Formalizing CW complexes in Lean

Hannah Scholz

Mathematical Institute of the University of Bonn

14.04.2024

Definition of CW-complexes

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.

Project Overview

The current progress in terms of files/topics looks like this:

- Definition and basic properties: In mathlib
- Finiteness notions on CW Complexes: PR
- Subcomplexes: Done
- Basic constructions: Done
- KSpaces: PR
- Additional Lemmas: Done
- Product: Done
- Examples: Needs clean-up
- Maps: Work in progress
- Quotients: To be done (by me)
- Equivalence to other definition: To be done (probably not by me)

March 2024

```
structure 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 = closedBall 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) : 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
```

April 2024

```
structure 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 = closedBall 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
```

July 2024

```
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 = closedBall 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
```

July 2024: Reason for change

```
def finite_subcomplex_finite_iUnion_finite_subcomplex
  (J : Type*) [_root_.Finite J] (sub : J → Set X)
  (cw : ∀ (j : J), hC.Subcomplex (sub j))
  (finite : ∀ (j : J), (hC.CWComplex_subcomplex _ (cw j)).Finite) :
  (hC.CWComplex_subcomplex _
    (hC.subcomplex_iUnion_subcomplex J sub cw)).Finite := ...
```

October 2024

```
class RelCWComplex.{u} {X : Type u} [TopologicalSpace X]
    (C D : 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 = closedBall 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)
    (D ∪ ⋃ (m < n) (j ∈ I m), map m j '' closedBall 0 1)
  closed' (A : Set X) (asubc : A ⊆ C) :
    ((∀ n j, IsClosed (A ∩ map n j '' closedBall 0 1)) ∧
    IsClosed (A ∩ D)) → IsClosed A
  union' : D ∪ ⋃ (n : ℕ) (j : cell n), map n j '' closedBall 0 1 = C
```

November 2024

```
class RelCWComplex.{u} {X : Type u} [TopologicalSpace X]
    (C D : 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 = closedBall 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)
  disjointBase' (n : ℕ) (i : cell n) : Disjoint (map n i '' ball 0 1) D
  mapsto (n : ℕ) (i : cell n) : ∃ I : Π m, Finset (cell m),
    MapsTo (map n i) (sphere 0 1)
    (D ∪ ⋃ (m < n) (j ∈ I m), map m j '' closedBall 0 1)
  closed' (A : Set X) (asubc : A ⊆ C) :
    ((∀ n j, IsClosed (A ∩ map n j '' closedBall 0 1)) ∧
    IsClosed (A ∩ D)) → IsClosed A
  isClosedBase : IsClosed D
  union' : D ∪ ⋃ (n : ℕ) (j : cell n), map n j '' closedBall 0 1 = C
```

November 2024

```
class RelCWComplex.{u} {X : Type u} [TopologicalSpace X] (C : Set X)
    (D : outParam (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 = closedBall 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)
  disjointBase' (n : ℕ) (i : cell n) : Disjoint (map n i '' ball 0 1) D
  mapsto (n : ℕ) (i : cell n) : ∃ I : Π m, Finset (cell m),
    MapsTo (map n i) (sphere 0 1)
    (D ∪ ⋃ (m < n) (j ∈ I m), map m j '' closedBall 0 1)
  closed' (A : Set X) (asubc : A ⊆ C) :
    ((∀ n j, IsClosed (A ∩ map n j '' closedBall 0 1)) ∧
    IsClosed (A ∩ D)) → IsClosed A
  isClosedBase : IsClosed D
  union' : D ∪ ⋃ (n : ℕ) (j : cell n), map n j '' closedBall 0 1 = C
```

November 2024: Reason for change

```
instance RelCWComplex_levelaux [RelCWComplex C D] (n : ℕ∞) :
    RelCWComplex (levelaux C D n) D where
  cell l := {x : cell C D l // l < n}
  map l i := map (C := C) (D := D) l i
  source_eq l i:= source_eq (C := C) (D := D) l i
  cont l i := cont (C := C) (D := D) l i
  cont_symm l i := cont_symm (C := C) (D := D) l i
  ...
```

January 2025

```
class RelCWComplex.{u} {X : Type u} [TopologicalSpace X] (C : Set X)
    (D : outParam (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
  continuousOn (n : ℕ) (i : cell n) :
    ContinuousOn (map n i) (closedBall 0 1)
  continuousOn_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)
  disjointBase' (n : ℕ) (i : cell n) : Disjoint (map n i '' ball 0 1) D
  mapsto (n : ℕ) (i : cell n) : ∃ I : Π m, Finset (cell m),
    MapsTo (map n i) (sphere 0 1)
    (D ∪ ⋃ (m < n) (j ∈ I m), map m j '' closedBall 0 1)
  closed' (A : Set X) (asubc : A ⊆ C) : ((∀ n j, IsClosed (A ∩ map n j '
    ' closedBall 0 1)) ∧ IsClosed (A ∩ D)) → IsClosed A
  isClosedBase : IsClosed D
  union' : D ∪ ⋃ (n : ℕ) (j : cell n), map n j '' closedBall 0 1 = C
```

Subcomplexes: The mathematics

Definition: Subcomplex

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:

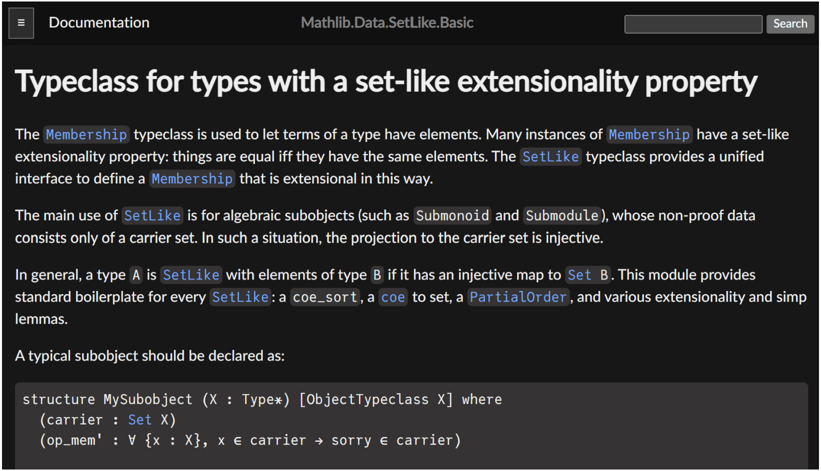
(i) E is closed.

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

Theorem: Subcomplexes are CW complexes

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.

What I should have done



Typeclass for types with a set-like extensionality property

The `Membership` typeclass is used to let terms of a type have elements. Many instances of `Membership` have a set-like extensionality property: things are equal iff they have the same elements. The `SetLike` typeclass provides a unified interface to define a `Membership` that is extensional in this way.

The main use of `SetLike` is for algebraic subobjects (such as `Submonoid` and `Submodule`), whose non-proof data consists only of a carrier set. In such a situation, the projection to the carrier set is injective.

In general, a type `A` is `SetLike` with elements of type `B` if it has an injective map to `Set B`. This module provides standard boilerplate for every `SetLike`: a `coe_sort`, a `coe` to set, a `PartialOrder`, and various extensionality and simp lemmas.

A typical subobject should be declared as:

```
structure MySubobject (X : Type*) [ObjectTypeclass X] where
  (carrier : Set X)
  (op_mem' : ∀ {x : X}, x ∈ carrier → sorry ∈ carrier)
```

What I actually did

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

The problem

```
instance CWComplex.instEmpty : CWComplex (∅ : Set X) := ...
```