

Formal Topology in Univalent Foundations

Ayberk Tosun

February 27, 2020

Chalmers University of Technology

Topology

understood \Downarrow constructively

Pointless topology

understood \Downarrow predicatively

Formal topology

What locales are like

- Abstraction of open sets of a topology.
- Logic of “observable properties”.
- CS view: logic of “semidecidable properties”.

What locales are like

- Abstraction of open sets of a topology.
- Logic of “observable properties”.
- CS view: logic of “semidecidable properties”.
- “Junior-grade topos theory”.

A poset \mathcal{O} such that

- finite subsets of \mathcal{O} have meets,
- all subsets of \mathcal{O} have joins, and
- binary meets distribute over arbitrary joins:

$$a \wedge \left(\bigvee_{i \in I} b_i \right) = \bigvee_{i \in I} (a \wedge b_i),$$

for any $a \in \mathcal{O}$ and I -indexed family b over \mathcal{O} .

Locales of downward-closed subsets

Given a poset

$$A : \text{Type}_m$$

$$\sqsubseteq : A \rightarrow A \rightarrow \text{hProp}_m$$

the type of **downward-closed subsets** of A is:

$$\sum_{(U : \mathcal{P}(A))} \prod_{(x \sqsubseteq y : A)} x \in U \rightarrow y \sqsubseteq x \rightarrow y \in U$$

where

$$\mathcal{P} : \text{Type}_m \rightarrow \text{Type}_{m+1}$$

$$\mathcal{P}(A) :\equiv A \rightarrow \text{hProp}_m$$

This forms a **frame**:

$$\begin{aligned}\top &:\equiv \lambda_ . 1 \\ A \wedge B &:\equiv \lambda x. (x \in A) \times (x \in B) \\ \bigvee_{i: I} \mathbf{B}_i &:\equiv \lambda x. \left\| \sum_{(i: I)} x \in \mathbf{B}_i \right\|\end{aligned}$$

Nuclei for frames

Question: can we get all frames out of posets in this way?

One way is to employ the notion of a **nucleus**.

Let F be a frame. A **nucleus** on F is an endofunction $\mathbf{j} : |F| \rightarrow |F|$ such that

$$(1) \quad \prod_{(x : A)} x \sqsubseteq \mathbf{j}(x) \quad [\text{extensiveness}],$$

$$(2) \quad \prod_{(x, y : A)} \mathbf{j}(x \wedge y) = \mathbf{j}(x) \wedge \mathbf{j}(y) \quad [\text{meet preservation}], \text{ and}$$

$$(3) \quad \prod_{(x : A)} \mathbf{j}(\mathbf{j}(x)) \sqsubseteq \mathbf{j}(x) \quad [\text{idempotence}].$$

Closure operators

In the particular case where F is the locale of downward-closed subsets for a poset $A : \text{Type}_m$, the nucleus can be seen as a **closure operator**—if it can be shown to be **propositional**.

$$\blacktriangleright \quad : \quad \underbrace{\mathcal{P}(A) \rightarrow \mathcal{P}(A)}$$

This is what we want.

Closure operators

In the particular case where F is the locale of downward-closed subsets for a poset $A : \text{Type}_m$, the nucleus can be seen as a **closure operator**—if it can be shown to be **propositional**.

$$\blacktriangleright \quad : \quad \underbrace{\mathcal{P}(A) \rightarrow A \rightarrow \text{hProp}_m}_{\text{This is what we want.}}$$

Closure operators

In the particular case where F is the locale of downward-closed subsets for a poset $A : \text{Type}_m$, the nucleus can be seen as a **closure operator**—if it can be shown to be **propositional**.

$$\blacktriangleright \quad : \quad \underbrace{\mathcal{P}(A) \rightarrow A \rightarrow \text{Type}_m}_{\text{This is what we have.}}$$

Baire space $(\mathbb{N} \rightarrow \mathbb{N})$

```
data  $\mathbb{D}$  :  $\text{Type}_0$  where
```

```
  [] :  $\mathbb{D}$ 
```

```
   $\_ \frown \_$  :  $\mathbb{D} \rightarrow \mathbb{N} \rightarrow \mathbb{D}$ 
```

```
IsDC :  $(\mathbb{D} \rightarrow \text{Type}_0) \rightarrow \text{Type}_0$ 
```

```
IsDC P =  $(\sigma : \mathbb{D}) (n : \mathbb{N}) \rightarrow P \sigma \rightarrow P (\sigma \frown n)$ 
```

Baire space $(\mathbb{N} \rightarrow \mathbb{N})$

```
data _◀_ (σ :  $\mathbb{D}$ ) (P :  $\mathbb{D} \rightarrow \text{Type}_0$ ) :  $\text{Type}_0$  where
  dir      : P σ  $\rightarrow$  σ ◀ P
  branch   : ((n :  $\mathbb{N}$ )  $\rightarrow$  (σ  $\frown$  n) ◀ P)  $\rightarrow$  σ ◀ P
  squash   : (φ ψ : σ ◀ P)  $\rightarrow$  φ  $\equiv$  ψ
```

We can now show that this defines a nucleus, without choice!

Baire space $(\mathbb{N} \rightarrow \mathbb{N})$

Using the following, and then *truncating from the outside* does not work.

```
data _◀★_ (σ :  $\mathbb{D}$ ) (P :  $\mathbb{D} \rightarrow \text{Type}_0$ ) :  $\text{Type}_0$  where
  dir      : P σ → σ ◀★ P
  branch   : ((n :  $\mathbb{N}$ ) → (σ  $\frown$  n) ◀★ P) → σ ◀★ P
  -- squash : (φ ψ : σ ◀ P) → φ ≡ ψ
```

Baire space $(\mathbb{N} \rightarrow \mathbb{N})$

We can now prove the following idempotence law, without using countable choice $(\prod_{(i : \mathbb{N})} \|B_i\| \rightarrow \|\prod_{(i : \mathbb{N})} B_i\|)$.

$\delta : \sigma \triangleleft P \rightarrow ((v : \mathbb{D}) \rightarrow P \ v \rightarrow v \triangleleft Q) \rightarrow \sigma \triangleleft Q$

$\delta \text{ (dir } u \in P) \quad \varphi = \varphi _ u \in P$

$\delta \text{ (branch } f) \quad \varphi = \text{branch } (\lambda n \rightarrow \delta \text{ (f } n) \varphi)$

$\delta \text{ (squash } u \triangleleft P_0 \ u \triangleleft P_1 \ i) \ \varphi = \text{squash } (\delta \ u \triangleleft P_0 \ \varphi) \ (\delta \ u \triangleleft P_1 \ \varphi) \ i$

$\text{idempotence} : \sigma \triangleleft (\lambda _ \rightarrow _ \triangleleft P) \rightarrow \sigma \triangleleft P$

$\text{idempotence } u \triangleleft \triangleleft P = \delta \ u \triangleleft \triangleleft P \ (\lambda _ v \triangleleft P \rightarrow v \triangleleft P)$

Baire space ($\mathbb{N} \rightarrow \mathbb{N}$)

— ζ inference à la Brouwer.

$\zeta : (n : \mathbb{N}) \rightarrow \text{IsDC } P \rightarrow \sigma \blacktriangleleft P \rightarrow (\sigma \frown n) \blacktriangleleft P$

$\zeta \ n \ dc \ (\text{dir } \sigma \varepsilon P) = \text{dir } (dc \ _ \ n \ \sigma \varepsilon P)$

$\zeta \ n \ dc \ (\text{branch } f) = \text{branch } \lambda \ m \rightarrow \zeta \ m \ dc \ (f \ n)$

$\zeta \ n \ dc \ (\text{squash } \sigma \blacktriangleleft P \ \sigma \blacktriangleleft P' \ i) = \text{squash } (\zeta \ n \ dc \ \sigma \blacktriangleleft P) (\zeta \ n \ dc \ \sigma \blacktriangleleft P') \ i$

$\zeta' : \text{IsDC } P \rightarrow \text{IsDC } (\lambda \ - \rightarrow - \blacktriangleleft P)$

$\zeta' \ P\text{-dc } \sigma \ n \ \sigma \blacktriangleleft P = \zeta \ n \ P\text{-dc } \sigma \blacktriangleleft P$

Baire space ($\mathbb{N} \rightarrow \mathbb{N}$)

`lemma : IsDC P → P σ → σ ◀ Q → σ ◀ (λ - → P - × Q -)`

`lemma dc σεP (dir σεQ) = dir (σσεP , σσεQ)`

`lemma dc σεP (branch f) = branch (λ n → lemma dc (dc _ n σσεP) (f n))`

`lemma dc σεP (squash σ◀Q σ◀Q' i) = squash (lemma dc σεP σ◀Q) (lemm`

`mp : IsDC P → IsDC Q → σ ◀ P → σ ◀ Q → σ ◀ (λ - → P - × Q -)`

`mp P-dc Q-dc (dir σσεP) σ◀Q = lemma P-dc σσεP σ◀Q`

`mp P-dc Q-dc (branch f) σ◀Q = branch (λ n → mp P-dc Q-dc (f n) (ζ n`

`mp P-dc Q-dc (squash σ◀P σ◀P' i) σ◀Q = squash (mp P-dc Q-dc σ◀P σ◀`

Baire space ($\mathbb{N} \rightarrow \mathbb{N}$)

This example can be accessed at:

<https://ayberkt.gitlab.io/msc-thesis/BaireSpace.html>