union topology $q^{-1}(C)$ is closed iff $q^{-1}(C) \cap A_i = C \cap A_i$ is closed in A_i for every $i \in \iota$, which is true by assumption. □

Definition 1.2.19. Let $Y = \bigvee_{j \in \mathbb{N}} B_k$, where B_k is the unit interval for every $j \in \mathbb{N}$. X has a 0-cell at the base point of the wedge sum, which we will label 0_Y and assume to be the 0 of all of the intervals. The rest of the 0-cells are the 1's of the intervals. The 1-cells are the interiors of the intervals.

Lemma 1.2.20. *Y together with the described cell-structure is a CW-complex.*

Proof. Completely analogous to the proof of Lemma 1.2.18. □

Lemma 1.2.21. *The space $X \times Y$ is not a CW-complex with respect to the cell-structure proposed in 1.2.16.*

Proof. We show that $X \times Y$ does not have weak topology by finding a set that, by the weak topology, should be closed in $X \times Y$ but is not. For $i \in \iota$ and $j \in \mathbb{N}$ we define $p_{i,j} = (1/i_j, 1/i_j) \in A_i \times B_j$ where i_j is the j'th element of the sequence i. Set $P := \{p_{i,j} \mid i \in \iota, j \in \mathbb{N}\}$.

Let us first show that P would be closed if $X \times Y$ had weak topology. We need to show that its intersection with every closed cell of $X \times Y$ is closed. The closed cells of $X \times Y$ are the following: The 0-cells are products of 0-cells, i.e. singletons of the form (x, y) where $x \in \{1_i \mid i \in \iota\} \cup \{0_X\}$ and $y \in \{1_j \mid j \in \mathbb{N}\} \cup \{0_Y\}$. The intersection of P with any closed 0-cell is empty and therefore closed. The 1-cells are products of 0-cells with 1-cells. The two different options are $A_i \times \{x\}$ with $i \in \iota$ and $x \in \{1_i \mid i \in \iota\} \cup \{0_X\}$ and $\{y\} \times B_j$ with $y \in \{1_j \mid j \in \mathbb{N}\} \cup \{0_Y\}$ and $j \in \mathbb{N}$. The intersection of P with any closed 1-cell is thus also empty and closed. So lastly let us consider the 2-cells. They are of the form $A_i \times B_j$ with $i \in \iota$ and $j \in \mathbb{N}$. For the intersection we get $P \cap (A_i \times B_j) = p_{i,j}$ which is closed. Therefore P would be closed in the weak topology.

Now we prove that P is not closed in $X \times Y$. We show that the complement P^c of P in $X \times Y$ is not open by showing that every open neighbourhood of $(0_X, 0_Y)$ contains a point of P. A base for the product topology is given by

$$\{U \times V \mid U \subset X \text{ is open in X}, V \subseteq Y \text{ is open in Y}\}.$$

It is easy to see that it suffices to prove our desired property for the base. Now let us examine, what open neighbourhoods of 0_X in X look like. By the definition of the wedge sum, an open neighbourhood of 0_X is of the form $\bigvee_{i\in\iota} U_i$, where U_i is an open neighbourhood of 0 in A_i for $i \in \iota$. For each of the U_i's there is an $x_i > 0$ such that $[0, x_i) \subseteq U_i$. It is therefore enough to show our claim for these sets. Arguing in the same manner for Y allows us to reduce our aim to the set

$$\{(\bigvee_{i\in\iota}[0, x_i)) \times (\bigvee_{j\in\mathbb{N}}[0, y_j)) \mid x_i > 0 \text{ for all } i \in \iota, y_j > 0 \text{ for all } j \in \mathbb{N}\}.$$

Picking such an open neighbourhood $(\bigvee_{i\in\iota}[0, x_i)) \times (\bigvee_{j\in\mathbb{N}}[0, y_j))$ we need to find a p in P such that p is in that neighbourhood. We pick an $i' \in \iota$ such that for every $j \in \mathbb{N}$ we have $i'_j > \max(j, 1/y_j)$. Then we pick $j' \in \mathbb{N}$ such that $j' > 1/x_{i'}$. That gives us $1/i'_{j'} < 1/j' < x_{j'}$ and $1/i'_{j'} < y_{j'}$, which means that

$$p_{i',j'} = (1/i'_{j'}, 1/i'_{j'}) \in [0, x_{i'}) \times [0, y_{j'}) \subseteq (\bigvee_{i\in\iota}[0, x_i)) \times (\bigvee_{j\in\mathbb{N}}[0, y_j)).$$

Thus P is not closed. □

K-spaces and the k-ification

Before we can move on to discuss the product of CW-complexes, we need to discuss its topology. Therefore, we will now study k-spaces and the k-ification.

A k-space, or also called a compactly generated space, is defined for our purposes as follows. Note that we mean quasi-compactness when talking about compactness.

Definition 1.2.22. Let X be a topological space. We call X a *k-space* if

$$A \subseteq X \text{ is open} \iff \text{for all compact sets } C \subseteq X \text{ the intersection } A \cap C \text{ is open in } C.$$

↗

There are a lot of different definitions in the literature. The most popular ones all agree on Hausdorff spaces. An overview of these different notions can be found on Wikipedia [Wik24].

It will also be helpful to characterise k-spaces via closed sets:

Lemma 1.2.23. *Let X be a topological space. X is a k-space iff*

$$A \subseteq X \textit{ is closed} \iff \textit{for all compact sets } C \subseteq X \textit{ the intersection } A \cap C \textit{ is closed in } C.$$

↗

Proof. We only show the forward direction as the backward direction follows in the same way. Of the equivalence that we now need to show, the forward direction is trivial. Thus let $A \subseteq X$ be a set such that for all compact sets $C \subseteq X$, the intersection $A \cap C$ is closed in C. It is enough to show that A^c is open. By definition of the k-space, that is the case if for every compact set $C \subseteq X$ the intersection $A^c \cap C$ is open in C. Take any compact $C \subseteq X$. By assumption, $A \cap C$ is closed in C. Since $A \cap C$ is the complement of $A^c \cap C$ in C, this immediately gives us that $A^c \cap C$ is open in C. □

We also define a way to make any topological space into a k-space which we call the k-ification:

Definition 1.2.24. Let X be a topological space. We can define another topological space X_c on the same set by setting

$$A \subseteq X_c \text{ is open} \iff \text{for all compact sets } C \subseteq X \text{ the intersection } A \cap C \text{ is open in } C.$$

We call X_c the *k-ification* of X.

It is easy to see that this gives us a finer topology:

Lemma 1.2.25. *If $A \subseteq X$ is open then $A \subseteq X_c$ is open.*

Again, it it useful to characterise the closed sets in the k-ification:

Lemma 1.2.26. *$A \subseteq X_c$ is closed iff $A \cap C$ is closed in C for all compact sets $C \subseteq X$.*

Proof. Completely analogous to the proof of Lemma 1.2.23. □

To show that the k-ification actually fulfils its purpose of turning any space into a k-space, we first need the following lemma:

Lemma 1.2.27. *$A \subseteq X$ is compact iff $A \subseteq X_c$ is compact.*

Proof. For the backward direction notice that Lemma 1.2.25 is another way of stating that the map $\mathrm{id}\colon X_c \to X$ is continuous. As the image of a compact set under a continuous map, that makes $A \subseteq X$ compact.

For the forward direction take $A \subseteq X$ compact. To show that $A \subseteq X_c$ is compact, take an open cover $(U_i)_{i\in\iota}$ of A in X_c. For every $i \in \iota$ there is, by definition of the k-ification, an open set $V_i \subseteq X$ such that $V_i \cap A = U_i \cap A$. $(V_i)_{i\in\iota}$ is an (open) cover of A in X:

$$A = A \cap \bigcup_{i\in\iota} U_i = \bigcup_{i\in\iota}(A \cap U_i) = \bigcup_{i\in\iota}(A \cap V_i) = A \cap \bigcup_{i\in\iota} V_i \subseteq \bigcup_{i\in\iota} V_i.$$

Thus there is a finite subcover $(V_i)_{i\in\iota'}$ of A in X. $(U_i)_{i\in\iota'}$ is now a finite subcover of A in X_c:

$$A = A \cap \bigcup_{i\in\iota'} V_i = \bigcup_{i\in\iota'}(A \cap V_i) = \bigcup_{i\in\iota'}(A \cap U_i) = A \cap \bigcup_{i\in\iota'} U_i \subseteq \bigcup_{i\in\iota'} U_i.$$

□

Now we are ready to move on to the promised lemma:

Lemma 1.2.28. *X_c is a k-space for every topological space X.*

Proof. We need to show that a set $A \subseteq X_c$ is open iff $A \cap C$ is open in C for every compact set $C \subseteq X_c$. The forward implication is again trivial.

For the backward implication take a set $A \subseteq X_c$ such that for every compact set $C \subseteq X_c$ the intersection $A \cap C$ is open in C. By the definition of the k-ification, it is enough to show that for every compact set $C \subseteq X$ the intersection $A \cap C$ is open in C. So let $C \subseteq X$ be a compact set. By Lemma 1.2.27, C is also compact in X. By assumption, this means that $A \cap C$ is open in $C \subseteq X_c$ (in the subspace topology of the k-ification). Thus there is an open set $B \subset X_c$ such that $A \cap C = B \cap C$. By the definition of the k-ification, $B \cap C$ is open in $C \subseteq X$. That means there is an open set $E \subseteq X$ such that $B \cap C = E \cap C$. But that now gives us $A \cap C = B \cap C = E \cap C$, with which we can conclude that $A \cap C$ is open in $C \subseteq X$ (in the subspace topology of the original topology of X). □

If we already have a k-space, then the k-ification just maintains the topology of our space:

Lemma 1.2.29. *Let X be a k-space. Then the topologies of X and X_c coincide.*

Proof. Notice that the characterisation of open sets in X and X_c respectively agree in this setting. □

Corollary 1.2.30. *The k-ification is idempotent, i.e. $(X_c)_c = X_c$.*

Now we will characterise continuous maps to and from the k-ification. Going from the k-ification is not a big issue:

Lemma 1.2.31. *Let $f\colon X \to Y$ be a continuous map of topological spaces. Then $f\colon X_c \to Y$ is continuous.*

Proof. This follows easily from Lemma 1.2.25. □

More interesting questions are, when a map to the k-ification or a map from a k-ification to a k-ification is continuous. The following two lemmas and proofs, that answer these questions, are based on Lemma 46.4 of [Mun14]. The next lemma is the first step towards the answer:

Lemma 1.2.32. *Let X be a compact space and $f\colon X \to Y$ be a continuous map. Then $f\colon X \to Y_c$ is continuous.*

Proof. We want to show that for every closed $A \subseteq Y_c$ the preimage $f^{-1}(A)$ is closed in X. Take any closed set $A \subseteq Y_c$. We know by Lemma 1.2.26 that $A \cap C$ is closed in C for every compact $C \subseteq Y$. As the image of a compact set, $f(X)$ is compact. Thus $A \cap f(X)$ is closed in $f(X) \subseteq Y$. By the definition of the subspace topology, there is a closed set $B \subseteq Y$ such that $A \cap f(X) = B \cap f(x)$. Now we have

$$f^{-1}(A) = f^{-1}(A \cap f(X)) = f^{-1}(B \cap f(X)) = f^{-1}(B)$$

which is closed as the preimage of a closed set under a continuous map. □

Now this helps us get the following lemma:

Lemma 1.2.33. *Let $f\colon X \to Y$ be a map of topological spaces such that for every compact $C \subseteq X$, the restriction $f|_C : C \to Y$ is continuous. Then $f\colon X_c \to Y_c$ is continuous.*

Proof. The last lemma together with our assumption tells us that for every compact $C \subseteq X$, the restriction $f|_C : C \to Y_c$ is continuous. To show the claim take any open $A \subseteq Y_c$. We need to show that $f^{-1}(A) \subseteq X_c$ is open. By definition of the k-ification this set is open if for all compact sets $C \subseteq X$ the intersection $f^{-1}(A) \cap C$ is open in C. Take any compact set $C \subseteq X$. As noted above, we now know that $f|_C : C \to Y_c$ is continuous. Or in other words we know that for every open $B \subseteq Y_c$ there is an open set $E \subseteq X$ such that $f^{-1}(B) \cap C = E \cap C$. Applying this to the set $A \subseteq Y_c$ gives us an open set $E \subseteq X$ such that $f^{-1}(A) \cap C = E \cap C$. But that is just another way of stating that $f^{-1}(A)$ is open in $C \subseteq X$. □

That yields the following corollary:

Corollary 1.2.34. *Let $f\colon X \to Y$ be a continuous map of topological spaces. Then $f\colon X_c \to Y_c$ is continuous.*

Proof. This situation trivially fulfils the conditions of the previous lemma. □

If we look at the discussion of the product of CW-complexes in some topology books, for example [Hat01] and [Lüc05], we will notice that the k-ification rarely gets discussed in detail. One possible reason for this is that most common spaces that you encounter are already k-spaces. Lemma 1.2.29 then allows you to ignore the k-ification entirely. We will therefore discuss in the remainder of this section which spaces are k-spaces and which are not. The first example are weakly locally compact spaces.

Definition 1.2.35. Let X be a topological space. We call X *weakly locally compact* if every point $x \in X$ has some compact neighbourhood.

This property is in some sources just called locally compact. The following proof is from Lemma 46.3 in [Mun14].

Lemma 1.2.36. *Weakly locally compact spaces are k-spaces.*

Proof. Let X be a weakly locally compact space. Let $A \subseteq X$. We need to show that A is open iff $A \cap C$ is open in C for every compact set C. The forward direction is trivial. So assume that for every compact set C the intersection $A \cap C$ is open in C. A is open if it is a neighbourhood of every point $x \in A$. So fix any $x \in A$. Since X is weakly locally compact, x has a compact neighbourhood C. By definition of neighbourhoods there is an open set $U \subseteq C$ such that $x \in U$ and we need to find an open set $V \subseteq A$ such that $x \in V$. We show that $A \cap U$ fulfils these conditions. It is obvious that $A \cap U \subseteq A$ and $x \in A \cap U$. So it is left to show that $A \cap U$ is open. By assumption $A \cap C$ is open in C. That means that there is an open set B such that $A \cap C = B \cap C$. This now gives us

$$A \cap U = A \cap C \cap U = B \cap C \cap U = B \cap U$$

which is open as the intersection of two open sets. □

Another big class of spaces which are k-spaces are sequential spaces.

Definition 1.2.37. A set A in a topological space X is *sequentially closed* if for every convergent sequence contained in A its limit point is also in A. The *sequential closure* of a set A in X is defined as

$$\operatorname{scl}(A) = \{x \in X \mid \text{there is a sequence } (a_n)_{n\in\mathbb{N}} \subseteq A \text{ such that } (a_n)_{n\in\mathbb{N}} \text{ converges to } x\}.$$

A *sequential space* is a space in which all sequentially closed sets are closed.

We will need the following characterisation of sequentially closed sets:

Lemma 1.2.38. *A set $A \subseteq X$ is sequentially closed iff $A = scl(A)$.*

Proof. This is easy to see from the definitions. ☐

The following proof is based on [Sco16] and Lemma 46.3 in [Mun14].

Lemma 1.2.39. *Sequential Spaces are k-spaces.*

Proof. Let X be a Sequential Space. By Lemma 1.2.23 it is enough to show that

$$A \subseteq X \text{ is closed} \iff \text{for all compact sets } C \subseteq X \text{ the intersection } A \cap C \text{ is closed in } C.$$

The forward direction is trivial. Let A be a set such that $A \cap C$ is closed in C for every compact set C. Since X is a sequential space, it is enough to show that A is sequentially closed, or by the previous lemma that $A = \operatorname{scl}(A)$. The inclusion $A \subseteq \operatorname{scl}(A)$ is obvious. For the backward inclusion take $x \in \operatorname{scl}(A)$. We need to show that $x \in A$. By definition there is a sequence $(a_n)_{n\in\mathbb{N}} \subseteq A$ that converges to x. It is well known (and can be shown directly from the definition of compactness) that the set $\{a_n \mid n \in \mathbb{N}\} \cup x$ is compact as the set of terms of a sequence together with the limit point of that sequence. By assumption, that gives us that $A \cap (\{a_n \mid n \in \mathbb{N}\} \cup x)$ is closed in $\{a_n \mid n \in \mathbb{N}\} \cup x$. In other words there is a closed set B such that

$$A \cap (\{a_n \mid n \in \mathbb{N}\} \cup x) = B \cap (\{a_n \mid n \in \mathbb{N}\} \cup x).$$

With that we get

$$x \in A \iff x \in A \cap (\{a_n \mid n \in \mathbb{N}\} \cup x) = B \cap (\{a_n \mid n \in \mathbb{N}\} \cup x) \iff x \in B$$

and for all $n \in \mathbb{N}$ we get $a_n \in B$ in the exact same way. Thus $(a_n)_{n\in\mathbb{N}} \subseteq B$. Since B is in particular sequentially closed, this gives us $x \in B$, which is enough by the above equivalence. ☐

In particular sequential spaces include metric spaces:

Lemma 1.2.40. *Metric spaces are sequential spaces.*

Proof. Let X be a metric space and A be a sequentially closed set. We need to show that A is closed which is equivalent to A^c being open. Assume towards a contradiction that A^c is not open. Then there is a point $x \in A^c$ such that for every $n \in \mathbb{N}$ the open ball $\mathrm{B}_{1/n}(x)$ contains a point $x_n \in A$. But then we have a sequence $(x_n)_{n\in\mathbb{N}} \subseteq A$ that converges to $x \in A^c$. Thus A is not sequentially closed. We have therefore arrived at a contradiction. ☐

Corollary 1.2.41. *Metric spaces are k-spaces.*

Lastly we will discuss spaces that are not k-spaces:

Lemma 1.2.42. *Let X be an anti-compact T_1 space. Then X_c has discrete topology.*

Proof. Let $A \subseteq X_c$ be any set. We need to show that it is open. By the definition of the k-ification, it is enough to show that $A \cap C$ is open in C for every compact set $C \subseteq X$. Since X is anti-compact, C is finite. And by T_1 every finite set has discrete topology. Thus $A \cap C$ is open in C and X_c has discrete topology. □

Corollary 1.2.43. *Let X be a non-discrete anti-compact T_1 space. Then X is not a k-space.*

Proof. This follows easily from the previous lemma and Lemma 1.2.29. □

That leads us to our first concrete example of a space that is not a k-space:

Example 1.2.44. Let X be any uncountable set. Equip X with the cocountable topology, i.e. let a set $A \subseteq X$ be open iff $A = \varnothing$ or A^c is countable. Then X is not a k-space.

Proof. It is easy to see by going through the axioms that the cocountable topology is indeed a topology. We will now show that this space satisfies the conditions of the previous corollary. X is clearly non-discrete. To see that X is a T_1 space take two distinct points a and b. Now let A be the set $X \setminus \{b\}$. This set is open since $\{b\}$ is countable and it obviously does not contain b. We lastly need to show that that X is anti-compact. To do that take any set $A \subseteq X$. Pick an (if possible infinite) countable subset $B \subseteq A$. Now for every $b \in B$ define $U_b = (X \setminus B) \cup \{b\}$. Since $U_b^c = B \setminus \{b\}$ is countable U_b, is open for every $b \in B$. It is also easy to see that $A \subseteq \bigcup_{b \in B} U_b$. Thus $(U_b)_{b \in B}$ is an open cover of A. But since for every $b \in B$ there is no $b' \in B$ with $b \neq b'$ and $b \in U_{b'}$, $(U_b)_{b \in B}$ cannot have a proper subcover. Therefore A can only be compact if all these possible covers are already finite. That can only be the case if B and with that A are finite. □

Other examples can be found on π-base [PiB24].

Constructing the product

We can now move on to discuss the correct version of Remark 1.2.16. For the rest of the section let X, Y be CW-complexes with families of characteristic maps $(Q_i^n \colon D_i^n \to X)_{n \in \mathbb{N}, i \in I_n}$ and $(P_j^m \colon D_j^m \to Y)_{m \in \mathbb{N}, j \in J_n}$. We will write the cells of X as e_i^n and the cells of Y as f_j^m. We want to show:

Theorem 1.2.45. *There is a CW-structure on $(X \times Y)_c$ with characteristic maps $(Q_i^n \times P_j^m \colon D_i^n \times D_j^m \to (X \times Y)_c)_{n,m \in \mathbb{N}, i \in I_n, j \in J_m}$. The indexing sets $(K_l)_{l \in \mathbb{N}}$ are given by $K_l = \bigcup_{n+m=l} I_n \times J_m$ for every $l \in \mathbb{N}$ and the cells are therefore of the form $e_i^n \times f_j^m$ for $n, m \in \mathbb{N}$, $i \in I_n$ and $j \in J_m$.*

We will split the proof up into lemmas to have a better overview of the proof. Let us first show that the issue that occurred with the counterexample in an earlier section in Lemma 1.2.21 works out here:

Lemma 1.2.46. $(X \times Y)_c$ *has weak topology, i.e.* $A \subseteq (X \times Y)_c$ *is closed iff* $\overline{e}_i^n \times \overline{f}_j^m \cap A$ *is closed for all* $n, m \in \mathbb{N}$, $i \in I_n$ *and* $j \in J_m$.

Proof. The forward direction follows from the fact that the product of closed sets is closed in the product topology and from Lemma 1.2.25 that tells us that the k-ification is finer than the product topology.

Moving on to the backward direction we know by Lemma 1.2.29 that the k-ification is a k-space and by Lemma 1.2.26 that A is closed if for every compact set $C \subseteq (X \times Y)_c$, $A \cap C$ is closed in C. Take such a compact set C. The projections $\mathrm{pr}_1(C)$ and $\mathrm{pr}_2(C)$ are compact as images of a compact set. By Lemma 1.1.20 there are finite sets $E \subseteq \{e_i^n \mid n \in \mathbb{N}, i \in I_n\}$ and $F \subseteq \{f_j^m \mid m \in \mathbb{N}, j \in J_m\}$ s.t $\mathrm{pr}_1(C) \subseteq \bigcup_{e \in E} e$ and $\mathrm{pr}_2(C) \subseteq \bigcup_{f \in F} f$. Thus

$$C \subseteq \mathrm{pr}_1(C) \times \mathrm{pr}_2(C) \subseteq \bigcup_{e \in E} e \times \bigcup_{f \in F} f = \bigcup_{e \in E} \bigcup_{f \in F} e \times f.$$

So C is included in a finite union of cells of $(X \times Y)_c$. Therefore

$$A \cap C = A \cap \left(\bigcup_{e \in E} \bigcup_{f \in F} e \times f \right) \cap C = \left(\bigcup_{e \in E} \bigcup_{f \in F} A \cap (e \times f) \right) \cap C$$

is closed since by assumption $A \cap (e \times f)$ is closed for every e and f and the union is finite. Thus $A \cap C$ is in particular closed in C. $\square$

Before we can discuss closure finiteness we need to think about what the frontiers of cells look like in $(X \times Y)_c$:

Lemma 1.2.47. *Let* $e_i^n \times f_j^m$ *for* $n, m \in \mathbb{N}$, $i \in I_n$ *and* $j \in J_m$. *The frontier of that cell is* $\partial e_i^n \times \overline{f}_j^m \cup \overline{e}_i^n \times \partial f_j^m$.

Proof. The definition of the frontier gives us:

$$\begin{aligned}
(Q_i^n \times P_j^m)(\partial D^{n+m}) &= (Q_i^n \times P_j^m)(\partial D^n \times D^m \cup D^n \times \partial D^m) \\
&= (Q_i^n \times P_j^m)(\partial D^n \times D^m) \cup (Q_i^n \times P_j^m)(D^n \times \partial D^m) \\
&= Q_i^n(\partial D^n) \times P_j^m(D^m) \cup Q_i^n(D^n) \times P_j^m(\partial D^m) \\
&= \partial e_i^n \times \overline{f}_j^m \cup \overline{e}_i^n \times \partial f_j^m.
\end{aligned}$$

The equality $\partial D^{n+m} = \partial D^n \times D^m \cup D^n \times \partial D^m$ is true in any metric space and can be verified explicitly. $\square$

Using this we get closure finiteness:

Lemma 1.2.48. $(X \times Y)_c$ *has closure finiteness, i.e. each frontier of a cell is contained in a finite union of closed cells of a lower dimension.*

Proof. By the above lemma we need to verify that for all $n, m \in \mathbb{N}$, $i \in I_n$ and $j \in J_m$, the set $\partial e_i^n \times \overline{f}_j^m \cup \overline{e}_i^n \times \partial f_j^m$ is contained in a finite union of closed cells of $(X \times Y)_c$ of dimension less than $n + m$. We can show this separately for $\partial e_i^n \times \overline{f}_j^m$ and $\overline{e}_i^n \times \partial f_j^m$. We will do the proof for the the former as both proofs work in the same way. Since X fulfils closure finiteness, there is a finite set E of cells of X of dimension less than n such that $\partial e_i^n \subset \bigcup_{e \in E} \overline{e}$. But that gives us $\partial e_i^n \times \overline{f}_j^m \subseteq \bigcup_{e \in E} \overline{e} \times \overline{f}_j^m$, which is a finite union of closed cells of $(X \times Y)_c$ of dimension less than $n + m$. $\square$

Now we can proof the desired theorem:

Proof of Theorem 1.2.45. It is well known and easy to see explicitly that the product of two Hausdorff spaces is again Hausdorff. Now we can go through the five conditions of Definition 1.1.2.

Property (i) is given by the fact that the product of bijective maps is again a bijection and continuity in both directions follows from Lemma 1.2.31 and Lemma 1.2.32.

For property (ii) pick any $n, m, n', m' \in \mathbb{N}$, $i \in I_n$, $j \in J_m$, $i' \in I_{n'}$ and $j' \in J_{m'}$ such that $(n+m, i, j) \neq (n'+m', i', j')$ then either $(n, i) \neq (n', i')$ or $(m, j) \neq (m', j')$. Thus

$$\overline{e}_i^n \times \overline{f}_j^m \cap \overline{e}_{i'}^{n'} \times \overline{f}_{j'}^{m'} = \left(\overline{e}_i^n \cap \overline{e}_{i'}^{n'}\right) \times \left(\overline{f}_j^m \cap \overline{f}_{j'}^{m'}\right) = \varnothing$$

since X and Y themselves fulfil property (ii).

We already covered property (iii) in Lemma 1.2.48 and property (iv) in Lemma 1.2.46. Property (v) is immediate. □

2 Lean and mathlib

In this chapter we will discuss some general concepts about Lean and its mathematical library mathlib. We will first explain the logic, i.e. the type theory, that is used in Lean. While it is helpful to know some theory about it, it is not necessary to understand the type theory in depth to formalise mathematics in Lean or read and understand the rest of this thesis. Most important are some constructions, that are explained at the end of the following section, and how you can use them in practice.

2.1 The type theory of Lean

Type theory was first proposed by Russell in 1908 [Rus08] as a way to axiomatise mathematics and resolve the paradoxes (most famously Russell's paradox) that were discussed at the time. While type theory has lost its relevance as a foundation of mathematics to set theory it has since been studied in both mathematics and computer science. It was first used in formal mathematics in 1967 in the formal language *AUTOMATH*. More about the history of type theory can be found in [KLN04]. Discussions of type theory in mathematics and especially its connections to homotopy theory forming the new area of *homotopy type theory* can be found in [Uni13]. We will now focus on the type theory as used in Lean. Its type theory along with the type theory of other proof assistants such as *Coq* are based on *constructive type theory* developed by Per Martin-Löf which makes use of dependent types [MS84]. A detailed account of Lean's type theory can be found in [Car19]. The following short discussion is based on [Avi+24].

Lean uses what is called a *dependent type theory*. In type theory every object has a type. A type can for example be the natural numbers or propositions, which we write in Lean as `Nat` or `ℕ` and `Prop` respectively. To assert that `n` is a natural number or that `p` is a proposition we write `n : ℕ` and `p : Prop`. Proofs of a proposition `p` also form a type written as `p`. If you want to say that `hp` is a proof of `p` then you can simply write `hp : p`. Something to note about proofs in Lean is that, contrary to other type theories, the type theory of Lean has *proof irrelevance* which means that two proofs of a proposition `p` are by definition assumed to be the same.

Even types themselves have types. In Lean the type of natural numbers `ℕ` has the type `Type`. The type of propositions `Prop` is also of type `Type`. As to not run into a paradox called Girard's paradox there is a hierarchy of types [Coq86]. The type of `Type` is `Type 1`, the type of `Type 1` is `Type 2` and so on. These are called type-universes. The notation `α : Type*` is a way of stating that `α` is a type in an arbitrary universe.

There are a few ways to construct new types from existing ones. Some of them are very similar to constructions on sets such as the cartesian product of types written as `α × β` or the type of functions from `α` to `β` written as `α → β` where `α` and `β` are types. Elements of `α × β` can be written as `(a, b)` for every `a : α` and `b : β`. Elements of `α → β` can be written as `fun a ↦ s a` for some `s : α → β`. Since these are quite self-explanatory, we will not go into more detail. We will now mainly discuss constructions that do not fulfil this criterium. A first example is the sum type of two types `α` and `β` written as `α ⊕ β` which is the equivalent to a disjoint union of sets. Elements of this type are of the form `Sum.inl a` for `a : α` or `Sum.inr b` for `b : β`. When given an `x : α ⊕ β` we can use the construction

```
match x with
| Sum.inl a => ···
| Sum.inr b => ···
```

to write two different definitions or proofs depending on whether `x` originates from an element of `α` or `β`. In this code snippet the names `a` and `b` are arbitrary.

The next two examples explain why this type theory is a dependent type theory: If we have a type `α` and for every `(a : α)` a type `β a` (i.e. `β` is a function assigning a type to every `a : α`) then we can construct the *pi type* or dependent function type written as `(a : α) → β a` or `Π (a : α), β a`. We can construct an element of this type by writing `fun a ↦ s a` for some `s : (a : α) → β a`. Here is an example: Assume that you want a function that for every pair in a cartesian product `α × α` for any type `α` returns the first element. Then this would be a function that depends on `α` and whose type is therefore the dependent function type `(α : Type*) → α × α → α`.

The dependent version of the cartesian product is called a *sigma type* and can be written as `(a : α) × β a` or `Σ a : α, β a` for `α` and `β` the same way as above. An element of the sigma type can be written as `⟨a, b⟩` for `a : α` and `b : β a`. When given an element of the sigma type `x : Σ a : α, β a` one can write `obtain ⟨a, b⟩ := x` to deconstruct `x`.

2.2 Implicit arguments and typeclass inference

A crucial factor that makes Lean more comfortable to use and makes the formalisation process feel closer to doing mathematics on paper is its use of *implicit arguments* and *typeclass inference*. We will explain both of these concepts in this section.

First let us discuss implicit arguments based on [Avi+24]. One way that we could define continuity in Lean is the following[1]:

```
structure Continuous' (X Y : Type*) (t : TopologicalSpace X)
    (s : TopologicalSpace Y) (f : X → Y) : Prop where
  isOpen_preimage : ∀ s, IsOpen s → IsOpen (f ⁻¹' s)
```

Where a structure is a construct that can bundle both data and properties after the keyword `where`. This structure has no data and one property which is named `isOpen_preimage`. `f ⁻¹' s` denotes the preimage of `s` under `f`. But now if we are given two types `X` and `Y` with topologies `s` and `t` respectively and a map `f : X → Y`, the statement that the map `f` is continuous would be expressed in the following way:

```
example (X Y : Type*) (t : TopologicalSpace X) (s : TopologicalSpace Y)
  (f : X → Y) : Continuous' X Y t s f := ···
```

where everything before the colon is the context we described above and after the colon equal you could write a proof.

[1]The code in this section will run if `import Mathlib.Topology.MetricSpace.Basic` is written at the top of the file.

One thing that we can notice is that the types X and Y are contained in the definition of f which means that Lean should be able to find that information itself. To tell Lean to do that you can replace the variables by underscores:

```
example (X Y : Type*) (t : TopologicalSpace X) (s : TopologicalSpace Y)
  (f : X → Y) : Continuous' _ _ t s f := ⋯
```

These two arguments are always clear from the context in this way. We therefore want to specify in the definition that they should not be given explicitly but instead inferred by the system. We use curly brackets to do this:

```
structure Continuous'' {X Y : Type*} (t : TopologicalSpace X)
    (s : TopologicalSpace Y) (f : X → Y) : Prop where
  isOpen_preimage : ∀ s, IsOpen s → IsOpen (f ⁻¹' s)
```

which enables us to write continuity like this:

```
example (X Y : Type*) (t : TopologicalSpace X) (s : TopologicalSpace Y)
  (f : X → Y) : Continuous'' t s f := ⋯
```

This is already a lot shorter than what we had above but there is still room for improvement, as on paper we would probably just write "f is continuous" since in most contexts X and Y will only have one specified topology each, that can be inferred by the reader. The same thing is also true in Lean and we can achieve this by typeclass inference. Typeclasses were first invented by Wadler and Blott in [WB89] to be used in the programming language Haskell. They are a way to overload operations for various different types. For example, you might want to write code that works for all types that have a topology. In Lean this is possible by just stating that your input type X is part of the typeclass TopologicalSpace. You can specify that something ia a typeclass with the keyword class. The definition of the typeclass of topological spaces in mathlib looks like this:

```
class TopologicalSpace (X : Type*) where
  protected IsOpen : Set X → Prop
  protected isOpen_univ : IsOpen univ
  protected isOpen_inter : ∀ s t, IsOpen s → IsOpen t → IsOpen (s ∩ t)
  protected isOpen_sUnion : ∀ s, (∀ t ∈ s, IsOpen t) → IsOpen (⋃₀ s)
```

Let us first explain what this code means: The keyword protected means that these properties should not be accessed directly because there are lemmas that should be used instead. Set X is the type that consists of all sets of elements of X. Thus the line protected IsOpen : Set X → Prop expresses that IsOpen is a property that can be assigned to a set in X. The rest of the lines discuss the properties of a topology. univ is the set that is composed of all elements of X and ⋃₀ s is the union over the set s. All of these explanations are not actually relevant to typeclasses, they are just for our understanding of the above code.

Typeclasses are also expected to be inferred automatically. Local instances of these typeclasses can be written with square brackets, which tells Lean to infer these automatically.

We can now look at the version of continuity that is almost identical to that of mathlib:

```
structure Continuous {X Y : Type*} [t : TopologicalSpace X]
    [s : TopologicalSpace Y] (f : X → Y) : Prop where
  isOpen_preimage : ∀ s, IsOpen s → IsOpen (f ⁻¹' s)
```

which enables us to write that f is continuous in the context explained above as follows:

```
example (X Y : Type*) (t : TopologicalSpace X) (s : TopologicalSpace Y)
    (f : X → Y) : Continuous f := ⋯
```

When you define a class you can then define instances of that class to be inferred whenever you talk you talk about a type with that instance. Mathlib defines the discrete topology on ℤ as an instance:

```
instance : TopologicalSpace ℤ := ⊥
```

where ⊥ is the smallest element in the order that can be defined on the topologies of a space, i.e. the finest topology which is the discrete topology. That makes it so that for any map `f : ℤ → ℤ` we can just write the following

```
example (f : ℤ → ℤ) : Continuous f := ⋯
```

and Lean automatically knows which topology we are talking about. We can additionally say that an instance implies another instance. If you have types `X` and `Y` which both have topologies defined on them this instance in mathlib gives you a topology on the product:

```
instance instTopologicalSpaceProd {X Y : Type*} [t₁ : TopologicalSpace X]
  [t₂ : TopologicalSpace Y] : TopologicalSpace (X × Y) := ⋯
```

which enables you to write the following

```
example {X Y Z : Type*} [TopologicalSpace X] [TopologicalSpace Y]
  [TopologicalSpace Z] (f : X × Y → Z) : Continuous f := ⋯
```

This also works across different typeclasses. We can write

```
example {X : Type*} [MetricSpace X] (f : X → X) : Continuous f := ⋯
```

which works because Lean knows that a metric space is by definition a pseudometric space which is a uniform space which is by definition a topological space.

3 Lean formalisation of CW-complexes

In this chapter we will discuss our formalisation of CW-complexes and will particularly focus on parts that diverge from the pure mathematics of the first chapter.

3.1 Definition and basic properties of a CW-complex

We choose to formalise the historical definition because the modern one would require us to consider the n-skeletons as topological spaces, which would create a lot of work unifying them to work well together. The historical definition, however, allows us to work in one topological space which avoids this issue.

The following is our definition of CW-complexes in Lean:

```
class CWComplex.{u} {X : Type u} [TopologicalSpace X] (C : Set X) where
  cell (n : ℕ) : Type u
  map (n : ℕ) (i : cell n) : PartialEquiv (Fin n → ℝ) X
  source_eq (n : ℕ) (i : cell n) : (map n i).source = ball 0 1
  cont (n : ℕ) (i : cell n) : ContinuousOn (map n i) (closedBall 0 1)
  cont_symm (n : ℕ) (i : cell n) : ContinuousOn (map n i).symm (map n i).target
  pairwiseDisjoint' :
    (univ : Set (Σ n, cell n)).PairwiseDisjoint (fun ni ↦ map ni.1 ni.2 '' ball
    0 1)
  mapsto (n : ℕ) (i : cell n) : ∃ I : Π m, Finset (cell m),
    MapsTo (map n i) (sphere 0 1) (⋃ (m < n) (j ∈ I m), map m j '' closedBall 0 1)
  closed' (A : Set X) (asubc : A ⊆ C) : IsClosed A ↔ ∀ n j, IsClosed (A ∩ map n
    j '' closedBall 0 1)
  union' : ⋃ (n : ℕ) (j : cell n), map n j '' closedBall 0 1 = C
```

The `.{u}` is a way to fix a universe level so that our definition of a CW-complex does not depend on a number of different universe levels: The one of `X` and the one of `cell n` for every $n \in \mathbb{N}$. `cell (n : ℕ)` represents the indexing set that we called I_n in Definition 1.1.2. `map (n : ℕ) (i : cell n)` represent what we called Q_i^n in that definition.

`Fin n` is the set containing n natural numbers starting at 0. `Fin n → ℝ` is one way to express $\mathbb{R}^n$ in Lean. `PartialEquiv` is a structure defined in mathlib as follows:

```
structure PartialEquiv (α : Type*) (β : Type*) where
  toFun : α → β
  invFun : β → α
  source : Set α
  target : Set β
  map_source' : ∀ ⦃x⦄, x ∈ source → toFun x ∈ target
  map_target' : ∀ ⦃x⦄, x ∈ target → invFun x ∈ source
  left_inv' : ∀ ⦃x⦄, x ∈ source → invFun (toFun x) = x
  right_inv' : ∀ ⦃x⦄, x ∈ target → toFun (invFun x) = x
```

It bundles two maps and two sets that get mapped to each other by the respective maps. Restricting the maps to these sets yields two maps that are the inverse of each other. We use this instead of a similar construction called `Equiv` for bijections to avoid explicitly having to deal with restrictions. The brackets ⦃⦄ are similar to the curly brackets and are used here since `x` can be inferred from the left sides of the implications.

The property `source_eq` specifies the source of the `PartialEquiv`. `cont` and `cont_symm` make the bijection into a homeomorphism giving us property (i) of Definition 1.1.2. The property `pairwiseDisjoint'` corresponds to property (ii) of Definition 1.1.2. We are adding the prime to its name because we will later see a lemma called `pairwiseDisjoint` that we prefer to use. `fun a : α ↦ f a` for `f : α → β` is a way to construct a map.

`mapsto` is the equivalent of property (iii) of the definition of a CW-complex. The Π defines a dependent function type which we discussed in Section 2.1. `Finset α` is the type of all finite sets in a type α. It can be imagined as a set bundled with the information that it is finite (but note that the actual definitions of `Finset α` and `Set α` are quite different). `MapsTo` is defined in mathlib as

```
def MapsTo (f : α → β) (s : Set α) (t : Set β) : Prop :=
  ∀ ⦃x⦄, x ∈ s → f x ∈ t
```

and is relatively self-explanatory.

`closed'` represents property (iv) of Definition 1.1.2 and `union'` represents property (v).

We have chosen this to be a class so that we can make use of typeclass inference which we explained in Section 2.2.

There are a few things to note about this formalisation of the definition. First of all it does not require X to be a Hausdorff space. This is done so that when you define a CW-complex, you can choose to first define the structure in this way and later show that it is a Hausdorff space to apply lemmas about CW-complexes, of which most will require that X is Hausdorff. Additionally we introduce a relative component: Instead of defining what it means for a space to be a CW-complex, we define what it means for a subspace C of X to be a CW-complex in X. This is useful, firstly, to be able to work with a nicer topology: If you consider S^1 as a CW-complex and a subspace of $\mathbb{R}$ you might find it easier to work with the topology on $\mathbb{R}$ instead of the subspace topology. Secondly, constructions such as attaching cells or taking disjoint unions of CW-complexes might be easier to work with, if you are already working in the same overarching type. This approach is inspired by [Gon+13], where the authors notice that it is helpful to consider subsets of an ambient group to avoid having to work with different group operations and similar issues.

One question that naturally arises is whether these changes to the definition preserve the notion of a CW-complex. Firstly note that if we choose X and C to be the same we recover Definition 1.1.2 exactly. Now let us think about what happens if we choose X and C to be different. Firstly this allows us to conclude that C is closed:

Lemma 3.1.1. *Let X be a Hausdorff space and and C a CW-complex in X as in the formalised definition. Then C is closed.*

Proof. Since $C \subseteq C$, it is enough to show that $C \cap \overline{e}_i^n$ is closed for every $n \in \mathbb{N}$ and $i \in I_n$. But by the property `union'` we know that $C \cap \overline{e}_i^n = \overline{e}_i^n$ which is closed by the same argument as in the proof of Lemma 1.1.6. □

This indeed excludes some CW-complexes:

Example 3.1.2. Let $I \subseteq \mathbb{R}$ be an open interval. Then I is a CW-complex in the sense of Definition 1.1.2.

Proof. Since I is homeomorphic to $\mathbb{R}$, it is by Lemma 1.2.3 enough to show that $\mathbb{R}$ admits the structure of a CW-complex. As 0-cells we choose every $z \in \mathbb{Z} \subseteq \mathbb{R}$. As 1-cells we choose the intervals $(z, z+1)$ for every $z \in \mathbb{Z} \subseteq \mathbb{R}$. Properties (i), (ii), (iii), and (v) of Definition 1.1.2 are easy to verify. We will therefore focus on property (iv), i.e. the weak topology. The forward implication follows in the same manner as in a lot of other proofs. Let us thus move on to the backwards direction. Take $A \subseteq \mathbb{R}$ and assume that $A \cap [z, z+1]$ is closed for all $z \in \mathbb{Z}$. We now need to show that A is closed. It is well-known that $\mathbb{R}$ is a metric space and by Lemma 1.2.40 it is in particular a sequential space. It is therefore enough to show that for every convergent sequence $(a_n)_{n\in\mathbb{N}} \subseteq A$ the limit point is also in A. Take an arbitrary convergent $(a_n)_{n\in\mathbb{N}} \subseteq A$. We call the limit point a. Then there exists a $z \in \mathbb{Z}$ such that $a \in (z, z+2)$. Thus there is a subsequence $(a'_n)_{n\in\mathbb{N}} \subseteq A \cap [z, z+2]$, which obviously also converges to a. But, by assumption, $A \cap [z, z+2] = (A \cap [z, z+1]) = (A \cap [z+1, z+2])$ is closed and therefore sequentially closed, which gives us that $a \in A \cap [z, z+2] \subseteq A$. $\square$

But remember that our definition in Lean still allows us to view an open interval as a CW-complex in itself.

And every space that fulfils the formalised definition also fulfils Definition 1.1.2:

Lemma 3.1.3. *Let C be a CW-complex in a Hausdorff space X as in the definition in the formalisation. Then C is a CW-complex as in Definition 1.1.2.*

Proof. Properties (i), (ii), (iii) and (v) of Definition 1.1.2 are immediate. Thus let us look at property (iv). We assume that

$$A \subseteq C \text{ is closed in } X \iff \overline{e}_i^n \cap A \text{ is closed in } X \text{ for all } n \in \mathbb{N} \text{ and } i \in I_n$$

and need to show that

$$A \subseteq C \text{ is closed in } C \iff \overline{e}_i^n \cap A \text{ is closed in } C \text{ for all } n \in \mathbb{N} \text{ and } i \in I_n.$$

It is easy to see that the forward direction is true. For the backwards direction take $A \subseteq C$ such that $A \cap \overline{e}_i^n$ is closed in C for all $n \in \mathbb{N}$ and $i \in I_n$. That means that for every $n \in \mathbb{N}$ and $i \in I_n$ there is a closed set $B_i^n \subseteq X$ such that $B_i^n \cap C = A \cap \overline{e}_i^n$. But since C is closed by Lemma 3.1.1 that means that $A \cap \overline{e}_i^n$ was already closed for every $n \in \mathbb{N}$ and $i \in I_n$. Thus we are done by assumption. $\square$

With that we can move onto the last important difference that our new definition has. While `Fin n → ℝ` is a way to represent $\mathbb{R}^n$ in Lean it does not actually carry the euclidean metric but the maximum metric. So instead of considering closed balls we are looking at cubes which does not change our definition since the two are homeomorphic. We could use the euclidean metric on $\mathbb{R}^n$ which would be written as `EuclideanSpace ℝ (Fin n)` but since we are mostly arguing abstractly about CW-complexes this is unnecessary and takes up more space.

When proving that something is a CW-complex in Lean, it can often be surprising how much longer and more technical the formalised proofs of closure finiteness, i.e. the property `mapsto`, can be. One reason for this is that our proofs in the first chapter heavily rely on our intuition about finiteness while Lean obligates us to show finiteness of the union explicitly and in detail.

The code in the rest of the section will always have the following assumptions:

```
variable {X : Type*} [t : TopologicalSpace X] [T2Space X] {C : Set X}
  [CWComplex C]
```

where `T2Space X` expresses that `X` is a Hausdorff space.

We also want to define some notation in Lean. Just as we defined $\overline{e}_i^n$ to represent $Q_i^n(D_i^n)$ and similar notations in Definition 1.1.2 we can do the same in the formalisation:

```
def openCell (n : ℕ) (i : cell C n) : Set X := map n i '' ball 0 1

def closedCell (n : ℕ) (i : cell C n) : Set X := map n i '' closedBall 0 1

def cellFrontier (n : ℕ) (i : cell C n) : Set X := map n i '' sphere 0 1
```

We can now state some of the properties of our definition with this new notation. We restate `pairwiseDisjoint'` in two ways:

```
lemma pairwiseDisjoint :
    (univ : Set (Σ n, cell C n)).PairwiseDisjoint (fun ni ↦ openCell ni.1 ni.2)
    := ⋯

lemma disjoint_openCell_of_ne {n m : ℕ} {i : cell C n} {j : cell C m}
    (ne : (⟨n, i⟩ : Σ n, cell C n) ≠ ⟨m, j⟩) :
    openCell n i ∩ openCell m j = ∅ := ⋯
```

The second one is especially convenient to use as the hypothesis `ne` can often be automatically verified by a tactic called `aesop`. Information on `aesop` can be found in [LF23].

The properties `closed'` and `union'` can be rewritten with the new notation as follows:

```
lemma closed (A : Set X) (asubc : A ⊆ C) :
  IsClosed A ↔ ∀ n (j : cell C n), IsClosed (A ∩ closedCell n j) := ⋯

lemma union : ⋃ (n : ℕ) (j : cell C n), closedCell n j = C := ⋯
```

As in Definition 1.1.2, we also want to define notation for the n-skeletons. In the first chapter we often chose to start inductions at -1 to make the base case trivial. When formalising we want to be able to use the already defined induction principles, that naturally start at 0. For that purpose we use an auxiliary definition called `levelaux` that is shifted by 1 in comparison to the usual notion of the n-skeleton, which we call `level`:

```
def levelaux (C : Set X) [CWComplex C] (n : ℕ∞) : Set X :=
  ⋃ (m : ℕ) (_ : m < n) (j : cell C m), closedCell m j

def level (C : Set X) [CWComplex C] (n : ℕ∞) : Set X :=
  levelaux C (n + 1)
```

Note that we are choosing `n` in ℕ∞ which is the type of natural numbers extended by infinity which can be written as ⊤. Since `level` is defined in terms of `levelaux` it is often trivial to derive a lemma about `level` from the corresponding lemma about `levelaux`.

We can also define what it means for a CW-complex to be finite dimensional, of finite type, or finite:

```
class FiniteDimensional.{u} {X : Type u} [TopologicalSpace X] (C : Set X)
    [CWComplex C] : Prop where
  eventually_isEmpty_cell : ∀ᶠ n in Filter.atTop, IsEmpty (cell C n)

class FiniteType.{u} {X : Type u} [TopologicalSpace X] (C : Set X) [CWComplex C] :
     Prop where
  finite_cell (n : ℕ) : Finite (cell C n)

class Finite.{u} {X : Type u} [TopologicalSpace X] (C : Set X) [CWComplex C] :
    Prop where
  eventually_isEmpty_cell : ∀ᶠ n in Filter.atTop, IsEmpty (cell C n)
  finite_cell (n : ℕ) : Finite (cell C n)
```

Property `eventually_isEmpty_cell` is stated in terms of a *filter* which is a concept that appears frequently in mathlib. They are often used to describe convergence in a topological way. As they will not be important to this thesis, we will not go into detail but information on filters can be found in [Bou66]. The property `eventually_isEmpty_cell` is equivalent to `∃ a, ∀ (b : ℕ), a ≤ b → IsEmpty (cell C b)`.

Interestingly, our approach to the formalised definition provides us with a different way to prove Lemma 1.1.14, i.e. the fact that the n-skeletons are closed: The n-skeletons are CW-complexes by Lemma 1.2.1, which we will prove in Lean in the next section, and therefore closed by Lemma 3.1.1.

To finish of this section here are the statements of some of the main results of Section 1.1. They correspond to the results 1.1.10, 1.1.15, 1.1.19, 1.1.20 and 1.1.21.

```
lemma iUnion_openCell_eq_level (n : ℕ∞) :
  ⋃ (m : ℕ) (_ : m < n + 1) (j : cell C m), openCell m j = level C n := ⋯

lemma cellFrontier_subset_finite_openCell (n : ℕ) (i : cell C n) :
    ∃ I : Π m, Finset (cell C m), cellFrontier n i ⊆
    (⋃ (m < n) (j ∈ I m), openCell m j) := ⋯

lemma isDiscrete_level_zero {A : Set X} : IsClosed (A ∩ level C 0) := ⋯

lemma compact_inter_finite (A : Set X) (compact : IsCompact A) :
  _root_.Finite
  (Σ (m : ℕ), {j : cell C m // ¬ Disjoint A (openCell m j)}) := ⋯

lemma compact_iff_finite : IsCompact C ↔ Finite C := ⋯
```

Where `{a : α // P a}` is the subtype of α of all `a : α` such that `P a` for `P : α → Prop`.

3.2 Constructions

We can now look at the formalisation of the constructions that we covered in Section 1.2.

3.2.1 Miscellaneous constructions

Formalising the CW-complex structure on the n-skeleton, the disjoint union and the image of CW-complex under a homeomorphism is relatively straightforward. For the first two we take advantage of the relative approach of looking at subspaces as CW-complexes, as this helps us avoid having to deal with the subspace and disjoint topology.

The following code snippet includes the statements and assumptions:

```
def CWComplex_of_Homeomorph.{u} {X Y : Type  u} [TopologicalSpace X]
    [TopologicalSpace Y] (C : Set X) (D : Set Y) [CWComplex C]
    (f : X ≃ₜ Y) (imf : f '' C = D) : CWComplex D := ⋯
```

```
variable {X : Type*} [t : TopologicalSpace X] [T2Space X] (C : Set X) [CWComplex
    C]

instance CWComplex_level (n : ℕ∞) : CWComplex (level C n) := ⋯
```

```
variable {D : Set X} [CWComplex D]

def CWComplex_disjointUnion (disjoint : Disjoint C D) :
    CWComplex (C ∪ D) := ⋯
```

Where `f : X ≃ₜ Y` is the statement that `f` is a homeomorphism. Note that the first and last constructions are not instances. This is because the typeclass inference has no way to know that it should look for the hypotheses `(f : X ≃ₜ Y) (imf : f '' C = D)` and `(disjoint : Disjoint C D)`, as this information is not contained in the statement. So labeling these as instances would not be helpful. But in `CWComplex_level` all the necessary information is contained in the statement `CWComplex (level C n)`, which means that this will work as an instance.

3.2.2 Subcomplexes

With the assumptions for the rest of the section

```
variable {X : Type*} [t : TopologicalSpace X] [T2Space X] {C : Set X}
  [CWComplex C]
```

this is our definition of a subcomplex:

```
class Subcomplex (C : Set X) [CWComplex C] (E : Set X) where
  I : Π n, Set (cell C n)
  closed : IsClosed E
  union : ⋃ (n : ℕ) (j : I n), openCell (C := C) n j = E
```

The `(C := C)` is used to specify which CW-complex we are talking about. We need to use this notation because we made `C` and implicit argument in the definition of `openCell`.

We chose this to be a class again so that when you want to talk about the CW-structure of a subcomplex you don't need to explicitly mention the subcomplex structure and that every subcomplex is a CW-complex.

But we still have one issue: We would like to have the instance

```
instance doesnotcompile (E : Set X) [subcomplex : Subcomplex C E] :
  CWComplex E := ⋯
```

which unfortunately does not work, as Lean has no way to infer `C` from `CWComplex E` and therefore does not know that it should be looking for the instance `[subcomplex : Subcomplex C E]`.

To remedy this we can define some notation that includes the variable `C`:

```
def Sub (E : Set X) (C : Set X) [CWComplex C] [Subcomplex C E] : Set X := E

scoped infixr:35 " ⇂ "  => Sub
```

The first line defines a synonym for `E` that in its context includes that `E` is a subcomplex of the CW-complex `C`. The second line then defines the actual notation.

We can now use this to make the instance from above work:

```
instance CWComplex_subcomplex (E : Set X) [subcomplex : Subcomplex C E] :
    CWComplex (E ⇂ C) := ⋯
```

When formalising the two alternative definitions of a subcomplex that we talked about in Section 1.2.4, we can relax the conditions of Lemma 1.2.8 because, by Lemma 3.1.1, every CW-complex is closed. This is our version of Lemma 1.2.8 in the formalisation:

```
def Subcomplex'' (C : Set X) [CWComplex C] (E : Set X) (I : Π n, Set (cell C n))
    (cw : CWComplex E)
    (union : ⋃ (n : ℕ) (j : I n), openCell (C := C) n j = E) : Subcomplex C E
    where
  I := I
  closed := cw.isClosed
  union := union
```

Note that this version does not require `E` to be a CW-complex with respect to cells determined by `C`.

With that we can prove the lemmas from Section 1.2.4. Here are two examples. They correspond to statements 1.2.13 and 1.2.15.

```
lemma cell_mem_finite_subcomplex (n : ℕ) (i : cell C n) :
    ∃ (E : Set X) (subE : Subcomplex C E), Finite (E ⇂ C) ∧ i ∈ subE.I n := ⋯

lemma compact_subset_finite_subcomplex {B : Set X} (compact : IsCompact B) :
  ∃ (E : Set X) (sub : Subcomplex C E), CWComplex.Finite (E ⇂ C) ∧ B ∩ C ⊆ E :=
    ⋯
```

3.2.3 Product of CW-complexes

Let us lastly take a look at the product.

We define k-spaces as follows:

```
class KSpace  (X : Type*) [TopologicalSpace X] where
  isOpen_iff A : IsOpen A ↔
  ∀ (B : Set X), IsCompact B → ∃ (C : Set X), IsOpen C ∧ A ∩ B = C ∩ B
```

When defining the k-ification, we need to be a little careful. We want it to be an instance derived from another instance defined on the same type. To tell the system what instance we are referring to, we define a type synonym and then the k-ification:

```
def kification (X : Type*) := X

instance instkification {X : Type*} [t : TopologicalSpace X] : TopologicalSpace
    (kification X) := ···
```

Sometimes it is convenient to use maps that go to or from the k-ification, we define them as bijections:

```
def tokification (X : Type*) : X ≃ kification X :=
  ⟨fun x ↦ x, fun x ↦ x, fun _ ↦ rfl, fun _ ↦ rfl⟩

def fromkification (X : Type*) : kification X ≃ X :=
  ⟨fun x ↦ x, fun x ↦ x, fun _ ↦ rfl, fun _ ↦ rfl⟩
```

Here are the statements of some of the lemmas in Section 1.2.5. They correspond to the statements 1.2.36 , 1.2.39 , 1.2.27 , 1.2.28 and 1.2.33 .

```
instance kspace_of_WeaklyLocallyCompactSpace {X : Type*}[TopologicalSpace X]
    [WeaklyLocallyCompactSpace X] :
  KSpace X := ···

instance kspace_of_SequentialSpace {X : Type*} [TopologicalSpace X]
    [SequentialSpace X]: KSpace X := ···

lemma isCompact_iff_isCompact_tokification_image {X : Type*} [TopologicalSpace X]
    (C : Set X) :
  IsCompact C ↔ IsCompact (tokification X '' C) := ···

instance kspace_kification {X : Type*} [TopologicalSpace X] :
  KSpace (kification X) := ···

lemma continuous_kification_of_continuousOn_compact {X Y : Type*}
    [tX : TopologicalSpace X] [tY : TopologicalSpace Y] (f : X → Y)
    (conton : ∀ (C : Set X), IsCompact C → ContinuousOn f C) :
  Continuous (X := kification X) (Y := kification Y) f := ···
```

When proving statements about different topologies on the same space, formalising in Lean can be very helpful to keep track of what topology needs to be considered when.

With that we can move on to the product. The assumptions for the rest of the section are:

```
variable {X : Type*} {Y : Type*} [t1 : TopologicalSpace X]
  [t2 : TopologicalSpace Y] [T2Space X] [T2Space Y] {C : Set X} {D : Set Y}
  [CWComplex C] [CWComplex D]
```

Since the proof of Theorem 1.2.45 is both long and technical it is useful to separate out some definitions and lemmas.

We first define the indexing sets in the product:

```
def prodcell (C : Set X) (D : Set Y) [CWComplex C] [CWComplex D] (n : ℕ) :=
    (Σ' (m : ℕ) (l : ℕ) (hml : m + l = n), cell C m × cell D l)
```

where the Σ′ is a sigma type that allows us to add properties, in this case `hml`.

When discussing the product in Section 1.2.5, we always identified $\mathbb{R}^{m+n}$ and $\mathbb{R}^m \times \mathbb{R}^n$. When formalising we need to make that identification explicit:

```
def prodisometryequiv {n m l : ℕ}  (hmln : m + l = n) (j : cell C m) (k : cell D
    l) : (Fin n → ℝ) ≃ᵢ (Fin m → ℝ) × (Fin l → ℝ) := ⋯
```

where $\simeq_i$ is a symbol used to denote a bijective isometry. Using this map, we can define the characteristic maps of the product:

```
def prodmap {n m l : ℕ} (hmln : m + l = n) (j : cell C m) (k : cell D l) :
    PartialEquiv (Fin n → ℝ) (X × Y) := ⋯
```

After some lemmas about these maps such as for example

```
lemma prodmap_image_sphere {n m l : ℕ} {hmln : m + l = n} {j : cell C m} {k :
    cell D l} :
    prodmap hmln j k '' sphere 0 1 = (cellFrontier m j) ×ˢ (closedCell l k) ∪
    (closedCell m j) ×ˢ (cellFrontier l k) := ⋯
```

we can move on to defining the product:

```
instance CWComplex_product_kification : CWComplex (X := kification (X × Y))
  (C ×ˢ D) := ⋯
```

where $\times^s$ is the product of sets. Finally we define a version of this instance for when the product is already a k-space as to not unnecessarily apply the k-ification:

```
instance CWComplex_product [KSpace (X × Y)] : CWComplex (C ×ˢ D) := ⋯
```

Conclusion

The aim of this thesis was to formalise the basic properties and some constructions of CW-complexes, a concept that is not yet in Lean's the mathematical library mathlib.

We chose the historical definition to formalise the concept of CW-complexes in Lean:

```
class CWComplex.{u} {X : Type u} [TopologicalSpace X] (C : Set X) where
  cell (n : ℕ) : Type u
  map (n : ℕ) (i : cell n) : PartialEquiv (Fin n → ℝ) X
  source_eq (n : ℕ) (i : cell n) : (map n i).source = ball 0 1
  cont (n : ℕ) (i : cell n) : ContinuousOn (map n i) (closedBall 0 1)
  cont_symm (n : ℕ) (i : cell n) : ContinuousOn (map n i).symm (map n i).target
  pairwiseDisjoint' :
  (univ : Set (Σ n, cell n)).PairwiseDisjoint (fun ni ↦ map ni.1 ni.2 '' ball 0
    1)
  mapsto (n : ℕ) (i : cell n) : ∃ I : Π m, Finset (cell m),
    MapsTo (map n i) (sphere 0 1) (⋃ (m < n) (j ∈ I m), map m j '' closedBall 0 1)
  closed' (A : Set X) (asubc : A ⊆ C) : IsClosed A ↔ ∀ n j, IsClosed (A ∩ map n
    j '' closedBall 0 1)
  union' : ⋃ (n : ℕ) (j : cell n), map n j '' closedBall 0 1 = C
```

One of the important properties that we were able to formalise is the relationship between compact sets and finite CW-complexes:

```
lemma compact_iff_finite : IsCompact C ↔ Finite C := ⋯

lemma compact_subset_finite_subcomplex {B : Set X} (compact : IsCompact B) :
    ∃ (E : Set X) (sub : Subcomplex C E), CWComplex.Finite (E ⇂ C) ∧ B ∩ C ⊆ E :=
  ⋯
```

Additionally we formalised the CW-complex structure on the k-ification of the product of two CW-complexes of which the more readable mathematical statement is the following:

Theorem. *Let X, Y be CW-complexes with families of characteristic maps $(Q_i^n\colon D_i^n \to X)_{n\in\mathbb{N}, i\in I_n}$ and $(P_j^m\colon D_j^m \to Y)_{m\in\mathbb{N}, j\in J_n}$. Let e_i^n be the cells of X and f_j^m be the cells of Y. Then there is a CW-structure on $(X \times Y)_c$ with characteristic maps $(Q_i^n \times P_j^m\colon D_i^n \times D_j^m \to (X \times Y)_c)_{n,m\in\mathbb{N}, i\in I_n, j\in J_m}$. The indexing sets $(K_l)_{l\in\mathbb{N}}$ are given by $K_l = \bigcup_{n+m=l} I_n \times J_m$ for every $l \in \mathbb{N}$ and the cells are therefore of the form $e_i^n \times f_j^m$ for $n, m \in \mathbb{N}$, $i \in I_n$ and $j \in J_m$.*

Ultimately the goal is to add this work into mathlib, so that others can build upon it. I have already started to contribute some of the auxiliary lemmas unrelated to CW-complexes that were needed along the way. There is still much that can be done: First of all the definition could be generalized to relative CW-complexes and one could implement the modern definition as well. There are still some constructions that could be useful such as the quotient of a CW-complex by a subcomplex. More high-level goals could be the Whitehead Theorem or cellular homology and cohomology.

German summary

Diese Arbeit befasst sich mit der Formalisierung von CW-Komplexen im Beweisassistenten Lean. Beweisassistenten können dazu genutzt werden, formal die Richtigkeit von Beweisen in einem logischen digitalen System zu überprüfen. Lean ist unter anderem wegen seiner umfangreichen mathematischen Bibliothek *mathlib* ein sehr beliebter Beweisassistent. Ein Konzept, das in dieser Bibliothek jedoch noch fehlt, sind die CW-Komplexe. In der Topologie sind sie häufig hilfreich, um Berechnungen, zum Beispiel von Homologie und Kohomologie, zu vereinfachen.

Im ersten Kapitel beschäftigen wir uns mit der mathematischen Theorie hinter den CW-Komplexen. Wir konzentrieren uns hierbei auf die historische und nicht die moderne Definition, da uns diese die Formalisierung erleichtert. Wir beweisen einige grundlegende Eigenschaften von CW-Komplexen und beschäftigen uns dann im Detail mit verschiedenen Konstruktionen. Besonders dem Produkt zweier CW-Komplexes widmen wir sehr viel Zeit: Wir zeigen an einem Gegenbeispiel, dass das Produkt nicht notwendigerweise wieder ein CW-Komplex sein muss, führen dann k-Räume ein und beweisen, dass die k-ifizierung eines Produktes von zwei CW-Komplexen immer ein CW-Komplex ist.

Im zweiten Kapitel behandeln wir kurz drei technische Details von Lean: Die Typentheorie, d.h. die zugrundeliegende Logik, von Lean, implizite Argumente und Typklasseninferenz. Diese Inhalte sind interessante Zusatzinformation, aber nicht unbedingt notwendig für das Verständnis der Arbeit.

Im dritten Kapitel beschäftigen wir uns dann mit der Formalisierung von CW-Komplexen in Lean. Wir besprechen, welche Designentscheidungen getroffen wurden und warum, und machen auf Unterschiede in der Formalisierung aufmerksam. Dabei zeigen und erklären wir Ausschnitte des Codes. Wir haben einen Großteil des Inhalts des ersten Kapitels in Lean formalisiert, unter anderem den Zusammenhang zwischen endlichen Unterkomplexen und kompakten Mengen und die CW-Komplex-Struktur auf der k-ifizierung des Produktes von CW-Komplexen. Den kompletten Code findet man unter `https://github.com/scholzhannah/CWComplexes`.

Symbol Index

D^n	The closed unit disk in $\mathbb{R}^n$, i.e. $D^n \coloneqq \{x \in \mathbb{R}^n \mid \|x\| \leq 1\}$.
S^n	The boundary of the unit disk in $\mathbb{R}^n$, i.e. $S^n \coloneqq \{x \in \mathbb{R}^n \mid \|x\| = 1\}$.
∂e^n	The frontier of an n-cell, i.e. $\partial e^n \coloneqq Q^n(\partial D^n)$. See Definition 1.1.2.
$\overline{e}^n$	A closed n-cell, i.e. $\overline{e}^n \coloneqq Q^n(D^n)$. See Definition 1.1.2.
e^n	An (open) n-cell, i.e. $e^n \coloneqq Q^n(\operatorname{int}(D^n))$. See Definition 1.1.2.

Bibliography

[Avi+24] Jeremy Avigad et al. *Theorem Proving in Lean 4*. Accessed: 23.07.2024. https://leanprover.github.io/theorem_proving_in_lean4/, 2024.

[Bec+24] Lars Becker et al. *Carleson Operators on Doubling Metric Measure Spaces*. 2024. arXiv: 2405.06423 [math.CA]. URL: https://arxiv.org/abs/2405.06423.

[Bou66] Nicolas Bourbaki. *General topology 1*. Paris: Hermann [u.a.], 1966.

[Buz24] Kevin Buzzard. *The Fermat's Last Theorem Project*. Lean community blog. Accessed: 12.08.2024. 2024. URL: https://leanprover-community.github.io/blog/posts/FLT-announcement/.

[Car19] Mario Carneiro. "The Type Theory of Lean". MA thesis. Carnegie Mellon University, 2019.

[Com22] Johan Commelin. "Liquid Tensor Experiment". ger. In: *Mitteilungen der Deutschen Mathematiker-Vereinigung* 30.3 (2022), pp. 166–170. DOI: doi:10.1515/dmvm-2022-0058. URL: https://doi.org/10.1515/dmvm-2022-0058.

[Coq86] Thierry Coquand. "An analysis of Girard's paradox". PhD thesis. INRIA, 1986.

[DMN23] Floris van Doorn, Patrick Massot, and Oliver Nash. "Formalising the h-Principle and Sphere Eversion". In: *Proceedings of the 12th ACM SIGPLAN International Conference on Certified Programs and Proofs*. CPP 2023. Boston, MA, USA: Association for Computing Machinery, 2023, pp. 121–134. ISBN: 9798400700262. DOI: 10.1145/3573105.3575688. URL: https://doi.org/10.1145/3573105.3575688.

[Dow52] C. H. Dowker. "Topology of Metric Complexes". In: *American journal of mathematics* 74.3 (1952), pp. 555–577. ISSN: 0002-9327.

[FRO24] Lean FRO. *Lean Focused Research Organization website*. https://lean-fro.org/. Accessed: 12.08.2024. 2024.

[Gon+13] Georges Gonthier et al. "A Machine-Checked Proof of the Odd Order Theorem". In: *Interactive Theorem Proving*. Ed. by Sandrine Blazy, Christine Paulin-Mohring, and David Pichardie. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 163–179. ISBN: 978-3-642-39634-2.

[Hat01] Allen Hatcher. *Algebraic topology*. 16. printing 2016. Cambridge: Cambridge Univ. Press, 2001. ISBN: 9780521795401.

[Jän01] Klaus Jänich. *Topologie*. ger. Siebente Auflage. Springer-Lehrbuch. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001. ISBN: 9783540412847.

[KLN04] Fairouz Dib Kamareddine, Twan Laan, and Rob P. Nederpelt. *A modern perspective on type theory from its origins until today*. Applied logic series 29. Dordrecht u.a: Kluwer Acad. Publ., 2004. ISBN: 1402023340.

[LF23] Jannis Limperg and Asta Halkjær From. "Aesop: White-Box Best-First Proof Search for Lean". In: *Proceedings of the 12th ACM SIGPLAN International Conference on Certified Programs and Proofs*. CPP 2023. Boston, MA, USA: Association for Computing Machinery, 2023, pp. 253–266. ISBN: 9798400700262. DOI: 10.1145/3573105.3575671. URL: https://doi.org/10.1145/3573105.3575671.

[Lüc05] Wolfgang Lück. *Algebraische Topologie Homologie und Mannigfaltigkeiten*. ger. 1. Aufl. Vieweg Studium : Aufbaukurs Mathematik. Wiesbaden: Vieweg, 2005. ISBN: 3528032189.

[LW69] Albert T. Lundell and Stephen Weingram. *The topology of CW complexes*. «The» university series in higher mathematics. New York u.a: van Nostrand Reinhold, 1969.

[MS84] Per Martin-Löf and Giovanni Sambin. *Intuitionistic type theory*. Vol. 9. Bibliopolis Naples, 1984.

[MU21] Leonardo de Moura and Sebastian Ullrich. "The Lean 4 Theorem Prover and Programming Language". In: *Automated Deduction – CADE 28*. Ed. by André Platzer and Geoff Sutcliffe. Cham: Springer International Publishing, 2021, pp. 625–635. ISBN: 978-3-030-79876-5.

[Mun14] James Raymond Munkres. *Topology*. 2. ed., Pearson new internat. ed. Harlow: Pearson, 2014. ISBN: 9781292023625.

[PiB24] The pi-Base Community. *Search for ' Generated by compact subspaces'*. π-base. Accessed: 25.07.2024. 2024. URL: https://topology.pi-base.org/spaces?q=%7EGenerated+by+compact+subspaces.

[Rus08] Bertrand Russell. "Mathematical Logic as Based on the Theory of Types". In: *American journal of mathematics* 30.3 (1908), pp. 222–262. ISSN: 0002-9327.

[Sco16] Brian M. Scott (https://math.stackexchange.com/users/12042/brian-m-scott). *A first countable Hausdorff space is compactly generated*. Mathematics Stack Exchange. Version: 24.11.2016. 2016. URL: https://math.stackexchange.com/q/2026250.

[Uni13] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study: https://homotopytypetheory.org/book, 2013.

[WB89] P. Wadler and S. Blott. "How to make ad-hoc polymorphism less ad hoc". In: *Annual Symposium on Principles of Programming Languages: Proceedings of the 16th ACM SIGPLAN-SIGACT symposium on Principles of programming languages; 11-13 Jan. 1989*. New York, NY, USA: ACM, 1989, pp. 60–76. ISBN: 0897912942.

[Whi18] J. H. C. Whitehead. "Combinatorial homotopy. I". In: *Bulletin (new series) of the American Mathematical Society* 55.3 (2018), pp. 213–245. ISSN: 0273-0979.

[Wik24] Wikipedia contributors. *Compactly generated space*. Wikipedia, The Free Encyclopedia. Accessed: 23.07.2024. 2024. URL: https://en.wikipedia.org/w/index.php?title=Compactly_generated_space&oldid=1233649259.