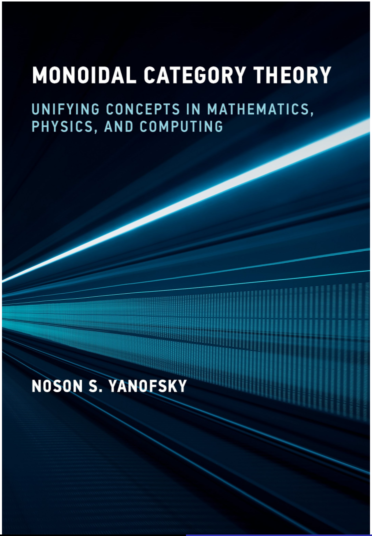
Monoidal Category Theory
Unifying Concepts in Mathematics, Physics, and Computing



Chapter 6:

Relationships Between

Monoidal Categories

Foreshadowing

- Chapter 6: Relationships Between Monoidal Categories
  - Section 6.1: Monoidal Functors and Natural Transformations
  - Section 6.2: Coherence Theorems
  - Section 6.3: When Coherence Fails
  - Section 6.4: Mini-course: Duality Theory

Foreshadowing

- In this chapter we show how monoidal categories are related to each other.
- In context, Chapter 4 taught how categories are related with functors,
- Chapter 5 showed that some of these categories have monoidal structures.
- Now we show that there are functors that describe the relationships between categories which also respect the monoidal structures.
- These functors highlight how the categories have shared properties. This is very important for our unification goal.
- We also prove important theorems relating monoidal categories to strict monoidal categories.

Foreshadowing

- Chapter 6: Relationships Between Monoidal Categories
  - Section 6.1: Monoidal Functors and Natural Transformations
    - Monoidal Functors
    - Symmetric Monoidal Functors
    - Monoidal Natural Transformations
    - Monoidal Equivalence
    - Examples

Monoidal Functors

- We have introduced monoidal categories.
- Now we will discuss how monoidal categories relate to each other.
- We saw that for a set function $f\colon M \longrightarrow M'$ to be a monoid homomorphism $f\colon (M, \star, e) \longrightarrow (M', \star', e')$ it must respect the operations:

$$f(x \star y) = f(x) \star' f(y) \text{ and } f(e) = e'.$$

- Here we describe a category theoretic versions of these conditions.

Monoidal Functors

Definition

Given monoidal categories $(\mathbb{A}, \otimes, \alpha, I, \lambda, \rho)$ *and* $(\mathbb{B}, \otimes', \alpha', I', \lambda', \rho')$, *a* **monoidal functor**
$(F, \tau, \upsilon)\colon (\mathbb{A}, \otimes, \alpha, I, \lambda, \rho) \longrightarrow (\mathbb{B}, \otimes', \alpha', I', \lambda', \rho')$ *is*

- *A functor* $F\colon \mathbb{A} \longrightarrow \mathbb{B}$.
- *A natural transformation called a* **mapping funnel**

$$\tau\colon \otimes' \circ (F \times F) \Longrightarrow F \circ \otimes.$$

That is, for all a and a′ in $\mathbb{A}$, *there is a morphism*

$$\tau_{a,a'}\colon F(a) \otimes' F(a') \longrightarrow F(a \otimes a')$$

which "funnels" all the elements into the parentheses.

Monoidal Functors

Definition

- *Let* $u\colon \mathbf{1} \longrightarrow \mathbb{A}$ *and* $u'\colon \mathbf{1} \longrightarrow \mathbb{B}$ *be the functors that pick out the units* I *and* I' *of* $\mathbb{A}$ *and* $\mathbb{B}$*, respectively. Then there is a natural transformation called a* **unital funnel**

$$\upsilon\colon u' \Longrightarrow F \circ u.$$

That is, a morphism in $\mathbb{B}$

$$\upsilon_*\colon I' \longrightarrow F(I).$$

Monoidal Functors

Definition

These natural transformations must satisfy the following coherence requirements:

- *The mapping funnel τ must cohere with itself and with the reassociators α and α' as in this* **hexagon coherence condition**

$$
\begin{array}{ccc}
Fa \otimes' (Fb \otimes' Fc) & \xrightarrow{\alpha'_{Fa,Fb,Fc}} & (Fa \otimes' Fb) \otimes' Fc \\
\Big\downarrow{\scriptstyle id_{Fa} \otimes' \tau_{b,c}} & & \Big\downarrow{\scriptstyle \tau_{a,b} \otimes' id_{Fc}} \\
Fa \otimes' F(b \otimes c) & & F(a \otimes b) \otimes' Fc \\
\Big\downarrow{\scriptstyle \tau_{a,b\otimes c}} & & \Big\downarrow{\scriptstyle \tau_{a\otimes b,c}} \\
F(a \otimes (b \otimes c)) & \xrightarrow[F(\alpha_{a,b,c})]{} & F((a \otimes b) \otimes c).
\end{array}
$$

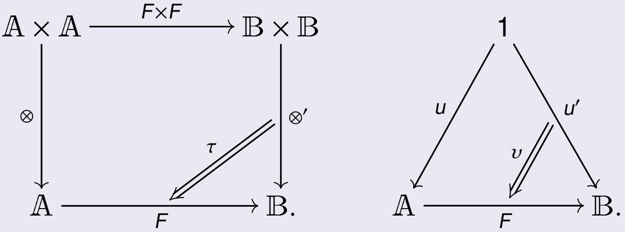
Definition

- *The mapping funnel τ and the unital funnel υ must cohere with the right and left unitors ρ and λ:*

$$
\begin{array}{ccc}
Fa \otimes' I' & \xrightarrow{id_{Fa} \otimes' \upsilon_*} & Fa \otimes' FI \\
\downarrow \rho'_{Fa} & & \downarrow \tau_{a,I} \\
Fa & \xleftarrow[F\rho_a]{} & F(a \otimes I)
\end{array}
\qquad
\begin{array}{ccc}
I' \otimes' Fa & \xrightarrow{\upsilon'_* \otimes' id_{Fa}} & FI \otimes' Fa \\
\downarrow \lambda'_{Fa} & & \downarrow \tau_{I,a} \\
Fa & \xleftarrow[F\lambda_a]{} & F(I \otimes a).
\end{array}
$$

Remark

The τ and υ natural transformations can be seen as a weakening of the commuting square definition of a monoid homomorphism. Rather than insisting that the diagrams commute, we insist that there are natural transformations from the composite of one side of the diagram to the other:



The τ and the υ are, in general, not isomorphisms.

Monoidal Functors

Remark

In the event that $\mathbb{A}$ and $\mathbb{B}$ are strict monoidal categories, then the hexagon coherence condition reduces to

$$
\begin{array}{ccccc}
 & & Fa \otimes' Fb \otimes' Fc & & \\
 & \swarrow^{id_{Fa} \otimes' \tau_{b,c}} & & \searrow^{\tau_{a,b} \otimes' id_{Fc}} & \\
Fa \otimes' F(b \otimes c) & & & & F(a \otimes b) \otimes' Fc \\
 & \searrow_{\tau_{a,b\otimes c}} & & \swarrow_{\tau_{a\otimes b,c}} & \\
 & & F(a \otimes b \otimes c). & &
\end{array}
$$

Monoidal Functors

Monoidal functors come in different flavors.

Definition

- *The above definition of a monoidal functor is also called a* **weak monoidal functor** *or a* **lax monoidal functor**.
- *If the τ and the υ natural transformations go the other way, that is, we have $\tau_{a,a'} \colon F(a \otimes a') \longrightarrow F(a) \otimes' F(a')$ and $\upsilon_* \colon F(I) \longrightarrow I'$ then F is called a* **oplax monoidal functor**.
- *If the τ and the υ are isomorphisms, then F is called a* **strong monoidal functor**.
- *If the τ and the υ are identity morphisms, then we have $F(a) \otimes' F(a') = F(a \otimes a')$ and $I' = F(I)$. Such an F is called a* **strict monoidal functor**.

Monoidal Functors

Every elementary school child knows the following example.

Example

We met the two strict monoidal categories $(\mathbf{R}, +, 0)$ *and* $(\mathbf{R}^+, \cdot, 1)$. *For every real number* $b > 1$, *there is a strict monoidal functor*

$$b^{(\ )}\colon (\mathbf{R}, +, 0) \longrightarrow (\mathbf{R}^+, \cdot, 1).$$

The fact that $b^{x+y} = b^x \cdot b^y$ *and* $b^0 = 1$ *means* $b^{(\ )}$ *strictly preserves the operations. There is also a strict monoidal functor*

$$Log_b(\ )\colon (\mathbf{R}^+, \cdot, 1) \longrightarrow (\mathbf{R}, +, 0).$$

The fact that $Log_b(x \cdot y) = Log_b(x) + Log_b(y)$ *and* $Log_b(1) = 0$ *means this functor strictly preserves the operations.*

Example

This brings to light a more general statement. If there are monoids M and M' that form strict monoidal categories, then a homomorphism from M to M' is a strict monoidal functor.

Monoidal Functors

Example

We met the free functor $F\colon \mathbb{S}\mathrm{et} \longrightarrow \mathbf{K}\mathbb{V}\mathrm{ect}$ *that takes a set* S *to* $F(S)$*, the vector space that has* S *as its basis. We also met the right adjoint to this functor,* $U\colon \mathbf{K}\mathbb{V}\mathrm{ect} \longrightarrow \mathbb{S}\mathrm{et}$*, that takes a vector space* V *to* $U(V)$*, its underlying set. Remember that the category of sets has two monoidal category structures:* $(\mathbb{S}\mathrm{et}, +, \emptyset)$ *and* $(\mathbb{S}\mathrm{et}, \times, \{*\})$*. There are also two monoidal structures on* $\mathbf{K}\mathbb{V}\mathrm{ect}$*. The adjunctions respect both monoidal structures as follows*

$$(\mathbb{S}\mathrm{et}, +, \emptyset) \underset{(U,\tau_1,\upsilon_1)}{\overset{(F,\tau,\upsilon)}{\rightleftarrows}} (\mathbf{K}\mathbb{V}\mathrm{ect}, \oplus, 0) \qquad \bot \qquad (\mathbb{S}\mathrm{et}, \times, \{*\}) \underset{(U',\tau'_1,\upsilon'_1)}{\overset{(F',\tau',\upsilon')}{\rightleftarrows}} (\mathbf{K}\mathbb{V}\mathrm{ect}, \otimes, \mathbf{K}) \cdot$$

Example (Continued.)

Let us examine each of these four functors and see how they respect the monoidal structures:

- $F(S + S')$ *is isomorphic to* $F(S) \oplus F(S')$*, and* $F(\emptyset) = 0$*. This is a strong monoidal functor.*
- *There is a map* $U(V) + U(V') \longrightarrow U(V + V')$*. This map is not an isomorphism because* $U(V) + U(V')$ *has two* 0*'s and* $U(V + V')$ *has only one. On the unit, there is* $U(0) = 0$*.*
- $F'(S \times S') \cong F'(S) \otimes F'(S')$ *and* $F'(\{*\}) = \mathbf{K}$*. This is a strong monoidal functor.*
- *There is a map* $U'(V) \times U'(V') \longrightarrow U'(V \otimes V')$ *because* $V \otimes V'$ *is defined by an equivalence relation on* $V \times V'$*. On the unit, there is* $U'(\mathbf{K}) = \mathbf{K} \neq \{*\}$*. This means* U' *is a lax monoidal functor.*

Monoidal Functors

Example (Continued.)

- *What about finite sets and finite dimensional vector spaces?*
- *Notice that although there is a free functor from* $\mathbb{FinSet}$ *to* **K**$\mathbb{FDVect}$*, the forgetful functor from finite dimensional vector spaces does not output finite sets.*
- *In other words, in general, the underlying set of a finite dimensional vector space is not a finite set.*

Monoidal Functors

Monoidal functors compose as follows. Given monoidal functor

$$(F, \tau, \upsilon) \colon (\mathbb{A}, \otimes, \alpha, I, \lambda, \rho) \longrightarrow (\mathbb{B}, \otimes', \alpha', I', \lambda', \rho')$$

and

$$(F', \tau', \upsilon') \colon (\mathbb{B}, \otimes', \alpha', I', \lambda', \rho') \longrightarrow (\mathbb{C}, \otimes'', \alpha'', I'', \lambda'', \rho''),$$

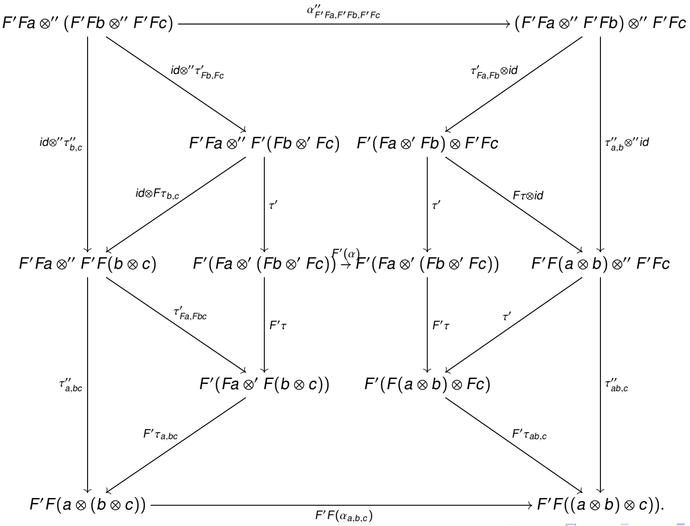
we form

$$(F' \circ F, \tau'', \iota'') \colon (\mathbb{A}, \otimes, \alpha, I, \lambda, \rho) \longrightarrow (\mathbb{C}, \otimes'', \alpha'', I'', \lambda'', \rho'').$$

The component of the mapping funnel τ'' at elements a and a' is defined as the composition of the two maps:

$$F'Fa \otimes'' F'Fa' \xrightarrow{\tau'_{Fa,Fa'}} F'(Fa \otimes' Fa') \xrightarrow{F'\tau_{a,a'}} F'F(a \otimes a'). \qquad \left(\text{composite: } \tau''_{a,a'}\right)$$

This satisfies the hexegon coherence condition because of the commutativity of the following diagram:



- The top hexagon commutes because τ' is a mapping funnel.
- The bottom hexagon commutes because τ is a mapping funnel and because of the functoriality of F'.
- The four triangles commute because of the definition of τ''.
- The left and right quadrilaterals commute because of the naturality of τ and τ'.

Since all the inner parts of the diagram commute, the outer hexagon of the diagram commutes. This ensures that the composed mapping funnel, τ'', satisfies coherence condition.

- With composition of monoidal functors, one can formulate the notion of an isomorphism made of monoidal functors.
- For example, the two strict monoidal functors here are inverse to each other and form a monoidal isomorphism.
- Another example of isomorphic monoidal categories are the two strict monoidal categories $(\mathbf{N}, +, 0)$ and $(\{1\}^*, \bullet, \emptyset)$. There is clearly a functor $n \mapsto 11 \cdots 1(n$ times). This functor preserves the tensor product and is an isomorphism.
- It should be noted that just as isomorphism of categories are a rarity, so too, isomorphism of monoidal categories are a rarity.

Symmetric Monoidal Functors

What about functors between symmetric monoidal categories?

Definition

Let $\mathbb{A}$ and $\mathbb{B}$ be symmetric monoidal categories with braidings γ and γ', respectively. A monoidal functor (F, τ, υ) from $\mathbb{A}$ to $\mathbb{B}$ is a **symmetric monoidal functor** *if the mapping funnels and the braidings cohere with each other as follows:*

$$\begin{array}{ccc}
Fa \otimes' Fa' & \xrightarrow{\gamma'_{Fa,Fa'}} & Fa' \otimes' Fa \\
\Big\downarrow{\scriptstyle \tau_{a,a'}} & & \Big\downarrow{\scriptstyle \tau_{a',a}} \\
F(a \otimes a') & \xrightarrow[F\gamma_{a,a'}]{} & F(a' \otimes a).
\end{array}$$

A Category Defined

- *The collection of monoidal categories and strong monoidal functors form the category* $\mathbb{MonCat}$.
- *There is a subcategory of strict monoidal categories and strong monoidal functors denoted* $\mathbb{StrMonCat}$.
- *The collection of symmetric monoidal categories and symmetric monoidal functors form a category* $\mathbb{SymMonCat}$.
- *There is a subcategory of strictly associative symmetric monoidal categories and symmetric monoidal functors between them denoted* $\mathbb{StrSymMonCat}$.
- *There are forgetful functors from the collections of symmetric monoidal categories to the collections of monoidal categories.*

A Category Defined (Continued.)

These categories are related as follows:

$$
\begin{array}{ccc}
\mathbb{SymMonCat} & \xrightarrow{\;U\;} & \mathbb{MonCat} \\
\uparrow & & \uparrow \\
\mathbb{StrSymMonCat} & \xrightarrow[\;U\;]{} & \mathbb{StrMonCat}.
\end{array}
$$

Monoidal Natural Transformations

Let us go up one level and define a natural transformation between monoidal functors.

Definition

Let $\mathbb{A}$ and $\mathbb{B}$ be symmetric monoidal categories and let (F, τ, υ) and (F', τ', υ') be monoidal functors from $\mathbb{A}$ to $\mathbb{B}$. A **monoidal natural transformation** *from (F, τ, υ) to (F', τ', υ') is a natural transformation $\mu \colon F \Longrightarrow F'$, i.e., for every a in $\mathbb{A}$, a morphism $\mu_a \colon F(a) \longrightarrow F'(a)$ which coheres with the mapping funnel and unital functors as follows:*

$$\begin{array}{ccc} F(a) \otimes' F(a') & \xrightarrow{\mu_a \otimes' \mu_{a'}} & F'(a) \otimes' F'(a') \\ \Big\downarrow{\scriptstyle \tau_{a,a'}} & & \Big\downarrow{\scriptstyle \tau'_{a,a'}} \\ F(a \otimes a') & \xrightarrow[\mu_{a \otimes a'}]{} & F'(a \otimes a') \end{array} \qquad \begin{array}{ccc} & I' & \\ {\scriptstyle \upsilon}\swarrow & & \searrow{\scriptstyle \upsilon'} \\ F(I) & \xrightarrow[\mu_I]{} & F'(I). \end{array}$$

A **symmetric monoidal natural transformation** *is a monoidal transformation between symmetric monoidal functors.*

Monoidal Natural Transformations

A Category Defined

- *The collection of monoidal categories, strong monoidal functors, and monoidal natural transformations forms a 2-category* $\overline{\mathbb{MonCat}}$.
- *There is a sub-2-category of strict monoidal categories, strong monoidal functors, and monoidal tranformations denoted* $\overline{\mathbb{StrMonCat}}$.
- *There are similar statements about the 2-categories of symmetric monoidal categories* $\overline{\mathbb{SymMonCat}}$ *and strictly associative symmetric monoidal categories* $\overline{\mathbb{StrSymMonCat}}$.

Symmetric Monoidal Functors

A Category Defined (Continued.)

These 2-categories fit together as follows:

$$
\begin{array}{ccc}
\overline{\mathbb{SymMonCat}} & \xrightarrow{\quad U \quad} & \overline{\mathbb{MonCat}} \\
\uparrow & & \uparrow \\
\overline{\mathbb{StrSymMonCat}} & \xrightarrow[\quad U \quad]{} & \overline{\mathbb{StrMonCat}}.
\end{array}
$$

Monoidal Natural Transformations

Given the notion of a symmetric monoidal natural transformation, we can easily describe what it means for two symmetric monoidal categories to be equivalent.

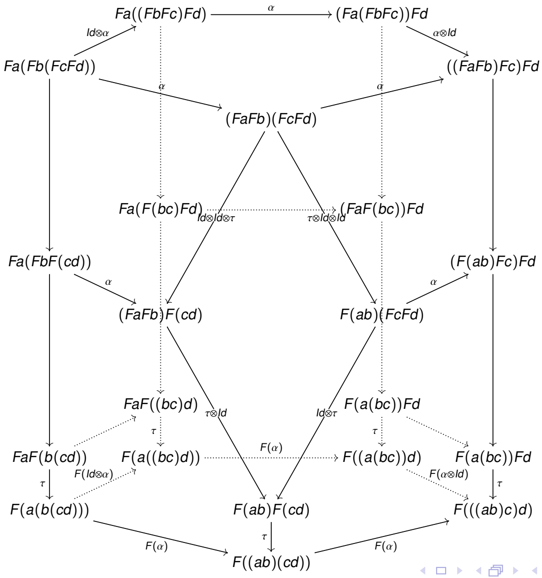
Definition

A **monoidal equivalence** *between* $(\mathbb{A}, \otimes, \alpha, I, \lambda, \rho)$ *and* $(\mathbb{A}', \otimes', \alpha', I', \lambda', \rho')$ *means that there is a monoidal functor* (F, τ, υ) *from* $\mathbb{A}$ *to* $\mathbb{A}'$ *and a monoidal functor* (F', τ', υ') *from* $\mathbb{A}'$ *to* $\mathbb{A}$ *with monoidal natural isomorphisms* $\mu\colon Id_{\mathbb{A}} \longrightarrow F' \circ F$ *and* $\mu'\colon F \circ F' \longrightarrow Id_{\mathbb{A}'}$. *An equivalence that uses strong monoidal functors has the property that the functor is full, faithful, and essentially surjective.*

Monoidal Functors

- We saw shows how two objects funnel into one.
- We also saw how three objects funnel into one.
- What about more objects?
- In the next slide, there is a diagram of four objects funneling into one.
- The slide after that has some needed orientation.

Monoidal Functors



Some orientation around this diagram is needed. All vertical maps are instances of τ's.

- The top is Mac Lane's pentagon condition.
- The bottom is the image of Mac Lane's pentagon condition under F.
- The center diamond commutes because of the naturality of τ.
- The top quadrilaterals of both the front left and the front right commute out of the naturality of α.
- The bottom hexagons of both the front left and the front right commute because of the hexagon coherence condition.
- The back top left and top right are the hexagon coherence condition tensored with the identity on each side.
- The bottom of the back left and the back right commute by the naturality of τ.
- The top of the back commutes by naturality of α and the bottom of the back commutes because it is an instance of the hexagon coherence condition.

- The main point is that this diagram consists of naturality squares, pentagons, and hexagons.
- If one assumes the pentagons and the hexagons commute, then between any two objects in the diagram, there is at most one morphism between them.
- It can be shown that for any n, the mapping funnel that combines all the n elements into one pair of parentheses is also made of pentagons, hexagons, and naturality squares.
- This is similar to our other coherence conditions where we assume some set of smaller diagrams commute and show that all the larger ones commute.

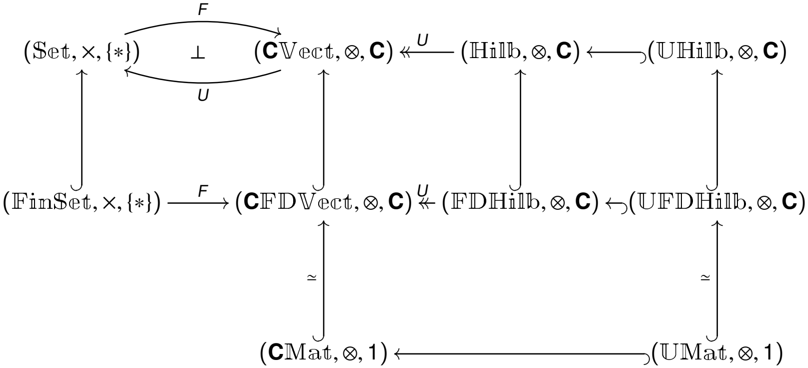
Examples

The rest of this section consists of examples of monoidal functors.

Examples from Linear Algebra

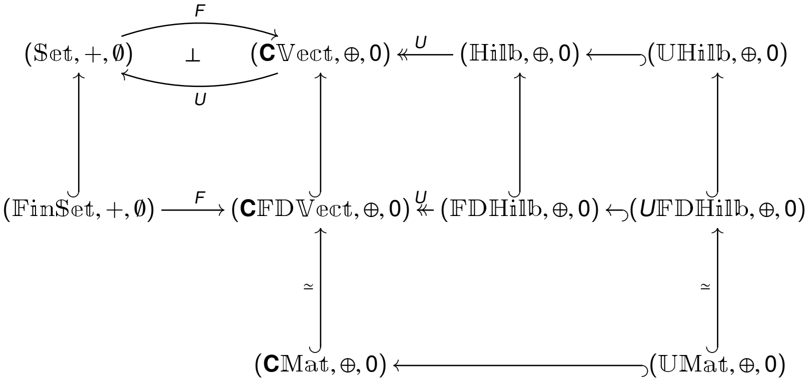
- In this diagram, we summarized all the functors that we have been dealing with relating sets, vector spaces, matrices, unitary operators, unitary matrices, and Hilbert spaces.
- We described monoidal structures on these categories in the mini-course on Advanced Linear Algebra.
- The next two slides extend that diagram to include the monoidal structures and the monoidal functors.
- All the categories are strictly associative symmetric monoidal categories and all the functors are symmetric monoidal functors (not neccessarily strict.)
- The next slide has the tensor products and Cartesian products monoidal structures.
- The slide after has the coproducts and direct sums structures.

Examples from Linear Algebra



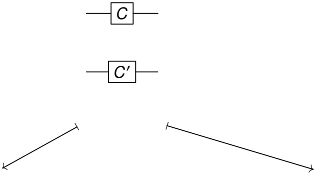
Monoidal categories with tensors and functors from linear algebra.

Examples from Linear Algebra



Monoidal categories with direct sums and functors from linear algebra.

The following three examples of monoidal functors are connected as follows



$$\begin{matrix} FuncDesc(C) \\ \oplus \\ FuncDesc(C') \end{matrix} \longmapsto \begin{bmatrix} MatrixDesc(C) & 0 \\ 0 & MatrixDesc(C') \end{bmatrix}$$

Monoidal functors with circuits, functions, and matrices.

Example

- *Here we introduced the functor* $FuncDesc\colon \mathbb{Circuit} \longrightarrow Bool\mathbb{Func}$.
- *We discussed the monoidal structure of* $\mathbb{Circuit}$ *here.*
- *We discussed and the monoidal structure of* $Bool\mathbb{Func}$ *here.*
- *The functor is a strict monoidal functor:*

$$FuncDesc(C \oplus C') = FuncDesc(C) \oplus FuncDesc(C').$$

The $\oplus$ *on the left is disjoint parallel processing of the two circuits, while the* $\oplus$ *on the right means the disjoint union of two functions.*

Examples with Circuits, Functions, and Matrices

Example

- *The functor MatrixDesc* $: \mathbb{C}\text{ircuit} \longrightarrow$ *Bool*$\mathbb{M}\text{at}$ *was introduced here.*
- *We discussed the monoidal structure of* $\mathbb{C}\text{ircuit}$ *here.*
- *We discussed the monoidal structure of Bool*$\mathbb{M}\text{at}$ *here.*
- *The functor is a strict monoidal functor because*

$$MatrixDesc(C \oplus C') = MatrixDesc(C) \oplus MatrixDesc(C').$$

Examples with Circuits, Functions, and Matrices

Example

- *The functor FuncEval*: *Bool*$\mathbb{Func} \longrightarrow$ *Bool*$\mathbb{Mat}$ *was introduced here.*
- *The category Bool*$\mathbb{Mat}$ *has a monoidal structure which is just addition on objects. On morphisms (Boolean matrices), this is disjoint union as described here.*
- *The functor FuncEval respects the monoidal structure.*

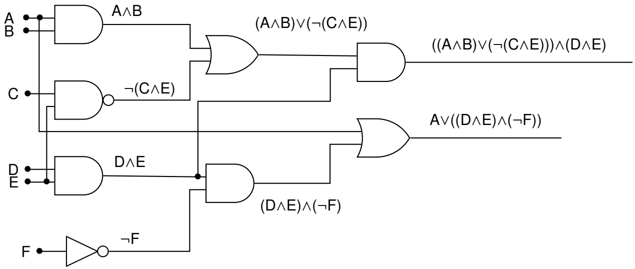
An Example with Logical Formulas

- Logical circuits are intimately related to logical formulas.
- We describe a category $\mathbb{Logic}$, whose morphisms are sequences of logical formulas.
- We will then describe a functor

$$L : \mathbb{Circuit} \longrightarrow \mathbb{Logic}$$

  which will take every circuit to the sequence of logical formulas that is associated with it.

An Example with Logical Formulas



The above logical circuit corresponds to the sequence of logical formulas:

$$((A \wedge B) \vee (\neg(C \wedge E))) \wedge (D \wedge E), \qquad A \vee ((D \wedge E) \wedge (\neg F)).$$

An Example with Logical Formulas

A Category Defined

- *The collection of logical formulas form a category* $\mathbb{L}\text{ogic}$.
- *The objects of* $\mathbb{L}\text{ogic}$ *are the natural numbers.*
- *The morphisms from m to n are equivalence classes of n-tuples of logical formulas where each formula uses at most m variables.*
- *Two n-tuples are considered equivalent if they are the same formulas except for an exchange of variable names.*
- *For example, the sequence described in the last slide represent an element of* $Hom_{\mathbb{L}\text{ogic}}(6, 2)$ *because there are 6 variables and 2 formulas.*

An Example with Logical Formulas

A Category Defined (Continued.)

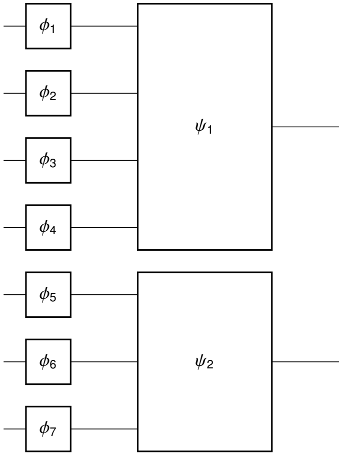
- *A general morphism will be written as* $\Phi\colon m \longrightarrow n$ *which corresponds to logical formulas* $(\phi_1, \phi_2, \ldots, \phi_n)$ *where each* ϕ_i *has at most m variables.*
- *A map* $\Phi\colon 0 \longrightarrow m$ *corresponds to an m-tuple of true and false values. A map* $\Phi\colon m \longrightarrow 0$ *corresponds to the empty sequence of formulas.*

An Example with Logical Formulas

A Category Defined (Continued.)

- *Composition in $\mathbb{Logic}$ can be understood by looking at an example of composition of circuits as shown in the next slide.*
- *There are seven circuits on the left which will correspond to a morphism in $\mathbb{Logic}$ written as $\Phi\colon m \longrightarrow 7$ or $\Phi = (\phi_1, \phi_2, \ldots, \phi_7)$.*
- *These compose into two circuits on the right.*
- *The four top wires enter the circuit that corresponds ψ_1.*
- *The four variables in ψ_1 will be changed to the formulas ϕ_1, ϕ_2, ϕ_3 and ϕ_4 that correspond to the four circuits on the left.*
- *Similarly with the three wires entering the bottom circuit.*

An Example with Logical Formulas



Composition of logical circuits.

An Example with Logical Formulas

A Category Defined (Continued.)

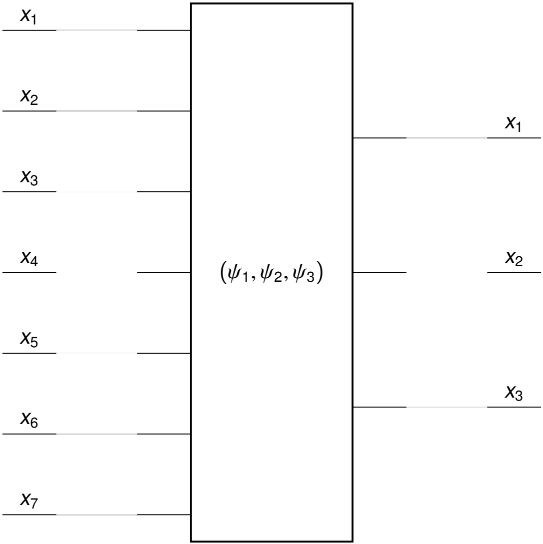
- *Let us be formal about composition.*
- *Let* $\Phi\colon m \longrightarrow n$ *correspond to the logical formulas* $(\phi_1, \phi_2, \ldots, \phi_n)$ *and* $\Psi\colon n \longrightarrow p$ *correspond to the logical formulas* $(\psi_1, \psi_2, \ldots \psi_p)$.
- *The composition* $\Psi \circ \Phi\colon m \longrightarrow p$ *corresponds to the logical formulas* $(\xi_1, \xi_2, \ldots, \xi_p)$ *where each* ξ_i *uses at most m variables.*
- *The* ξ_i *is defined by substituting the jth variable with* ϕ_j.

An Example with Logical Formulas

A Category Defined (Continued.)

The identity map $id_n \colon n \longrightarrow n$ *in* $\mathbb{Logic}$ *is the sequence of* n *logical formulas* $(x_1, x_2, \ldots, x_n)$. *The fact that composition with such identity maps does not change the equivalence classes of logical formulas can be seen by looking at the next slide.*

An Example with Logical Formulas



Composition with identities of logical circuits

An Example with Logical Formulas

A Category Defined (Continued.)

- *This category of logical formulas has a symmetric monoidal structure which corresponds to the disjoint union of logical circuits,* $(\mathbb{L}\mathrm{ogic}, \oplus, 0)$.
- *The monoidal structure on the objects is simply addition, i.e.,* $m \oplus n = m + n$.
- *Given* $\Phi\colon m \longrightarrow n$ *which correspond to the logical formulas* $(\phi_1, \phi_2, \ldots, \phi_n)$
- *and* $\Psi\colon m' \longrightarrow n'$ *which corresponds to the logical formulas* $(\psi_1, \psi_2, \ldots, \psi_{n'})$,
- *then* $\Phi \oplus \Psi\colon m + m' \longrightarrow n + n'$ *corresponds to the logical formulas* $(\phi_1, \phi_2, \ldots, \phi_n, \psi_1, \psi_2, \ldots, \psi_{n'})$ *where each of the* ϕ*'s and* ψ*'s use at most* $m + m'$ *variables.*

An Example with Logical Formulas

A Category Defined (Continued.)

- *Since the category and monoidal structure of* $\mathbb{Logic}$ *was made to be similar to* $\mathbb{Circuit}$*, it is not surprising that there is a strict monoidal functor* $L : \mathbb{Circuit} \longrightarrow \mathbb{Logic}$ *that takes every logical circuit to the logical formulas it describes.*
- *In detail, on objects,* $L(m) = m$*.*
- L *takes a circuit with* m *inputs and* n *outputs to* n *logical formulas each using up to* m *variables.*

An Example with Logical Formulas

A Category Defined (Continued.)

- *The generators of the category are the logical gates AND, OR, NOT, NAND, etc.*
- *The functor takes these gates to the logical formulas* $A \wedge B$, $A \vee B$, $\neg A$, *and* $\neg(A \wedge B)$, *etc.*
- *The functor* L *respects the composition in the categories because the composition in* $\mathbb{Logic}$ *was made to mimic the composition in* $\mathbb{Circuit}$.
- *Similarly,* L *is a strict monoidal functor. Interestingly,* L *is not a symmetric monoidal functor because* $\mathbb{Circuit}$ *is not a symmetric monoidal category.*

Foreshadowing

- Chapter 6: Relationships Between Monoidal Categories
  - Section 6.2: Coherence Theorems
    - Strictification Theorem for Monoidal Categories
    - Coherence Theorem for Monoidal Categories
    - Coherence Theorem for Symmetric Monoidal Categories

Introduction

In this Section we state theorems about monoidal categories and we state theorems about the relationship of monoidal categories and strict monoidal categories. For the most part, we leave the technical details of the proof for the text and here we just give an overview.

- We start by proving a coherence theorem that says that Assoc is the paradigm of a monoidal category in the sense that it is the free monoidal category on one generator. This will lead to profound statements about every monoidal category.
- Then we will state and prove a strictification theorem which says that every monoidal category is monoidally equivalent to a strict monoidal category. Since this is a fundamental theorem about monoidal categories, we prove it in two different ways.

Strictification Theorem

- In both proofs we associate a strict monoidal category to a monoidal category. We will then show that the original monoidal category is monoidally equivalent to the strict one. The tensor products of the strict monoidal categories are (i) string concatenation, and (ii) function composition. These are two paradigms of strictly associative operations.
- Then the strictification theorem is used to prove the coherence theorem.
- We conclude with similar theorems about symmetric monoidal categories.

The Coherence Theorem for Monoidal Categories

The following theorem is the first coherence theorem and future coherence theorems will follow the same form as this one. Before we tackle it, we will go back to the free monoid adjunction. The ideas in that example will be needed. In particular, think about the free monoid on a one-object set.

Reminder

The following adjunction is a paradigm for many examples of **free-forgetful adjunctions**.

Example

- *There is a forgetful functor* $U\colon \mathbb{M}\mathrm{onoid} \longrightarrow \mathbb{S}\mathrm{et}$ *that takes every monoid to its underlying set.*
- *There is the free monoid functor* $F\colon \mathbb{S}\mathrm{et} \longrightarrow \mathbb{M}\mathrm{onoid}$ *that takes every set* S *to* S^**.*
- *We saw that* F *is left adjoint to* U*. This means that for all sets* S *and for all monoids* M*, there is the following natural isomorphism:*

$$Hom_{\mathbb{M}\mathrm{onoid}}(F(S), M) \cong Hom_{\mathbb{S}\mathrm{et}}(S, U(M)).$$

- *Let us to examine the free monoid on one object, say* $*$*.*

Example (Continued.)

- *The monoid will consist of* $*, **, ***, \ldots$. *There will also be the empty set as the unit. This monoid is isomorphic to the monoid of natural numbers* $(\mathbf{N}, +, 0)$.
- *The universal property says that for every set function* $f\colon \{*\} \longrightarrow U(M)$ *— which is a function that picks out an element m of M — there is a monoid homomorphism* $f'\colon \mathbf{N} \longrightarrow M$.
- *the output of the function* f' *is m, mm, mmm,* $\ldots$.
- *Let us restate this in a way that will be useful. The free monoid on one object will have the property that for every monoid M and every element m in M, there is a unique morphism from the free monoid on one object to M that takes* $*$ *to m.*

.

Reminder

Example (Continued.)

- *Let us summarize the properties of $F(\{*\})$ in three different ways.*
  - *The monoid $F(\{*\})$ is the free monoid on one generator.*
  - *For every object m in M, there is a unique morphism $f\colon F(*) \longrightarrow M$ such that $f(*) = m$.*
  - *We have*

$$Hom_{\mathbb{Monoid}}(F(\{*\}), M) \cong Hom_{\mathbb{Set}}(\{*\}, U(M)) \cong U(M).$$

This means that there is an isomorphism $Hom_{\mathbb{Monoid}}(F(\{*\}), M) \cong U(M)$ where $U(M)$ is the set of elements of M.

Now back to the main theorem.

Theorem

The Coherence Theorem for Monoidal Categories. *We state it in four equivalent ways.*

- *For every monoidal category* $(\mathbb{C}, \otimes, I, \alpha, \rho, \lambda)$ *and every object* x *in* $\mathbb{C}$*, there is a unique strict monoidal functor* $F_x \colon \mathbb{A}\mathrm{ssoc} \longrightarrow \mathbb{C}$ *such that* $F(\bullet) = x$*.*
- *There is an isomorphism of categories*

$$Hom(\mathbb{A}\mathrm{ssoc}, \mathbb{C}) \cong Hom_{\mathbb{C}\mathrm{at}}(\mathbf{1}, \mathbb{C})$$

  where the Hom set on the left is strict monoidal functors.
- *The monoidal category* $\mathbb{A}\mathrm{ssoc}$ *is the free monoidal category on one generator.*
- *Two morphisms in* $\mathbb{C}$ *in the image of* F_x *generated by identities,* α, ρ, λ *and their inverses, and built up by* $\circ$ *and* $\otimes$*, are equal.*

Strictification Theorem

Proof.

Remember the association category, $\mathbb{Assoc}$. The objects are bracketed words and between any two objects there is exactly one morphism. Let $(\mathbb{C}, \otimes, I, \alpha, \lambda, \rho)$ be a monoidal category. For any object x in $\mathbb{C}$ we will show that there is a strict monoidal functor $F_x \colon \mathbb{Assoc} \longrightarrow \mathbb{C}$. For the association of one letter $\bullet$, the functor $F_x(\bullet) = x$. It is obvious how to define F_x on objects of $\mathbb{Assoc}$. For an association $\bullet\bullet$, the functor will go to $x \otimes x = xx$. The association $\bullet(\bullet\bullet)$ will go to $x(xx)$. Within $\mathbb{Assoc}$ there is a unique maps $\bullet(\bullet\bullet) \longrightarrow (\bullet\bullet)\bullet$ that F_x takes to α in $\mathbb{C}$. There are also maps λ and ρ in $\mathbb{Assoc}$ which F_x takes to λ and ρ in $\mathbb{C}$. □

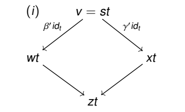
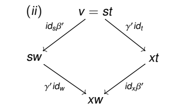
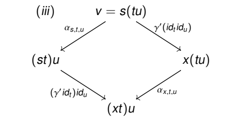
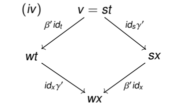
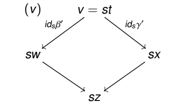
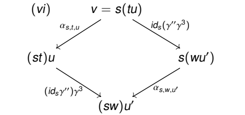
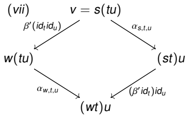
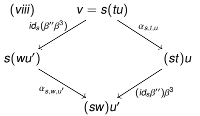
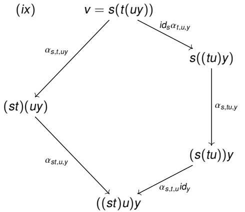
Continued.

In the proof, we have to examine how to complete maps of the form

$$w \longleftarrow v = s \otimes t \longrightarrow x.$$

There are nine such completeing maps as can be seen in the next slide. Each of them comutes out of naturality, or bifunctoriality, or is a pentagon. This is central for the proof. □

Strictification Theorem

$\beta \neq \gamma$	$\beta = \beta' \otimes id$	$\beta = id \otimes \beta'$	$\beta = \alpha_{s,t,u}$
$\gamma = \gamma' \otimes id$	 By induction on v.	 By bifunctoriality of $\otimes$.	 By naturality of α.
$\gamma = id \otimes \gamma'$	 By bifunctoriality of $\otimes$.	 By induction on v.	 By naturality of α.
$\gamma = \alpha_{s,t,u}$	 By naturality of α.	 By naturality of α.	 By pentagon coherence condition of α.

Nine cases of completing two maps coming out of a tensor product.

Strictification Theorem

Let us deal with the other three ways to state the coherence theorem.

- The natural isomorphism follows from the fact that x in $\mathbb{C}$ is chosen by a $F' : \mathbf{1} \longrightarrow \mathbb{C}$. This functor determines and is determined by the unique functor $F_x : \mathbb{Assoc} \longrightarrow \mathbb{C}$.
- By "free monoidal category" we mean that there is a forgetful functor $U : \mathbb{MonCat} \longrightarrow \mathbb{Cat}$. This functor has a left adjoint $Free : \mathbb{Cat} \longrightarrow \mathbb{MonCat}$ which takes a category to its free monoidal category.The equation says that $Free(\mathbf{1}) = \mathbb{Assoc}$.
- Since in $\mathbb{Assoc}$ there is only one morphism between each bracketing in $\mathbb{Assoc}$, the functors F_x only go to one morphism. Notice in an arbitrary monoidal category there need not be any morphism between bracketing of words where the underlying order is not the same. This means that there might not be a morphism of the form $a \otimes b \longrightarrow b \otimes a$.

Strictification Theorem

Theorem (Strictification Theorem)

Strictification Theorem for Monoidal Categories. *Every monoidal category is monoidally equivalent to a strict monoidal category.*

Strictification Theorem

Proof.

Using string concatenation. Let $(\mathbb{C}, \otimes, \alpha, I, \lambda, \rho)$ be a monoidal category. As we saw, the category of strings on an alphabet with the concatenation operation is a strict monoidal category. We form such a strict monoidal category $(\mathbb{C}^{\bullet}, \bullet, \emptyset)$ that will be monoidally equivalent to $(\mathbb{C}, \otimes, \alpha, I, \lambda, \rho)$. □

Strictification Theorem

Proof.

Using function composition. As we saw, the collection of endomorphisms of a category with composition is a strict monoidal category. Let $(\mathbb{C}, \otimes, \alpha, I, \lambda, \rho)$ be a monoidal category. We form such a strict monoidal category $(\mathbb{C}^\circ, \circ, Id_{\mathbb{C}})$. The monoidal category $\mathbb{C}$ will be monoidally equivalent to a subcategory of this strict monoidal category. □

Strictification Theorem

This theorem shows that the inclusion 2-functor

$$\overline{\mathbb{StrMonCat}} \hookrightarrow \overline{\mathbb{MonCat}}$$

is not only full and faithful, but it is almost essentially surjective in the sense that every monoidal category is monoidally *equivalent* (not necessarily isomorphic) to a strict monoidal category.

Strictification Theorem

How does this theorem help us? When we use our favorite category like $\mathbb{Set}$, $\mathbb{Group}$, $\mathbb{Graph}$, or $\mathbb{Top}$, we know that the Cartesian product is not strict. This theorem tells us that although it is not strict, it is strongly monoidally equivalent to another category that does have a strict monoidal product. Since this equivalence is monoidal, most of the properties of the strict monoidal category are the same as the original category. So, what is true about the strict monoidal category is true about the equivalent original monoidal category. Therefore when dealing with a monoidal category, you might as well imagine it is a strict monoidal category.

The Coherence Theorem for Monoidal Categories

Now we use the strictification theorem to prove the coherence theorem.

Theorem

Let $\mathbb{C}$ be a monoidal category and $\mathbb{C}^x$ be a strict monoidal category that is equivalent to $\mathbb{C}$ with a monoidal functor $L : \mathbb{C} \longrightarrow \mathbb{C}^x$. ($\mathbb{C}^x$ can be $\mathbb{C}^\bullet$ or $\mathbb{C}^\circ$ given in the two proofs of the strictification theorem.) Let u and v be two ways of bracketing any objects in $\mathbb{C}$, i.e., they are two functors $\mathbb{C}^n \longrightarrow \mathbb{C}$ that only use the tensor product, identities, and composition. If ϕ and ψ are two natural transformations from u to v which are generated by identities, α, ρ, λ and their inverses, and are built up using $\circ$ and $\otimes$, then $\phi = \psi$.

The Coherence Theorem for Symmetric Monoidal Categories

Now that we have dealt with coherence for monoidal categories, we can move on to coherence theory for symmetric monoidal categories. Here we work with the category $\mathbb{Sym}$ rather than the category $\mathbb{Assoc}$.

Theorem

The Coherence Theorem for Symmetric Monoidal Categories. *We state it in four equivalent ways.*

- *For every symmetric monoidal category* $(\mathbb{C}, \otimes, I, \alpha, \rho, \lambda, \gamma)$ *and every object* x *in* $\mathbb{C}$, *there is a unique strict symmetric monoidal functor* $F_x \colon \mathbb{S}\mathrm{ym} \longrightarrow \mathbb{C}$ *such that* $F_x(1) = x$.
- *There is an isomorphism of categories*

$$Hom(\mathbb{S}\mathrm{ym}, \mathbb{C}) \cong Hom_{\mathbb{C}\mathrm{at}}(\mathbf{1}, \mathbb{C})$$

where the Hom set on the left is strict symmetric monoidal functor.

Theorem

The Coherence Theorem for Symmetric Monoidal Categories

- *The symmetric monoidal category* $\mathbb{S}\mathrm{ym}$ *is the free symmetric monoidal category on one generator.*
- *Two morphisms in* $\mathbb{C}$ *generated by images of* F_x *of identities,* $\alpha, \rho, \lambda, \gamma$ *and their inverses, built up by* $\circ$ *and* $\otimes$ *are equal if the two underlying permutations are the same.*

Strictification Theorem for Symmetric Monoidal Categories

The strictification theorem for monoidal categories extends to symmetric monoidal categories.

Theorem (Strictification Theorem)

Strictification Theorem for Symmetric Monoidal Categories. *Every symmetric monoidal category is monoidally equivalent to a strictly associative symmetric monoidal category. This equivalence is via symmetric monoidal functors and symmetric monoidal natural transformations. This is done by making the underlying monoidal category strict and then transport the symmetric structure to the strict monoidal structure. .*

Strictification Theorem for Symmetric Monoidal Categories

This theorem shows that the inclusion 2-functor

$$\overline{\mathbb{StrSymMonCat}} \hookrightarrow \overline{\mathbb{SymMonCat}}$$

is not only full and faithful, but it is almost essentially surjective in the sense that every symmetric monoidal category is monoidally equivalent (not necessarily isomorphic) via symmetric monoidal functors and symmetric monoidal natural transformations to a strictly associative symmetric monoidal category.

Foreshadowing

- Chapter 6: Relationships Between Monoidal Categories
  - Section 6.3: When Coherence Fails
    - The Flexibility of Coherence
    - Examples

The Flexibility of Coherence

While the coherence theorem is one of the central ideas in the world of monoidal categories, it is only the beginning of the story. We will see that there are many other structures, and for each structure there are many different coherence conditions.
Let us examine the pentagon coherence condition. Consider the category of vector spaces and let V, W, and X be three vector spaces. Using the tensor product, we form the vector spaces $V \otimes (W \otimes X)$ and $(V \otimes W) \otimes X$. There is a linear isomorphism $\alpha\colon V \otimes (W \otimes X) \longrightarrow (V \otimes W) \otimes X$ defined by

$$\langle v, \langle w, x \rangle\rangle \longmapsto \langle\langle v, w\rangle, x\rangle$$

which satisfies the pentagon coherence condition.

The Flexibility of Coherence

Formally, the fact that the pentagon commutes means that

$$(\alpha_{V,W,X} \otimes id_Y) \circ (\alpha_{V,WX,Y}) \circ (id_V \otimes \alpha_{W,X,Y}) = (\alpha_{VW,X,Y}) \circ (\alpha_{V,W,XY}).$$

Since the α's are isomorphisms and we can work with the inverses of the morphisms, we can rewrite this as

$$(\alpha_{VW,X,Y})^{-1} \circ (\alpha_{V,W,XY})^{-1} \circ (\alpha_{V,W,X} \otimes id_Y) \circ (\alpha_{V,WX,Y}) \circ (id_V \otimes \alpha_{W,X,Y})$$

$$= id_{V(W(XY))}.$$

That is, rather than saying that the composite of the three maps in the pentagon coherence condition is the same map as the composite of the two other maps, we can say that going around the pentagon completely brings one right back to the element where one started.

Now let us look at an example where the pentagon coherence condition fails. Consider the linear isomorphism $\alpha' \colon V \otimes (W \otimes X) \longrightarrow (V \otimes W) \otimes X$ defined by

$$\langle v, \langle w, x \rangle\rangle \longmapsto (-1)\langle\langle v, w\rangle, x\rangle.$$

While this is a legitimate isomorphism, the pentagon coherence condition is *emphatically* not satisfied. Using the three composable maps means taking the elements to the composition of three -1's meaning that

$$\langle v, \langle w, \langle x, y\rangle\rangle \longmapsto (-1)\langle\langle\langle v, w\rangle, x\rangle, y\rangle.$$

In contrast, using the other two maps has two (an even number) (-1)s which gives the map

$$\langle v, \langle w, \langle x, y\rangle\rangle \longmapsto \langle\langle\langle v, w\rangle, x\rangle, y\rangle.$$

These are different maps.

Another way of saying this is that going all the way around the pentagon condition gives us

$$(\alpha'_{VW,X,Y})^{-1} \circ (\alpha'_{V,W,XY})^{-1} \circ (\alpha'_{V,W,X} \times id_Y) \circ (\alpha'_{V,WX,Y}) \circ (id_V \times \alpha'_{W,X,Y})$$

$$= (-1) Id_{V(W(XY))}.$$

This is not the identity. Rather than saying that α' is coherent, we say that it has an "obstruction to coherence." Notice that going around the pentagon with α' does not give the identity, but going around the pentagon with α' **twice** is a composition of ten (an even number) linear maps ($-1^{10} = 1$) and hence does give the identity. This example can be generalized: for every n, there are reassociators that commute when going around the pentagon n times and not less.

The Flexibility of Coherence

- In my thesis, different types of pentagon coherence conditions were studied. There was a general reassociators that have no requirement of making the pentagon commute at all.
- That is, they will not commute going around the pentagon any amount of times.
- This reassociator is simply an isomorphism and has no commuting axiom. This is like an associahedron with open pentagons instead of filled-in pentagons. These shapes are interesting.
- Such an associahedron for four letters will be equivalent to a circle and not a (commutative) disk.
- Such an associahedron for five letters is equivalent to a sphere with six pentagons removed and only four naturality squares in place. This is different than the coherent version where we saw that there is a unique morphism between any two vertices as a filled in sphere.
- One can learn about how un-coherent a reassociator is by looking at these shapes.

The Flexibility of Coherence

- While working on these issues, I came upon an interesting shape.
- Consider the associahedron on seven letters, A_7.
- There are 132 ways of bracketing 7 letters and hence this shape has 132 vertices. The figure on the next slide shows shows a part of A_7 which looks like 36 objects.

$$
\begin{array}{ccccccccccc}
(a(b(cd)))[e[fg]] & \longrightarrow & (a((bc)d))[e[fg]] & \longrightarrow & ((a(bc))d)[e[fg]] & \longrightarrow & (((ab)c)d)[e[fg]] & \longleftarrow & ((ab)(cd))[e[fg]] & \longleftarrow & (a(b(cd)))[e[fg]] \\
\downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\
(a(b(cd)))[[ef]g] & \longrightarrow & (a((bc)d))[[ef]g] & \longrightarrow & ((a(bc))d)[[ef]g] & \longrightarrow & (((ab)c)d)[[ef]g] & \longleftarrow & ((ab)(cd))[[ef]g] & \longleftarrow & (a(b(cd)))[[ef]g] \\
\downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\
[(a(b(cd)))[ef]]g & \longrightarrow & [(a((bc)d))[ef]]g & \longrightarrow & [((a(bc))d)[ef]]g & \longrightarrow & [(((ab)c)d)[ef]]g & \longleftarrow & [((ab)(cd))[ef]]g & \longleftarrow & [(a(b(cd)))[ef]]g \\
\downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow \\
[[(a(b(cd)))e]f]g & \longrightarrow & [[(a((bc)d))e]f]g & \longrightarrow & [[((a(bc))d)e]f]g & \longrightarrow & [[(((ab)c)d)e]f]g & \longleftarrow & [[((ab)(cd))e]f]g & \longleftarrow & [[(a(b(cd)))e]f]g \\
\uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow \\
[(a(b(cd)))e][fg] & \longrightarrow & [(a((bc)d))e][fg] & \longrightarrow & [((a(bc))d)e][fg] & \longrightarrow & [(((ab)c)d)e][fg] & \longleftarrow & [((ab)(cd))e][fg] & \longleftarrow & [(a(b(cd)))e][fg] \\
\uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow & & \uparrow \\
(a(b(cd)))[e[fg]] & \longrightarrow & (a((bc)d))[e[fg]] & \longrightarrow & ((a(bc))d)[e[fg]] & \longrightarrow & (((ab)c)d)[e[fg]] & \longleftarrow & ((ab)(cd))[e[fg]] & \longleftarrow & (a(b(cd)))[e[fg]]
\end{array}
$$

Part of the associahedra for seven letters.

The Flexibility of Coherence

- Notice that each row and column has three arrows one way and two arrows the other way. That is, each row and column is actually an instance of Mac Lane's pentagon.
- For readability, The figure on the next slide shows a "zoomed-in" view of the upper left-hand corner of the last figure.
- The horizontal α maps move the round parentheses and the vertical β maps move the square parentheses.
- By examining the subscripts of the α and β maps, one sees that each square commutes because of naturality.

The Flexibility of Coherence

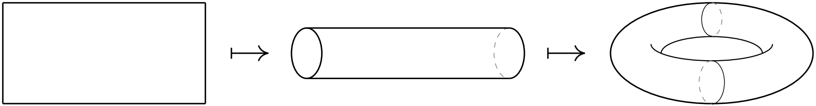
$$
\begin{array}{ccccc}
(a(b(cd)))[e[fg]] & \xrightarrow{(id_a \otimes \alpha_{b,c,d}) \otimes id_{e[fg]}} & (a((bc)d))[e[fg]] & \xrightarrow{\alpha_{a,bc,d} \otimes id_{e[fg]}} & ((a(bc))d)[e[fg]] \\
\downarrow {\scriptstyle id_{a(b(cd))} \otimes \beta_{e,f,g}} & & \downarrow {\scriptstyle id_{a((bc)d)} \otimes \beta_{e,f,g}} & & \downarrow {\scriptstyle id_{(a(bc))d} \otimes \beta_{e,f,g}} \\
(a(b(cd)))[[ef]g] & \xrightarrow{(id_a \otimes \alpha_{b,c,d}) \otimes id_{[ef]g}} & (a((bc)d))[[ef]g] & \xrightarrow{\alpha_{a,bc,d} \otimes id_{[ef]g}} & ((a(bc))d)[[ef]g] \\
\downarrow {\scriptstyle \beta_{a(b(cd)),ef,g}} & & \downarrow {\scriptstyle \beta_{((a(bc))d),ef,g}} & & \downarrow {\scriptstyle \beta_{(a(bc))d,ef,g}} \\
[(a(b(cd)))[ef]]g & \xrightarrow[(id_a \otimes \alpha_{b,c,d}) \otimes id_{[ef]g}]{} & [(a((bc)d))[ef]]g & \xrightarrow[\alpha_{a,bc,d} \otimes id_{[ef]g}]{} & [((a(bc))d)[ef]]g
\end{array}
$$

The top left part of the torus for seven letters.

The Flexibility of Coherence

- Notice also that the top row and the bottom row of the last part of A_7 are exactly the same.
- Similarly, the left column is exactly the same as the right column. (This is similar to the Pac-Man game screen, where the players can go out the top and come in at the bottom and go out the left and come in on the right.)
- Since the top and the bottom rows are the same, we can “bend the paper” and glue the edges together as in the next figure.
- Since the left and right columns are the same, we can “bend the tube” and paste them together as depicted.
- This means that within A_7 there is a hollow doughnut or torus.
- This is true for A_7 and also true for any higher associahedra. There are many other interesting shapes within the associahedra, permutohedra, and permuto-associahedra.

The Flexibility of Coherence



Folding part of the associahedra for seven letters into a torus.

Examples

- There are many other coherence conditions discussed in the literature.
- For example, we saw that we can require the reassociator to be the identity (strict monoidal categories) or an isomorphism (monoidal categories). What if we require that the reassociator just be a morphism that satisfies the pentagon condition?
- A category with a monoidal product that has a reassociator that need not be an isomorphism is called a **pre-monoidal category**. There is a lot of applications for such structures in diverse areas — including an application to quark confinement in the physics literature.
- We will see more coherence conditions in Chapter 7 where we will discuss symmetric monoidal categories that fail the symmetry condition. Such a categorical structure is called a **braided monoidal category**.

Examples

- Not only are there many different types of coherence conditions for categories, there are also many different coherence conditions for functors.
- We outlined some of the varieties in the definition But this is not the end of the story.
- In my thesis,there were functors F with a natural transformation mapping funnel $\tau\colon \otimes' \circ (F \times F) \Longrightarrow F \circ \otimes$ that do not necessarily satisfy the hexagon coherence condition.
- Various types of mapping funnels for n letters were studied.

Examples

- The point we are making in this section is that coherence conditions are not toggle switches that are either on or off.
- Rather, there are a whole spectrum of coherence conditions and every coherence condition implies different properties.
- The different structures are used to describe various mathematical and real-world phenomena.
- Just as algebra is used in every aspect of mathematics and science, coherence theory — also called **higher-dimensional algebra** — arises in many areas of mathematics and science.
- It is certain we will see much more coherence theory in the coming decades.

Mini-course:

Duality Theory

Foreshadowing

- Chapter 6: Relationships Between Monoidal Categories
  - Section 6.4: Mini-course: Duality Theory
    - Boolean Algebras
    - Stone Duality
    - Baby Stone Duality
    - Juvenile Stone Duality
    - Stone Duality
    - Dualizing Objects
    - Esakia Duality
    - Priestley Duality
    - Pointless Topology

Introduction to Duality Theory

- Category theory unifies many different areas. One of the best ways to unify two subjects is to show that two seemingly different categories essentially have the same structure.
- A **duality theorem** is a statement showing that there exists an equivalence of one category with another, e.g., $\mathbb{A} \simeq \mathbb{B}$, or more typically, $\mathbb{A}^{op} \simeq \mathbb{B}$.
- This shows that the two categories are two ways of describing the same structure. Categories that are dual to each other are mirror images of each other, and we can study one category by looking at the properties of the other.
- A monomorphism in one category corresponds to an epimorphism in the other category. An initial object in one corresponds to a terminal object in the other.
- Many times duality theorems are stated as adjunctions between $\mathbb{A}^{op}$ and $\mathbb{B}$. We look at subcategories of that are equivalent.

Examples of duality that we already saw

- $\mathbb{Rel}^{op} \cong \mathbb{Rel}$
- $\mathbf{K}\mathbb{Mat}^{op} \cong \mathbf{K}\mathbb{Mat}$
- $\mathbf{K}\mathbb{FDVect}^{op} \cong \mathbf{K}\mathbb{FDVect}$.

Notice that these examples are actually stronger than the usual duality theorems because they are isomorphisms rather than equivalences, and furthermore, they are instances of **self duality**, i.e., they show a category equivalent to the opposite of itself.

- Many other examples of duality theorems will be shown in this mini-course.
- We focus on a group of duality theorems collectively known as **Stone duality**.
- These theorems show that certain types of topological structures are related to certain types of algebraic structures.
- The algebraic structures are centered around Boolean Algebras.

Towards the Definition of a Boolean Algebra

One of the main purposes of this text is to show how different fields are related and unified. Boolean algebra is a subject that has been unifying different fields long before category theory came on the scene. A Boolean algebra is a structure that is used in mathematical logic, partial orders, logical circuits, and algebra. One way to view a Boolean Algebra is as a special type of partial order.

Towards the Definition of a Boolean Algebra

Definition

- *A partial order P with a strict Cartesian category structure $(P, \wedge, 1)$ is a* **bounded meet semilattice**.
- *Between such partial orders, we are interested in maps that preserve the $\wedge$, i.e., $f(x \wedge y) = f(x) \wedge f(y)$, and preserve the 1, i.e., $f(1) = 1$.*
- *The category of such partial orders and maps will be denoted $\mathbb{BMslattice}$.*

Towards the Definition of a Boolean Algebra

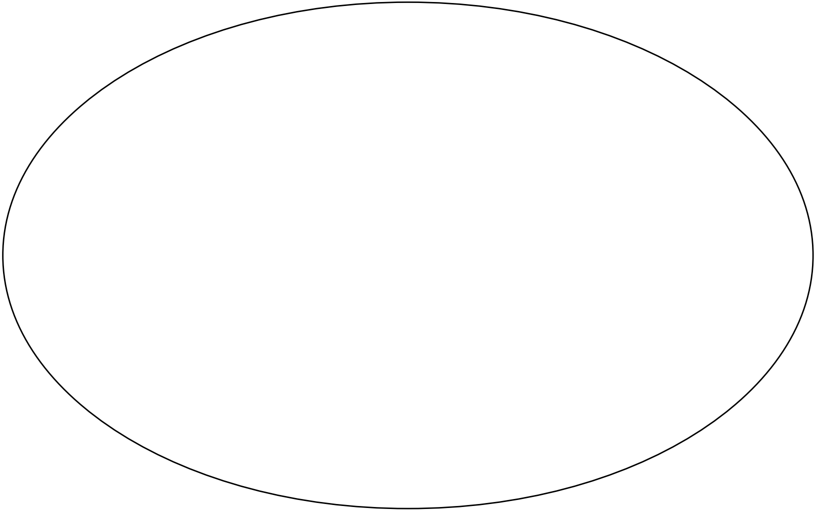
BMslattice

.

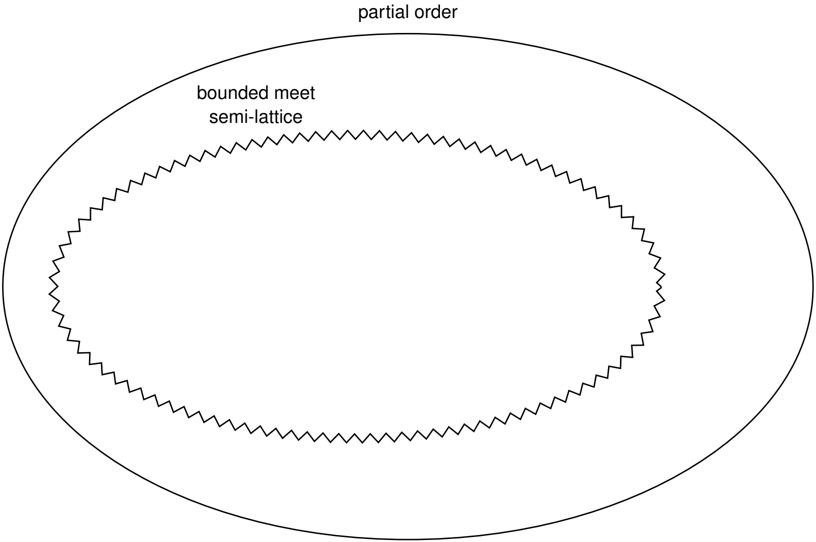
.

Towards the Definition of a Boolean Algebra

partial order



Towards the Definition of a Boolean Algebra



Towards the Definition of a Boolean Algebra

Definition

- *A partial order P with a strict co-Cartesian category structure* $(P, \vee, 0)$ *is a* **bounded join semilattice**.
- *Between such partial orders, we are interested in maps that preserve the* $\vee$*, i.e.,* $f(x \vee y) = f(x) \vee f(y)$*, and preserve the* 0*, i.e.,* $f(0) = 0$.
- *The category of such partial orders and maps will be denoted* $\mathbb{BJslattice}$.
- *There is an isomorphism of categories* $\mathbb{BMslattice} \longrightarrow \mathbb{BJslattice}$ *that takes every bounded meet lattice* $(P, \wedge, 1)$ *to the same underlying set of the partial order with the opposite order* $(P, \leq^R, 1)$ *where* $x \leq^R y$ *iff* $y \leq x$. *This isomorphism swaps* $\wedge$ *for* $\vee$*, and* 0 *for* 1.

Towards the Definition of a Boolean Algebra

BMslattice

.

.

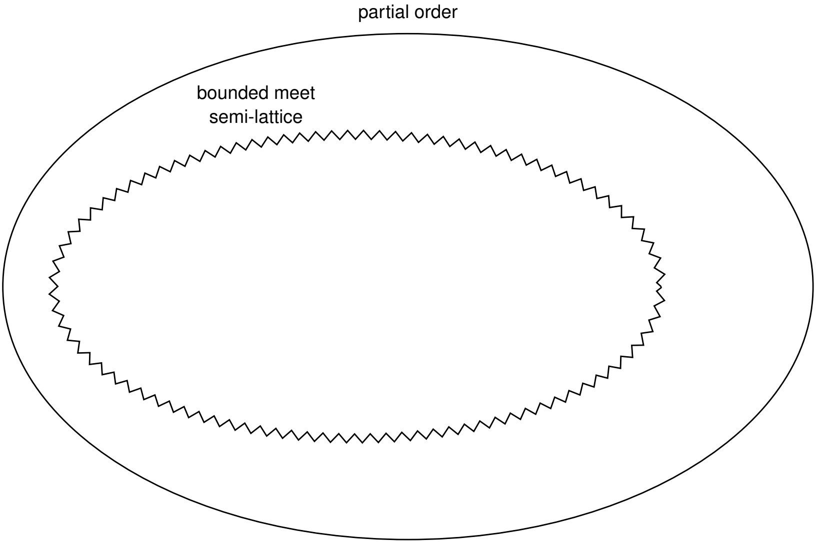
$\mathbb{BMslattice}$

.

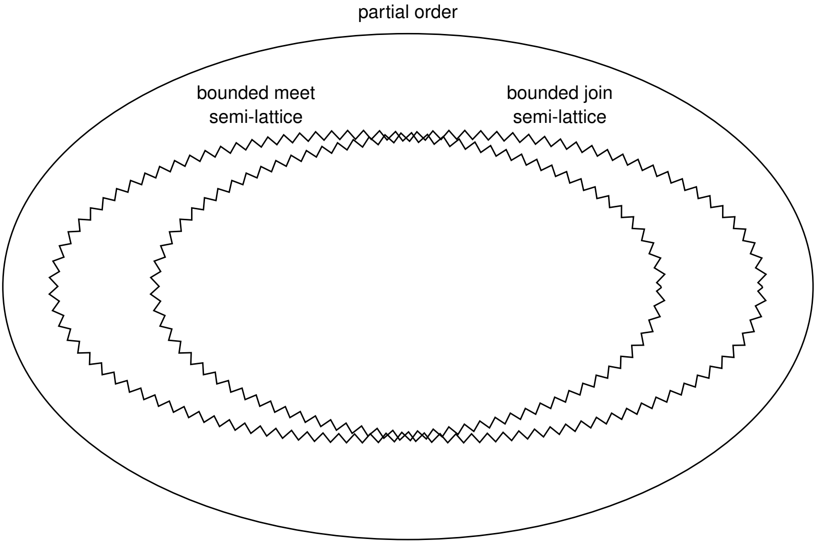
$\mathbb{BJslattice}$.

$$\mathbb{BMslattice} \xrightarrow{\ \cong\ } \mathbb{BJslattice}.$$

Towards the Definition of a Boolean Algebra



Towards the Definition of a Boolean Algebra



Towards the Definition of a Boolean Algebra

Definition

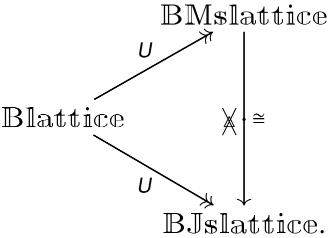
- *A partial order P with a Cartesian and a co-Cartesian structure $(P, \wedge, 1, \vee, 0)$ is called a* **bounded lattice***.*
- *The category of all such partial orders with maps that preserves the two operations and the 0 and 1 is denoted $\mathbb{Blattice}$.*
- *There are obvious forgetful functors $\mathbb{Blattice} \longrightarrow \mathbb{BMslattice}$ and $\mathbb{Blattice} \longrightarrow \mathbb{BJslattice}$.*

Towards the Definition of a Boolean Algebra

$$\begin{array}{c}\mathbb{BMslattice}\\ \Big\downarrow \cong \\ \mathbb{BJslattice}.\end{array}$$

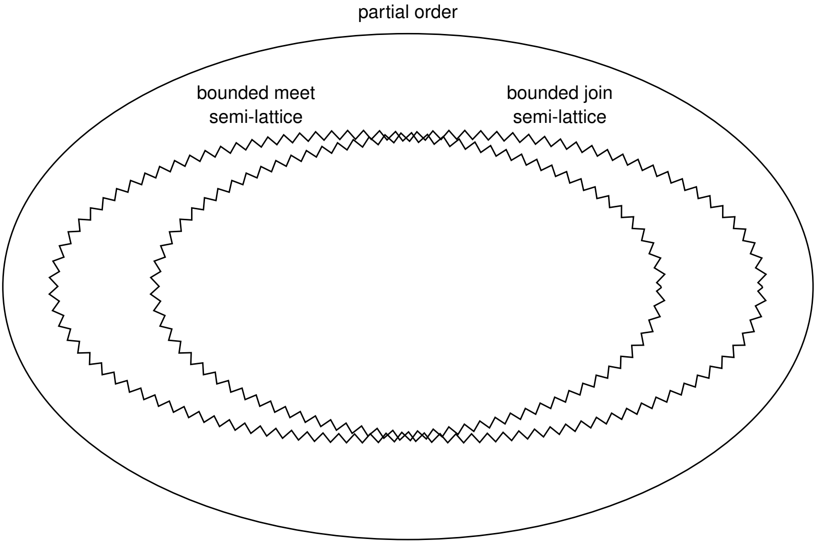
Towards the Definition of a Boolean Algebra

$$\mathbb{Blattice} \qquad \begin{array}{c} \mathbb{BMslattice} \\ \Big\downarrow \cong \\ \mathbb{BJslattice}. \end{array}$$

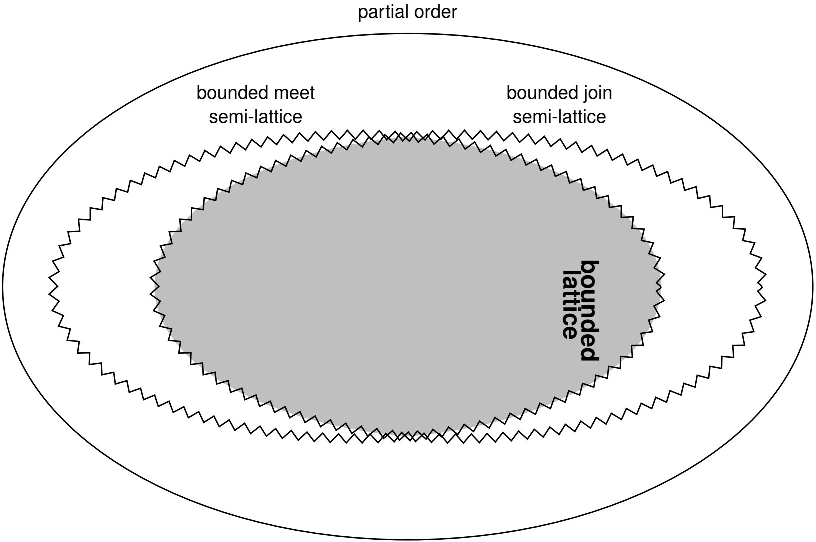


Notice that the right triangle need not commute.

Towards the Definition of a Boolean Algebra



Towards the Definition of a Boolean Algebra



Towards the Definition of a Boolean Algebra

Definition

- *A bounded lattice that satisfies*

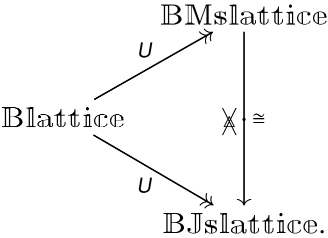
$$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$$

  and

$$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$$

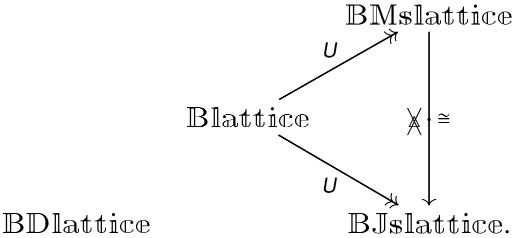
  is called a **bounded distributive lattice**.
- *The category of bounded distributive lattices has the same maps as bounded lattices and is denoted* $\mathbb{BDlattice}$.
- *There is an inclusion functor* $\mathbb{BDlattice} \hookrightarrow \mathbb{Blattice}$.

Towards the Definition of a Boolean Algebra



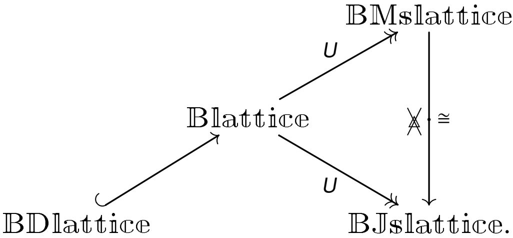
Notice that the right triangle need not commute.

Towards the Definition of a Boolean Algebra



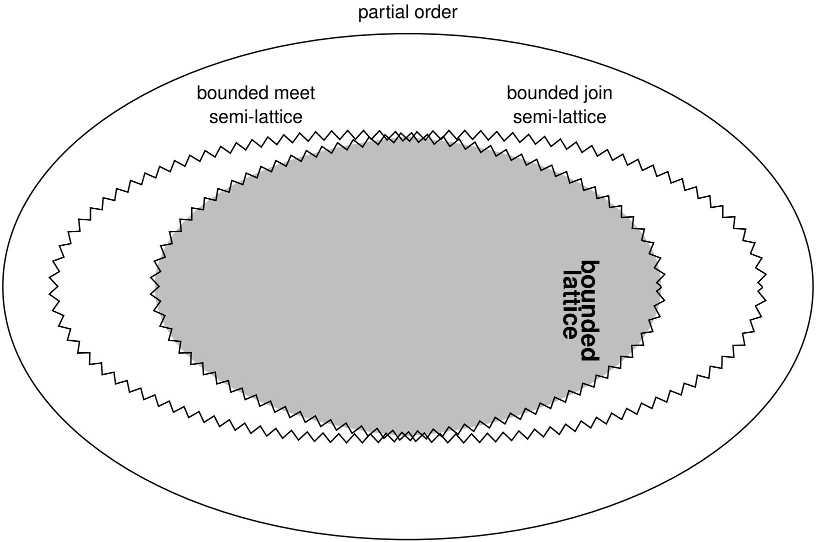
Notice that the right triangle need not commute.

Towards the Definition of a Boolean Algebra

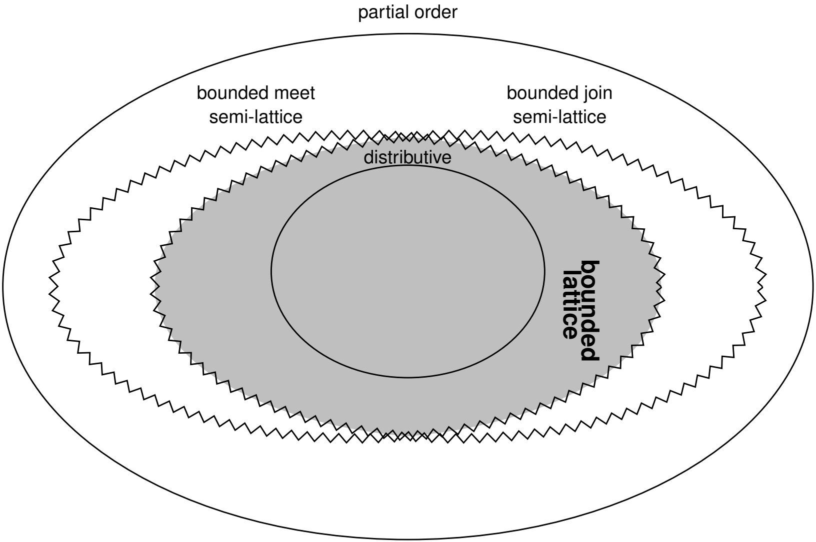


Notice that the right triangle need not commute.

Towards the Definition of a Boolean Algebra



Towards the Definition of a Boolean Algebra



Towards the Definition of a Boolean Algebra

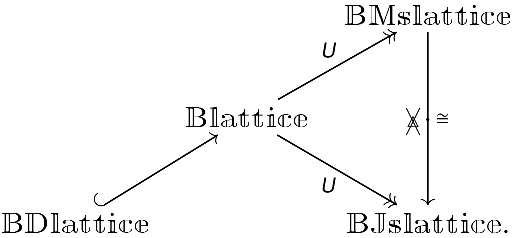
Definition

- *A* **Heyting algebra** *is a bounded distributive lattice with a binary operation $\Rightarrow$ which satisfies the following for all x, y, and z in P*

$$(x \wedge y) \leq z \qquad \text{if and only if} \qquad y \leq (x \Rightarrow z).$$

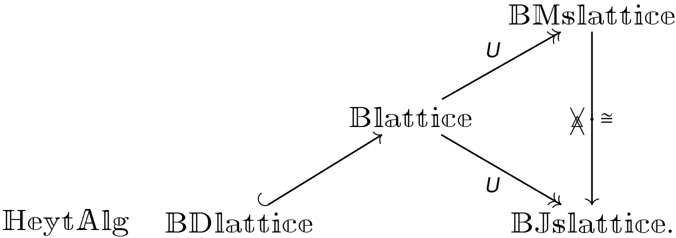
- *Categorically, this says that for every object x in the partial order, the map $x \wedge (\ \ )$ has a right adjoint $x \Rightarrow (\ \ )$.*
- *Later we name a category where the product has such a right adjoint a "Cartesian closed category."*
- *The category of all Heyting algebras and maps that preserves $\wedge$, $\vee$, $\Rightarrow$, 0, and 1 is denoted $\mathbb{HeytAlg}$.*
- *There is a forgetful functor $\mathbb{HeytAlg} \longrightarrow \mathbb{BDlattice}$.*

Towards the Definition of a Boolean Algebra

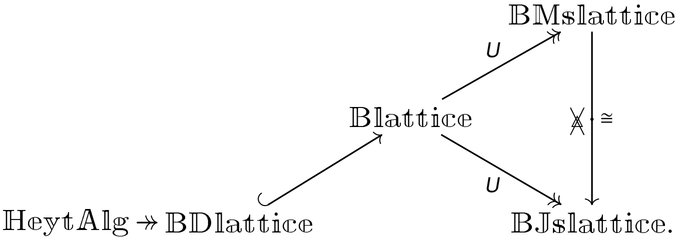


Notice that the right triangle need not commute.

Towards the Definition of a Boolean Algebra



Notice that the right triangle need not commute.



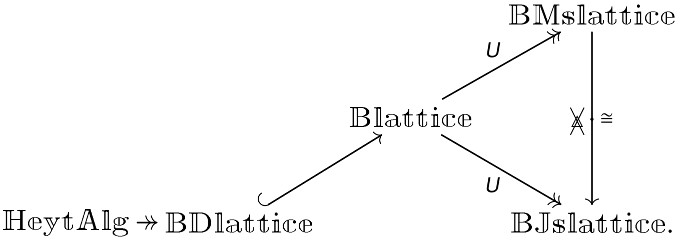
Notice that the right triangle need not commute.

Towards the Definition of a Boolean Algebra

Definition

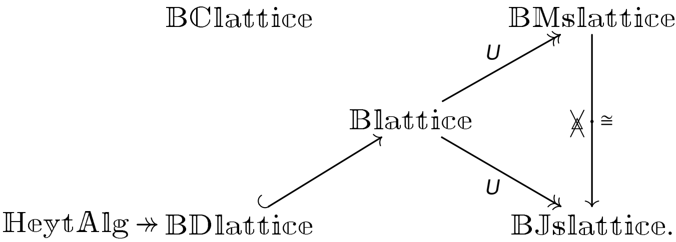
- *Within a bounded lattice, the* **complement** *of an element x is an element y such that $x \wedge y = 0$ and $x \vee y = 1$.*
- *If a complement of an element exists, then it is unique, and we denote the complement of x as x'.*
- *A bounded lattice where every element has a complement is called a* **complemented lattice**.
- *The category of all complemented lattices and maps that preserve the 0 and 1 and the operations is denoted $\mathbb{BClattice}$.*
- *There is a forgetful functor $\mathbb{BClattice} \longrightarrow \mathbb{Blattice}$.*

Towards the Definition of a Boolean Algebra



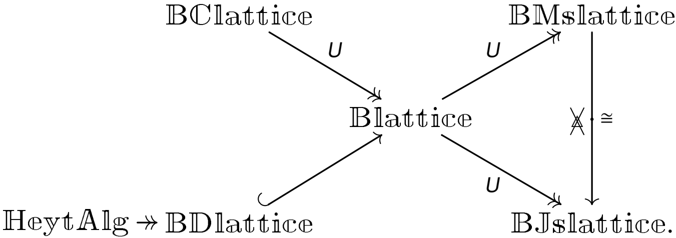
Notice that the right triangle need not commute.

Towards the Definition of a Boolean Algebra



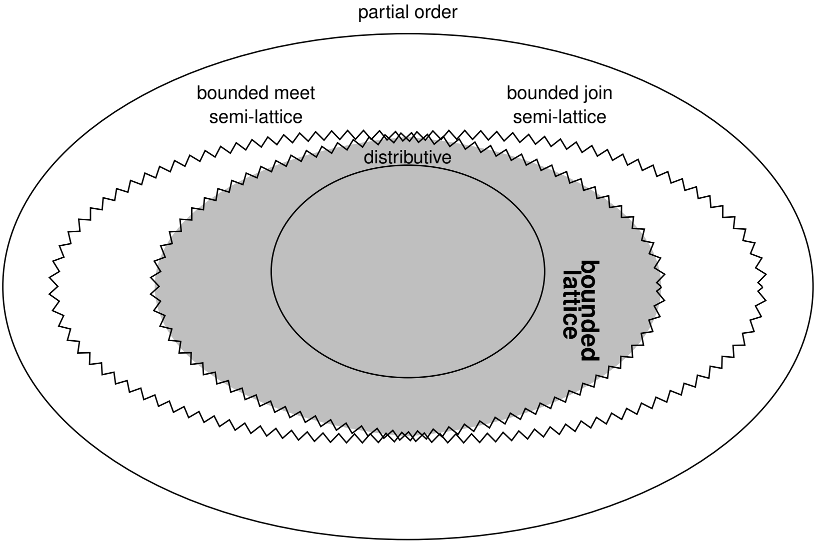
Notice that the right triangle need not commute.

Towards the Definition of a Boolean Algebra

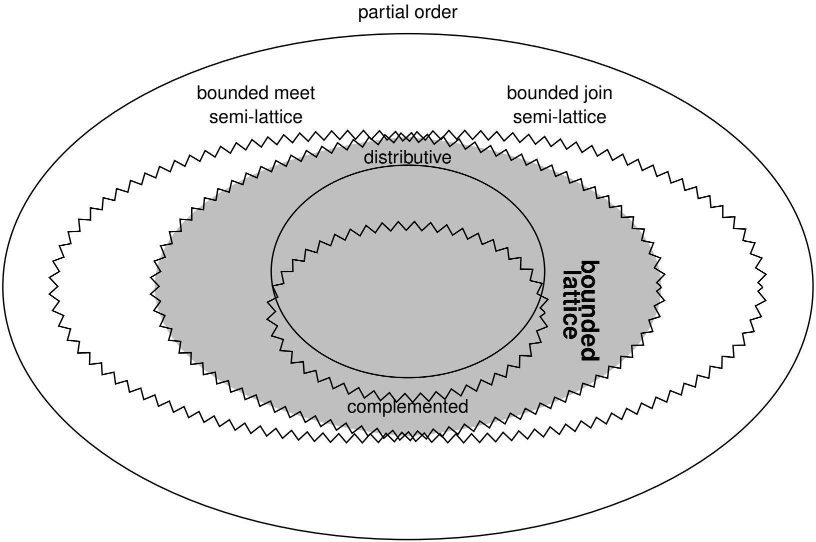


Notice that the right triangle need not commute.

Towards the Definition of a Boolean Algebra



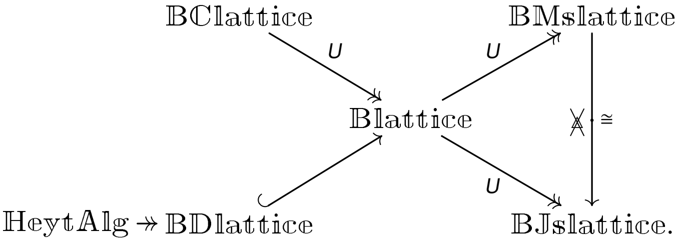
Towards the Definition of a Boolean Algebra



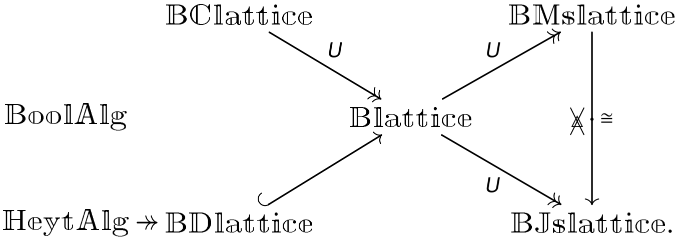
The Definition of a Boolean Algebra

Definition

- *A* **Boolean algebra** *is a complemented, distributive lattice.*
- *A map of Boolean algebras is a set function that preserve* $\wedge$, $\vee$, 0, *and* 1.
- *The category of all Boolean algebras and maps between them is denoted* $\mathbb{BoolAlg}$.
- *There are inclusions* $\mathbb{BoolAlg} \hookrightarrow \mathbb{BClattice}$ *and* $\mathbb{BoolAlg} \hookrightarrow \mathbb{BDlattice}$.
- *There is a full subcategory* $\mathbb{FinBoolAlg}$ *of Boolean algebras with only a finite number of elements.*
- *Every Boolean algebra can be seen as a Heyting algebra by defining* $x \Rightarrow y$ *as* $x' \vee y$. *This entails an embedding* $\mathbb{BoolAlg} \hookrightarrow \mathbb{HeytAlg}$.

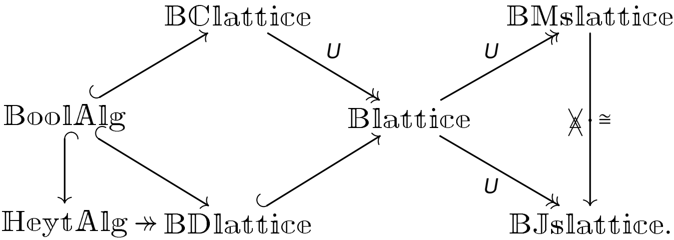


Notice that the right triangle need not commute.



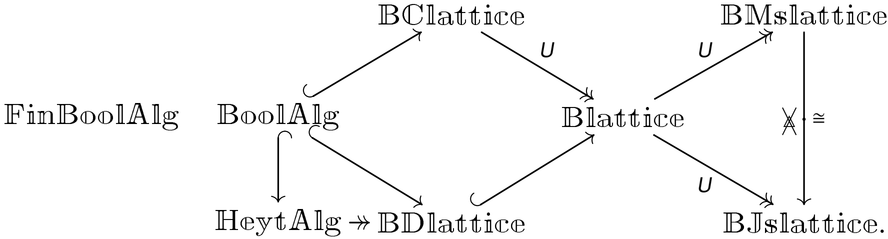
Notice that the right triangle need not commute.

The Definition of a Boolean Algebra



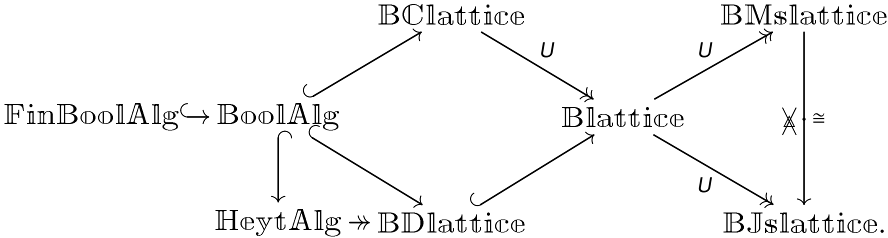
Notice that the right triangle need not commute.

The Definition of a Boolean Algebra



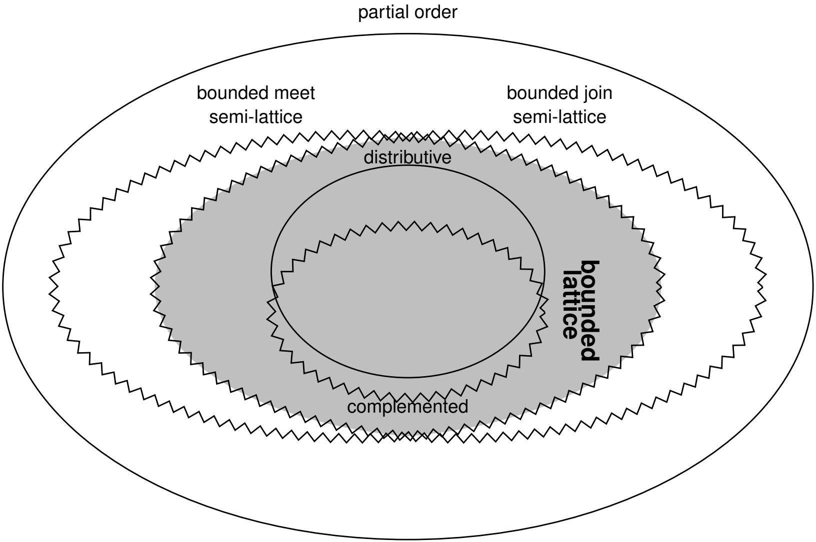
Notice that the right triangle need not commute.

The Definition of a Boolean Algebra

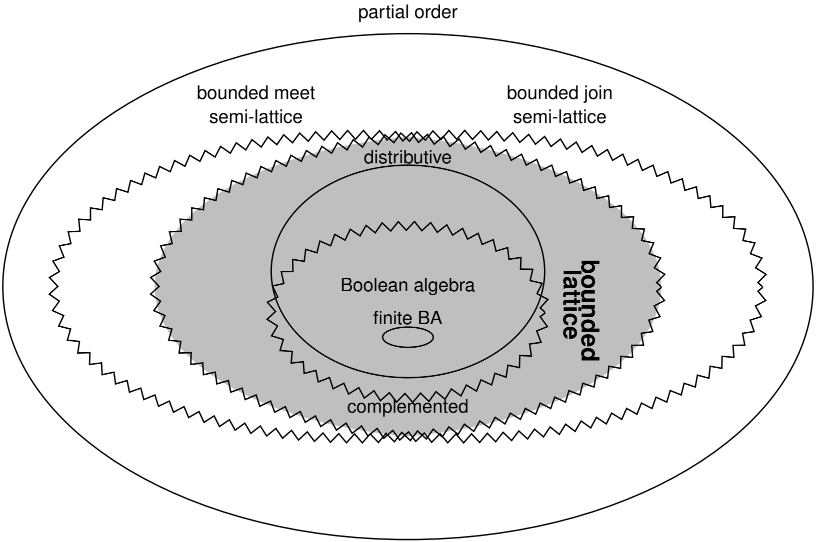


Notice that the right triangle need not commute.

The Definition of a Boolean Algebra



The Definition of a Boolean Algebra



The Definition of a Boolean Algebra

We started with notions of a partial order and we found algebraic operations. We can also go the other way and start by defining algebraic operations and show that these entail a partial order structure.

Definition

A **Boolean algebra** $(B, \wedge, \vee, (\ )', 0, 1)$ *is*

- *a set* B,
- *with the following operations* $\wedge\colon B \times B \longrightarrow B$ *and* $\vee\colon B \times B \longrightarrow B$, $(\ )'\colon B \longrightarrow B$,
- *and two constants* 0 *and* 1.
- *The* $\wedge$ *and* $\vee$ *are commutative and associative.*
- *All three operations are idempotent, i.e.* $x \wedge x = x$, $x \vee x = x$, *and* $x'' = x$.

The Definition of a Boolean Algebra

Theorem

The two definitions of Boolean algebras are equivalent.

Proof.

We saw that given the partial order we can form the meet and the join using products and coproducts respectfully. One can also go the other way: given the meet and the join, we determine the partial order by defining $x \leq y$ when $x \wedge y = x$. Equivalently, we can define $x \leq y$ when $x \vee y = y$. □

The Properties of a Boolean Algebra

Theorem

The following are some equations that a Boolean algebra satisfies:

(i) $(x \vee y)' = x' \wedge y'$	*(ii)* $(x \wedge y)' = x' \vee y'$	*(iii)* $x \wedge x = x$	*(iv)* $x \vee x = x$
(v) $x \wedge (x \vee y) = x$	*(vi)* $x \vee 1 = 1$	*(vii)* $x \wedge 1 = x$	*(viii)* $x \vee 0 = x$
(ix) $x \wedge (y \wedge z) = (x \wedge y) \wedge z$	*(x)* $x \wedge x' = 0$	*(xi)* $x \wedge y = y \wedge x$	*(xii)* $x \wedge 0 = 0$
(xiii) $x \vee (y \vee z) = (x \vee y) \vee z$	*(xiv)* $x \vee x' = 1$	*(xv)* $x'' = x$	*(xvi)* $0' = 1$.

Examples of Boolean Algebras

Example

- *The world's smallest example of a Boolean algebra is the partial order* $\{*\}$ *with one element. In this Boolean algebra* $0 = 1 = *$. *This Boolean algebra is the terminal object in* $\mathbb{BoolAlg}$.
- *The next smallest Boolean algebra is* $\{0, 1\}$ *with* $0 < 1$ *(i.e. category* **2** *with two objects and a nontrivial morphism from* 0 *to* 1*). We denote this Boolean algebra as* $\mathbf{2}_{BA}$. *The* $\wedge$ *operations is* $0 \wedge 0 = 1 \wedge 0 = 0 \wedge 1 = 0$, *and* $1 \wedge 1 = 1$. *The* $\vee$ *operation is* $1 \vee 0 = 0 \vee 1 = 1 \vee 1 = 1$, *and* $0 \vee 0 = 0$. *We have* $0' = 1$ *and* $1' = 0$. *This Boolean algebra is the initial object in* $\mathbb{BoolAlg}$.
- *For any set* S, *the powerset of* S, $\mathcal{P}(S)$, *is a partial order and a Boolean algebra. The operations are the intersection,* $\cup$, *the union,* $\cap$, *and the complement,* $(\ )^c$, *defined for* $T \subseteq S$ *as* $T^c = S - T$.

Examples of Boolean Algebras

Example

- *Another way to view the powerset of a set as a Boolean algebra is by thinking of characteristic functions. Let* $\mathbf{2}_{Set}$ *be the set with two elements* $\{0,1\}$*. For any set* S*, the* $\mathcal{P}(S) = Hom_{\mathbb{S}\mathrm{et}}(S, \mathbf{2}_{Set})$*. This Hom set inherits the structure of a Boolean algebra from* $\mathbf{2}_{Set}$*. If* $f\colon S \longrightarrow \mathbf{2}_{Set}$ *and* $g\colon S \longrightarrow \mathbf{2}_{Set}$*, then* $(f \wedge g)\colon S \longrightarrow \mathbf{2}_{Set}$ *is defined as* $(f \wedge g)(s) = f(s) \wedge g(s)$ *for all* $s \in S$*. The* $\wedge$ *operation is defined on* $\mathbf{2}_{Set}$ *as* $\mathbf{2}_{BA}$*. There is a similar definition for* $\vee$*. Function* $f'\colon S \longrightarrow \mathbf{2}_{Set}$ *is defined as* $f'(s) = 1 - f(s)$*.*
- *For any natural number* n*, the set* $(\mathbf{2}_{BA})^n$ *(the product of* n *copies of* $\mathbf{2}_{BA}$*) is also a Boolean algebra. The operations are done point-wise. For example,* $(0,1,1,0,1) \vee (1,1,0,0,0) = (1,1,1,0,1)$ *and* $(0,1,1,0,1)' = (1,0,0,1,0)$*. Notice that* $(\mathbf{2}_{BA})^n$ *is the same as* $Hom(\{1,2,\ldots,n\}, \mathbf{2}_{Set})$*.*

Examples of Boolean Algebras

Example

- *Let N be a square free number, i.e., a number whose prime decomposition does not have a prime number that is squared or any higher power. For example,* $105 = 3 \cdot 5 \cdot 7$ *or* $715 = 5 \cdot 11 \cdot 13$. *The set of all the factors of N forms a Boolean algebra. The $\wedge$ is the highest common factor. For example, if $N = 105$, then $15 \wedge 35 = (3 \cdot 5) \wedge (5 \cdot 7) = 5$. The $\vee$ is the least common multiple. For example, if $N = 105$, then $15 \vee 35 = (3 \cdot 5) \vee (5 \cdot 7) = 105$. If x is a factor, then x' is the product of all the prime factors not in x. For example, if $N = 105$ then $3' = 5 \cdot 7 = 35$. If $N = 715$, then $11' = 5 \cdot 13 = 65$. The 0 of this Boolean algebra is the factor 1. The 1 of this Boolean algebra is N. In essence, this example can be seen as the Boolean algebra $\mathcal{P}(S)$ where S is the set of prime factors of N.*

Examples of Boolean Algebras

Example

- *In a topological space, a* **closed set** *is a set whose complement is an open set. A set is called* **closed-open set** *or a* **clopen set** *if it is both open and closed, i.e., both the set and its complement are open. Consider the set of clopen sets of any topological space. Using DeMorgan's law, one can see that the union and intersection of clopen sets are clopen. Similarly, the complements of clopen sets are clopen. The empty set is clopen and is the* 0. *The whole topological space is clopen and is the* 1. *We conclude that the clopen sets of a topological space form a Boolean algebra.*

Examples of Boolean Algebras

Example

- *If you know about* **finite automata** *(sometimes called* **finite state machines***), then you know that* **regular languages** *are the languages that finite automata recognize. The collection of all regular languages forms a Boolean algebra. This is because the intersection, union, and complement of regular languages are regular languages. The* 0 *of this Boolean algebra is the empty language, and the* 1 *is* Σ^**, the language consisting of all words in the alphabet.*

The Properties of a Boolean Algebra

- Before we start exploring Stone duality, there is another duality associated with Boolean algebras.
- The **duality principle** says that any true statement about Boolean algebras is also true when all the joins are swapped for meets, and the 0s are swapped for 1s.
- This duality comes about when we swap a partial order $\leq$ with its reverse partial order $\leq^R$ where $x \leq^R y$ iff $y \leq x$.
- Such a change in the partial order swaps $\wedge$ for $\vee$ and swaps 0 for 1.
- Categorically, this means that there is a functor $D\colon \mathbb{BoolAlg} \longrightarrow \mathbb{BoolAlg}$ that takes every partial order $(P, \leq)$ to $(P, \leq^R)$.

The Properties of a Boolean Algebra

Theorem

The same colored boxes are dual to each other.

(i) $(x \vee y)' = x' \wedge y'$	(ii) $(x \wedge y)' = x' \vee y'$	(iii) $x \wedge x = x$	(iv) $x \vee x = x$
(v) $x \wedge (x \vee y) = x$	(vi) $x \vee 1 = 1$	(vii) $x \wedge 1 = x$	(viii)$x \vee 0 = x$
(ix) $x \wedge (y \wedge z) = (x \wedge y) \wedge z$	(x) $x \wedge x' = 0$	(xi) $x \wedge y = y \wedge x$	(xii) $x \wedge 0 = 0$
(xiii) $x \vee (y \vee z) = (x \vee y) \vee z$	(xiv) $x \vee x' = 1$	(xv) $x'' = x$	(xvi) $0' = 1$.

Baby Stone Duality

Theorem

The opposite of the category of finite Boolean algebras is equivalent to the category of finite sets. That is, $\mathbb{FinBoolAlg}^{op} \simeq \mathbb{FinSet}$.

Baby Stone Duality

Proof.

- There is a functor $\mathcal{P}\colon \mathbb{FinSet} \longrightarrow \mathbb{FinBoolAlg}^{op}$ that takes a finite set S to the powerset of S, $\mathcal{P}(S)$, which is a finite Boolean algebra.
- This is the contravariant powerset functor which takes the set function $f\colon S \longrightarrow T$, to $\mathcal{P}(f)\colon \mathcal{P}(T) \longrightarrow \mathcal{P}(S)$.
- Such functions respect all the operations of a Boolean algebra. For example, if X and Y are subsets of T, then

$$\begin{aligned}\mathcal{P}(f)(X \cap Y) &= \{x \in S : f(x) \in X \cap Y\} \\ &= \{x \in S : f(x) \in X\} \cap \{x \in S : f(x) \in Y\} \\ &= \mathcal{P}(f)(X) \cap \mathcal{P}(f)(Y).\end{aligned}$$

- The target of this functor is $\mathbb{FinBoolAlg}^{op}$.

□

Proof.

- There is an **atom functor** $At : \mathbb{FinBoolAlg}^{op} \longrightarrow \mathbb{FinSet}$ that is the quasi-inverse of $\mathcal{P}$.
- First a definition: within a Boolean algebra, an element a is called an **atom** if there is nothing smaller than it other than 0.
- Formally, a is an atom if $0 \lneq a$ and for all x, if $x \leq a$, then $x = a$ or $x = 0$. If you think of a Boolean algebra as a type of lattice with 1 on the top and 0 on the bottom, then the set of atoms are those elements right above 0.
- For any finite Boolean algebra, every element b is made up of the finite join of all the atoms that are less than b. In symbols,

$$b = \bigvee \{x : x \text{ is an atom, and } x \leq b\}.$$

The functor At takes a finite Boolean algebra B to $At(B)$, the finite set of atoms of B.

Proof.

- These two functors form the equivalence stated in the theorem.
- For a finite Boolean algebra B, there is an isomorphism $\phi\colon B \longrightarrow \mathcal{P}(At(B))$. In detail, $\mathcal{P}(At(B))$ is the powerset of atoms of B. The isomorphism ϕ is defined as

$$b \quad \mapsto \quad \{x : x \text{ is an atom, and } x \leq b\}.$$

- The map ϕ has the following properties:
  - ϕ is a surjection.
  - ϕ is an injection.
  - ϕ preserves 0 and 1.

□

Proof.

- For every finite set S, there is an isomorphism $\psi\colon S \longrightarrow At(\mathcal{P}(S)) = \{\{s\} : s \in S\}$.
- This isomorphism is the function defined by

$$s \in S \qquad \mapsto \qquad \{s\} \in At(\mathcal{P}(S)).$$

- It is easy to see that this function is an injection and a surjection of sets.

$\square$

Baby Stone Duality

- This duality theorem shows that every finite Boolean algebra has 2^n elements for some n.
- Any two finite Boolean algebras of the same size are isomorphic.
- The way to see this is that if there are two Boolean algebras, B_1 and B_2, both of size 2^n, then there are two sets of atoms, S_1 and S_2, each of size n. Two finite sets of the same size entail a set isomorphism $f\colon S_1 \longrightarrow S_2$. The functor $\mathcal{P}$ takes isomorphisms to isomorphisms. By composing the isomorphsims as follows

$$B_2 \longrightarrow \mathcal{P}(S_2) \longrightarrow P(S_1) \longrightarrow B_1,$$

we have shown that the Boolean algebras are isomorphic.

Baby Stone Duality

- Before we move on to the next step, it will be useful to look at the functors from a more categorical prospective.
- The functor $\mathcal{P}$ is defined as $\mathcal{P}(S) = Hom_{\mathbb{Set}}(S, \mathbf{2}_{Set})$ where $\mathbf{2}_{Set}$ is the set with two elements.
- The functor At can also be seen from a more categorical prospective. Consider an atom a in Boolean algebra B. We can view this atom as a Boolean algebra map $f_a : B \longrightarrow \mathbf{2}_{BA}$ defined as

$$f_a(b) = \begin{cases} 1 & : a \leq b \\ 0 & : a \not\leq b. \end{cases}$$

- In words, f_a takes all those elements above a to 1 and the rest to 0. Thus, the set of atoms $At(B)$ for a finite Boolean algebra B can be seen as a subset of $Hom_{\mathbb{FinBoolAlg}}(B, \mathbf{2}_{BA})$. We will generalize this soon.

The Big Picture of Stone Duality

Algebra **Topology**

$\simeq$

$\simeq$

$\simeq$

$\simeq$

$$\mathbb{FinBoolAlg}^{op} \underset{\mathcal{P}}{\overset{At}{\rightleftarrows}} \mathbb{FinSet} \quad (\simeq)$$

The Big Picture of Stone Duality

Algebra **Topology**

$\simeq$

$\simeq$

$\simeq$

$\simeq$

$$\mathbb{F}\mathrm{in}\mathbb{B}\mathrm{ool}\mathbb{A}\mathrm{lg}^{op} \underset{\mathcal{P}}{\overset{At}{\rightleftarrows}} \mathbb{F}\mathrm{in}\mathbb{S}\mathrm{et} \quad (\simeq)$$

- While finite sets are fine, let us generalize to all sets.
- In order to do that we will need to explore beyond finite Boolean algebras.
- The powerset of an infinite set is an infinite Boolean algebra.
- What other properties will the powerset of an infinite set have?

Definition

- *Taking the meets and joins a finite number of times gives us all finite meets and joins. A Boolean algebra that has arbitrary (not just finite) meets and joins is called* **complete***. Categorically, this says the partial order category has all products and coproducts. If X is an arbitrary subset of elements of a Boolean algebra, then we write $\bigvee_{x \in X} x$ for their join, and $\bigwedge_{x \in X} x$ for their meet.*
- *A Boolean algebra is* **atomic** *if every element is the join of a set of its atoms (0 is the join of the empty set of atoms).*
- *The collection of all complete, atomic Boolean algebras and Boolean algebra homomorphisms is denoted $\mathbb{CABoolAlg}$ Thus we have the following embeddings:*

$$\mathbb{FinBoolAlg} \hookrightarrow \mathbb{CABoolAlg} \hookrightarrow \mathbb{BoolAlg}.$$

Juvenile Stone Duality

Theorem

The opposite of the category of complete atomic Boolean algebras is equivalent to the category of sets, that is $\mathbb{CABoolAlg}^{op} \simeq \mathbb{Set}$.

Juvenile Stone Duality

Proof.

- The proof follows almost exactly like the previous proof.
- There is a functor $\mathcal{P}\colon \mathbb{Set} \longrightarrow \mathbb{CABoolAlg}^{op}$ which is defined for a set S as $\mathcal{P}(S) = Hom_{\mathbb{Set}}(S, \mathbf{2}_{Set})$. For any set S — finite or infinite — the powerset $\mathcal{P}(S)$ is complete because it has arbitrary large unions and intersections. It is also atomic because the singletons are atoms, and every set is the arbitrary union of its single elements. Thus we have that the powerset of any set is a complete atomic Boolean algebra.
- There is a functor $At\colon \mathbb{CABoolAlg}^{op} \longrightarrow \mathbb{Set}$ that takes any complete atomic Boolean algebra to its set of atoms.
- For a complete atomic Boolean algebra B, there is an isomorphism $\phi\colon B \longrightarrow \mathcal{P}(At(B))$.
- For an arbitrary set S, there is an isomorphism $\psi\colon S \longrightarrow At(\mathcal{P}(S)) = \{\{s\} : s \in S\}$.

□

Juvenile Stone Duality

Remark

- *A consequence of this theorem is that we now have a way of thinking about the category* $\mathbb{Set}^{op}$*.*
- *This category is not a typical category (technically called a* **concrete category***) where every morphism corresponds to some type of function.*
- *In* $\mathbb{Set}$ *there is a unique function* $\emptyset \longrightarrow \{*\}$ *which corresponds to a unique map* $\{*\} \longrightarrow \emptyset$ *in* $\mathbb{Set}^{op}$*. What can this morphism mean?*
- *It takes the element* $*$ *to where? By meditating on our theorem which says* $\mathbb{CABoolAlg} \simeq \mathbb{Set}.^{op}$ *we can think of* $\mathbb{Set}^{op}$ *as follows: the objects are sets and a morphism* $S \longrightarrow T$ *corresponds to the Boolean morphisms* $\mathcal{P}(S) \longrightarrow \mathcal{P}(T)$*.*
- *In particular, the morphism* $\{*\} \longrightarrow \emptyset$ *in* $\mathbb{Set}^{op}$ *corresponds to the Boolean algebra homomorphism* $\mathcal{P}(\emptyset) = \{*\} \longrightarrow \mathcal{P}(\{*\}) = \{0, 1\}$ *with* $* \mapsto 1$*.*

Remark

- *This way of understanding* $\mathbb{Set}^{op}$ *provides an important idea useful in understanding the rest of this mini-course.*
- *A morphism* $f_x \colon \{*\} \longrightarrow S$ *in* $\mathbb{Set}$ *(and* $\mathbb{Top}$*) picks out the element* x *in* S*.*
- *Corresponding to* f_x *in* $\mathbb{Set}$ *is a morphism* $\bar{f}_x \colon S \longrightarrow \{*\}$ *in* $\mathbb{Set}^{op}$ *and a Boolean algebra homomorphism* $\hat{f}_x \colon \mathcal{P}(S) \longrightarrow \mathcal{P}(\{*\}) = \{0, 1\}$.
- *The homomorphism* $\hat{f}_x$ *is a characteristic function that picks out those subsets of* S *(elements of* $\mathcal{P}(S)$*) that contain* x*.*
- *This idea — that maps picking out an element are equivalent to maps which determine if that element is in a subset — is absolutely central to Stone duality and all of its generalizations.*

Juvenile Stone Duality

- Before we move on, it pays to think of sets from a more general point of view.
- Let S and T be sets. We can think of these sets as topological spaces with the discrete topologies (S, σ_d) and (T, τ_d).
- From this perspective, it is important to notice that $Hom_{\mathbb{Set}}(S, T) = Hom_{\mathbb{Top}}((S, \sigma_d), (T, \tau_d))$. In other words, one can think of sets as special types of topological spaces where every map between the topological spces is considered continuous.
- This entails an inclusion functor $\mathbb{Set} \hookrightarrow \mathbb{Top}$. Another way to say this is that the category of sets is equivalent to the subcategory of topological spaces where all the topologies are discrete. Notice also that when using the discrete topology, all open sets are also closed sets, i.e., every open set is clopen. We will continue our journey using the language of topology rather than sets.

Algebra | **Topology**

$\simeq$

$\simeq$

$\simeq$

$\simeq$

$$\mathbb{FinBoolAlg}^{op} \underset{\mathcal{P}}{\overset{At}{\underset{\simeq}{\rightleftarrows}}} \mathbb{FinSet}$$

The Big Picture of Stone Duality

Algebra		**Topology**
	$\simeq$	
	$\simeq$	
	$\simeq$	
$\mathbb{CABoolAlg}^{op}$	$\simeq$	$\mathbb{Set}$

$$\mathbb{FinBoolAlg}^{op} \underset{\mathcal{P}}{\overset{At}{\rightleftarrows}} \mathbb{FinSet} \qquad (\simeq)$$

The Big Picture of Stone Duality

Algebra | **Topology**

$\simeq$

$\simeq$

$\simeq$

$$\mathbb{CABoolAlg}^{op} \underset{\mathcal{P}}{\overset{At}{\rightleftarrows}} \mathbb{S}\mathrm{et} \quad (\simeq)$$

$$\mathbb{FinBoolAlg}^{op} \underset{\mathcal{P}}{\overset{At}{\rightleftarrows}} \mathbb{FinSet} \quad (\simeq)$$

Algebra		**Topology**
	$\simeq$	
	$\simeq$	
	$\simeq$	

$$
\begin{array}{ccc}
\mathbb{CABoolAlg}^{op} & \underset{\mathcal{P}}{\overset{At}{\rightleftarrows}} \; \simeq & \mathbb{Set} \\
\uparrow & & \uparrow \\
\mathbb{FinBoolAlg}^{op} & \underset{\mathcal{P}}{\overset{At}{\rightleftarrows}} \; \simeq & \mathbb{FinSet}
\end{array}
$$

Stone Duality

Let us generalize from complete atomic Boolean algebras to all Boolean algebras. To do that, we need to go beyond sets. We define special types of topological spaces.

Definition

- *A topological space is* **compact** *if it has a type of finiteness condition. Formally, a topological space T is compact if every collection $C = \{V_i\}$ of open sets of T which has the property that the union of those open sets is equal to the entire T, has a finite subset $\bar{C} \subset C$ such that the union of all the open sets in $\bar{C}$ is also equal to the entire space T. Intuitively, this says that the topological space does not need many open sets to cover it. Notice that an infinite topological space with the discrete topology is not compact because the cover consisting of the infinite set of elements does not have a finite subcover.*

Definition

- *A topological space is* **Hausdorff** *if any two points in the space can be separated by two disjoint open sets. Formally, a topological space T is Hausdorff if for any two different points x and y, there exists open sets A and B such that $x \in A$ and $y \in B$ with $A \cap B = \emptyset$. Intuitively, a Hausdorff space has a lot of open sets to separate points.*
- *A topological space is* **totally disconnected** *if the connected components in are the one-point sets.*
- *A topological space is a* **Stone space** *if it is compact, Hausdorff, and totally disconnected.*

The category of Stone spaces and continuous maps between them is denoted $\mathbb{Stone}$.

Theorem

The opposite of the category of Boolean algebras is equivalent to the category of Stone spaces, that is $\mathbb{BoolAlg}^{op} \simeq \mathbb{Stone}$.

Stone Duality

Proof.

- There is a functor $Clp\colon \mathbb{Stone} \longrightarrow \mathbb{BoolAlg}^{op}$ which is defined for Stone space T as $Clp(T) = Hom_{\mathbb{Stone}}(T, \mathbf{2}_{SS})$ where $\mathbf{2}_{SS}$ is the two-object Stone space $\{0, 1\}$.
- Within $\mathbf{2}_{SS}$, both $\{0\}$ and $\{1\}$ are open (and hence closed) sets.
- The functor *Clp* takes any Stone space to the Boolean algebra of its clopen sets.
- There is a functor $Ulf\colon \mathbb{BoolAlg}^{op} \longrightarrow \mathbb{Stone}$ which is defined for a Boolean algebra B as $Ulf(B) = Hom_{\mathbb{BoolAlg}}(B, \mathbf{2}_{BA})$. This Hom set is a Stone space.

□

In order to ensure that the reader can better communicate with poor souls who do not already know category theory, we will describe these maps in a non-categorical language.

Definition

For a Boolean algebra map $f\colon B \longrightarrow \mathbf{2}_{BA}$*, the set of elements that go to* 1*, i.e.* $F = f^{-1}(1)$*, form a structure called an* **ultrafilter***. Following Remark* **??***, we might think of these ultrafilters as types of elements of* B*. The set* F *satisfies the following list of properties:*

- *F is a* **filter** *because*
  - *F is non-empty: for example,* $1 \in F$ *(because* $f(1) = 1$*.)*
  - *F is upward closed: if* $a \in F$ *and* $a \leq b$*, then* $b \in F$ *(because* f *is order preserving.)*
  - *F is closed under meet: if* $a \in F$ *and* $b \in F$*, then* $a \wedge b \in F$ *(because* f *preserves the meet.)*

Definition

- *F is* **proper filter***: F does not contain everything, i.e., $F \neq B$ or equivalently $0 \notin F$ (because $f(0) = 0$).*
- *F is* **prime filter***: If $a \vee b \in F$, then $a \in F$ or $b \in F$ (because $f(a \vee b) = f(a) \vee f(b)$).*
- *F is* **maximal filter***. For all $a \in B$, either $a \in F$ or $a' \in F$ (because $f(a') = f(a)'$, so either $f(a) = 1$ or $f(a') = 1$).*

For each $a \in B$ there is a special filter called a **principle filter generated by** *a which is $\{b \in B : a \leq b\}$. This corresponds to the map $f_a : B \longrightarrow \mathbf{2}_{BA}$. For a Boolean algebra, the principle filter generated by a is an ultrafilter if and only if a is an atom.*

Algebra **Topology**

$\simeq$

$\simeq$

$\simeq$

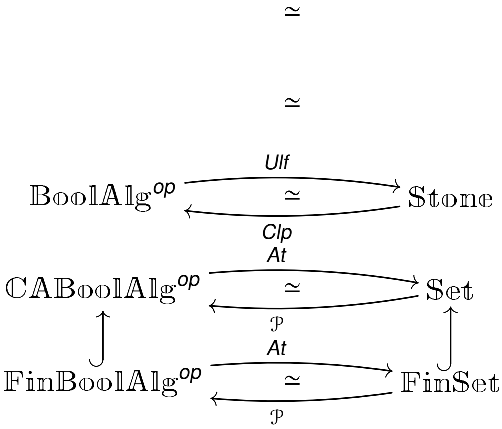
$$
\begin{array}{ccc}
\mathbb{CABoolAlg}^{op} & \underset{\mathcal{P}}{\overset{At}{\rightleftarrows}} \; \simeq & \mathbb{Set} \\
\uparrow & & \uparrow \\
\mathbb{FinBoolAlg}^{op} & \underset{\mathcal{P}}{\overset{At}{\rightleftarrows}} \; \simeq & \mathbb{FinSet}
\end{array}
$$

The Big Picture of Stone Duality

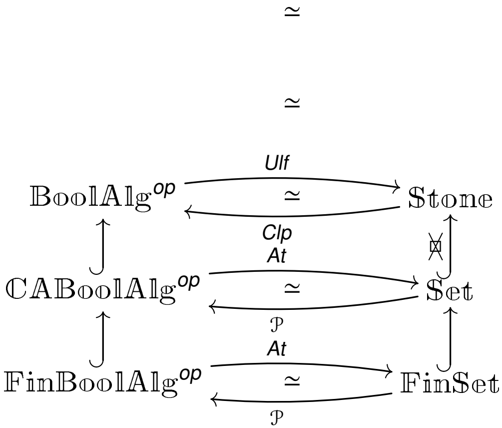
Algebra		**Topology**
	$\simeq$	
	$\simeq$	
$\mathbb{BoolAlg}^{op}$	$\simeq$	$\mathbb{Stone}$
$\mathbb{CABoolAlg}^{op}$	$\underset{\mathcal{P}}{\overset{At}{\simeq}}$	$\mathbb{Set}$
$\uparrow$		$\uparrow$
$\mathbb{FinBoolAlg}^{op}$	$\underset{\mathcal{P}}{\overset{At}{\simeq}}$	$\mathbb{FinSet}$

The Big Picture of Stone Duality

Algebra **Topology**

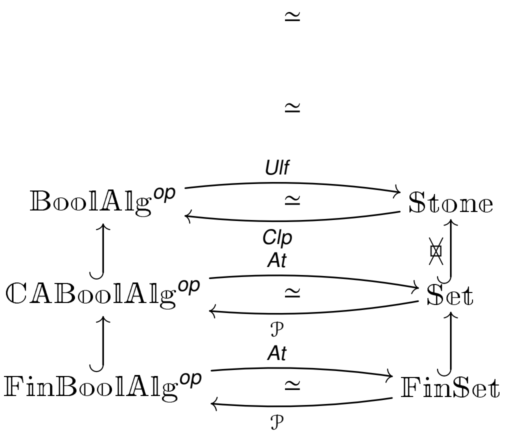


Algebra **Topology**



The Big Picture of Stone Duality

Algebra **Topology**



The second square from the bottom doesn't commute.

About Stone Duality

Remark

- *We have seen the object* **2** *in many different contexts. (See Important Categorical Idea.)*
- *It is the collection with two objects that form a set, a partial order, a Boolean algebra, and a Stone space. There will be more soon.*
- *We distinguish the various* **2***'s with subscripts as $\mathbf{2}_{Set}$, $\mathbf{2}_{BA}$ and $\mathbf{2}_{SS}$.*
- *Such an structure that has various incarnations in different categories is called a* **dualizing object** *or a* **schizophrenic object**. *The duality depends on such dualizing objects.*
- *It is again the power of category theory to see many different duality theorems as coming from one idea. The point of these dualizing objects is that category theory does not neatly separate different structures in different areas. Rather these objects connect and unify different areas.*

About Stone Duality

Remark

- *These proofs of the isomorphisms of the duality theorem are very similar to the proof where we showed a finite dimensional vector space is naturally isomorphic to its double dual.*
- *That is, for a finite dimensional vector space V, there is an isomorphism*

$$V \longrightarrow Hom_{\mathbf{K}\mathbb{FDVect}}(Hom_{\mathbf{K}\mathbb{FDVect}}(V, \mathbf{K}), \mathbf{K}).$$

- *Once again category theory shows connections between disparate areas. The ideas of Stone duality take us back to where Eilenberg and Mac Lane started category theory.*

About Stone Duality

History

- *This fact that every Boolean algebra is isomorphic to some collection of sets of elements (subsets or clopens) is called* **Stone's representation theorem**.
- *It was proved by Marshall H. Stone in 1936 (before category theory existed). It is the main theorem of this field.*
- *One of the consequences of the theorem is that one can easily check some fact about Boolean algebras by simply testing them on collections of a set.*

About Stone Duality

Marshall H. Stone

(1903-1989)

Esakia Duality

- Let us generalize from Boolean algebras to Heyting algebras.
- This was done by Leo Esakia in 1974.
- We will now deal with partial orders that are not necessarily complemented but have an $\Rightarrow$ operation.
- In order to deal with Heyting algebras, we have to be concerned with topological spaces with more structure. The extra idea needed is an ordering.

Esakia Duality

Definition

- *Given a partial order $(P, \leq)$, an* **upward-closed set** *X is a set such that for all $x \in X$, we have $x \leq y$ implies $y \in X$.*
- *An* **ordered topological space** *$(T, \tau, \leq)$ is a set T, a topology τ on the set T, and a partial order $\leq$ on the set T.*
- *An* **Esakia space** *is a Stone space such that*
  - *upward-closed sets are closed: for all $x \in T$, $\uparrow x = \{y \in T : x \leq y\}$ is closed, and*
  - *downward-clopen sets are clopen: for all clopen $C \subseteq X$, the set*

    $$\downarrow C = \{y \in T : y \leq x \text{ for some } x \in C\}$$

    is clopen.
- *The category of Esakia spaces and their morphisms is denoted $\mathbb{Esakia}$. There is an inclusion $\mathbb{Stone} \hookrightarrow \mathbb{Esakia}$.*

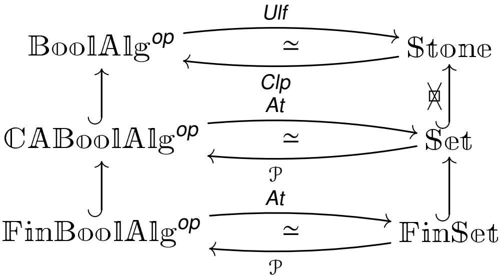
Theorem

The opposite of the category of Heyting algebras is equivalent to the category of Esakia spaces, that is $\mathbb{HeytAlg}^{op} \simeq \mathbb{Esakia}$.

Algebra **Topology**

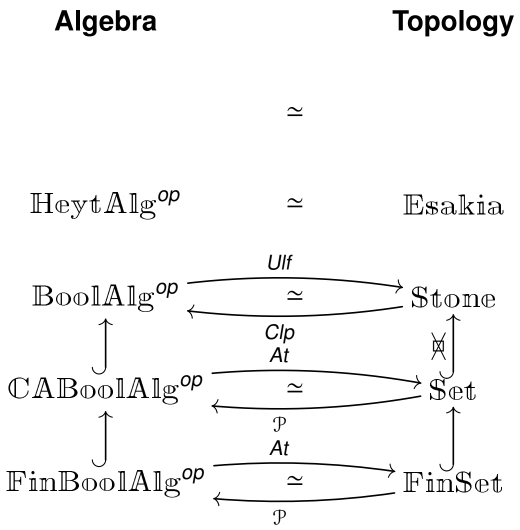
$\simeq$

$\simeq$



The second square from the bottom doesn't commute.

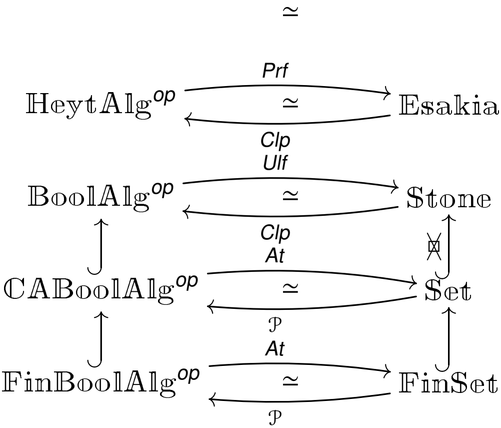
The Big Picture of Stone Duality



The second square from the bottom doesn't commute.

The Big Picture of Stone Duality

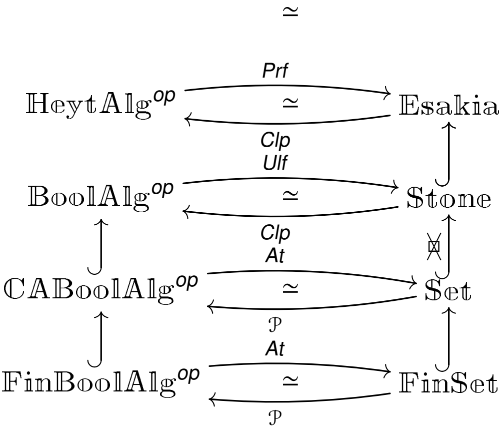
Algebra **Topology**



The second square from the bottom doesn't commute.

The Big Picture of Stone Duality

Algebra **Topology**



The second square from the bottom doesn't commute.

Priestley Duality

- Let us generalize from Heyting algebras to bounded distributive lattices.
- This was done by Hilary Priestley in the early 1970's.
- In doing this, we deal with partial orders that do not even have a $\Rightarrow$ operation.

Definition

- *An ordered topological space is a* **Priestley space** *if it satisfies the* **Priestley separation axiom***:*

  If $x \not\leq y$, then there exists a clopen up-set U such that $x \in U$ and $y \notin U$.

- *A morphism between two Priestley spaces is continuous and order-preserving.*
- *The collection of all Prieslty spaces and their morphisms form the category $\mathbb{Priestley}$.*
- *There is an inclusion $\mathbb{Esakia} \hookrightarrow \mathbb{Priestley}$.*

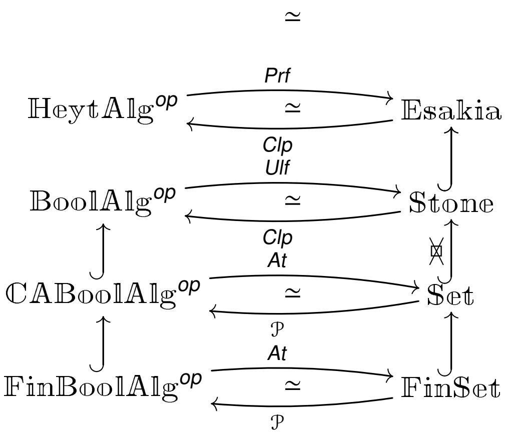
Priestley Duality

Theorem

The opposite of the category of bounded distributive lattice is equivalent to the category of Priestley spaces, that is $\mathbb{BDlattice}^{op} \simeq \mathbb{Priestley}$.

The Big Picture of Stone Duality

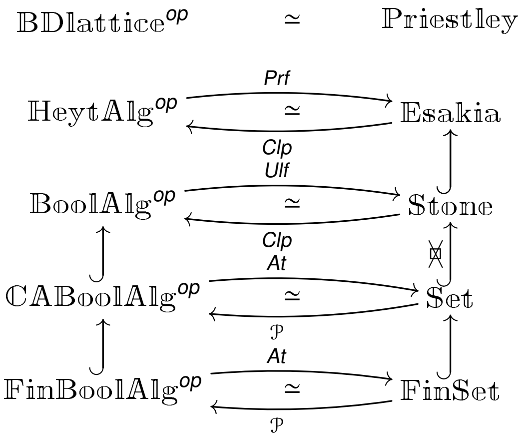
Algebra **Topology**



The second square from the bottom doesn't commute.

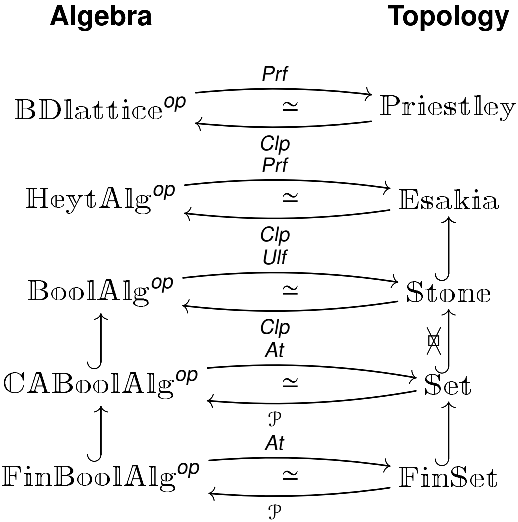
The Big Picture of Stone Duality

Algebra **Topology**



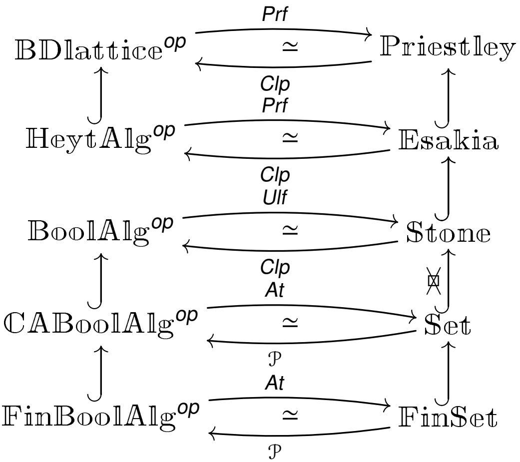
The second square from the bottom doesn't commute.

The Big Picture of Stone Duality



The second square from the bottom doesn't commute.

Algebra **Topology**



The second square from the bottom doesn't commute.

Pointless Topology

- Stone duality is not the only duality that shows a relationship between algebraic and topological ideas.
- Pointless topology is related to Stone duality and shows another relationship.
- A topological space is a set of points with open sets that satisfy certain requirements.
- The central question: To what extent are the properties of a topological space determined by the structure of the open sets (while ignoring the points)?
- This gives us the humorous title of “pointless topology.”

- There is a functor $\mathcal{O}$ that takes every topological space (T, τ) to the partial order $\mathcal{O}(T)$.
- This partial order reflects the structure of the subsets of the topological space with the meet, $\wedge$, being the intersection of open sets, and the join, $\vee$, being the union of open sets.
- As such, the partial order has finite meets and all joins. The partial order also satisfies a distributivity requirement.
- Let us make a formal definition of such a partial order.

Pointless Topology

Definition

- *A* **locale** L *is a partial order that has all finite meets and arbitrary joins. Furthermore, there is a requirement that an infinite distributive law is satisfied:*
    - *for any element* $U \in L$ *and any set of elements* $\{V_i\}$ *in* L*, we have*
      $$U \wedge \bigvee_i V_i = \bigvee_i (U \wedge V_i).$$
- *Morphisms between locales are interesting. We want such morphisms to mimic continuous maps between topological spaces. Remember that* $f\colon T_1 \longrightarrow T_2$ *is a continuous map of topological spaces if* f^{-1} *takes open sets to open sets. The important point is that* f^{-1} *goes the other way. With this in mind, we define a morphism of locales* $f\colon L_1 \longrightarrow L_2$ *to be a map such that* f^{-1} *preserves finite meets and all joins.*
- *The collection of locales and locale maps forms a category* $\mathbb{Locale}$.

Definition

- *A better way to deal with the awkwardness of the directions of the maps is to discuss the opposite of the category of locales. Define a* **frame** *exactly as a locale was defined.*
- *A morphism of frames* $f\colon F_1 \longrightarrow F_2$ *is a map of partially ordered sets that preserves finite meets and all joins.*
- *The collection of all frames and frame morphisms is denoted* $\mathbb{F}\mathrm{rame}$*, and by definition* $\mathbb{L}\mathrm{ocale} = \mathbb{F}\mathrm{rame}^{op}$*.*

Now for the duality of the topological structures and the partial orders.

Theorem

There is an adjunction between the category of locales — which is the opposite of the category of frames — and the category of topological spaces.

$$\mathbb{F}\mathrm{rame}^{op} \underset{\mathcal{O}}{\overset{Pt}{\rightleftarrows}} \mathbb{T}\mathrm{op} \qquad (\top)$$

Pointless Topology

Proof.

- The functor $\mathcal{O}\colon \mathbb{Top} \longrightarrow \mathbb{Frame}^{op}$ takes every topological space (T, τ) to $\mathcal{O}(T)$, the frame of its open sets.
- Every map of topological spaces $f\colon (T_1, \tau_1) \longrightarrow (T_2, \tau_2)$ goes to the frame map $f^{-1}\colon \mathcal{O}(T_2) \longrightarrow \mathcal{O}(T_1)$.
- The functor that goes the other way picks out the set of points of a frame $Pt\colon \mathbb{Frame}^{op} \longrightarrow \mathbb{Top}$.
- This functor takes every frame F to the topological space $Pt(F)$.
- The points of the topological space are defined as the frame maps from F to $\mathbf{2}_{Fr}$ which is the two-element partial order $0 < 1$, i.e.,

$$Pt(F) = Hom_{\mathbb{Frame}}(F, \mathbf{2}_{Fr}) = Hom_{\mathbb{Locale}}(\mathbf{2}_{Fr}, F).$$

- The frame of $*$ is the partial order $\emptyset < \{*\}$, i.e., $\mathbf{2}_{Fr}$.

Proof.

- There is an adjunction:

$$Hom_{\mathbb{Top}}(T, Pt(F)) \cong$$

$$Hom_{\mathbb{Locale}}(\mathcal{O}(T), F) \cong Hom_{\mathbb{Frame}}(F, \mathcal{O}(T)).$$

- The unit of the adjunction for a topological space T is

$$T \longrightarrow Hom_{\mathbb{Frame}}(\mathcal{O}(T), \mathbf{2}_{Fr})$$

which is defined for $x \in T$ as

$$x \quad \mapsto \quad f_x : \mathcal{O}(T) \longrightarrow \mathbf{2}_{Fr}.$$

- For an open set $O \in \mathcal{O}(T)$, $f_x(O) = 1$ if and only if $x \in O$. In other words, f_x picks out those open sets that contain x.

□

Proof.

- The counit of the adjunction for a frame F is

$$F \longrightarrow \mathcal{O}(Hom_{\mathbb{F}\mathrm{rame}}(F, \mathbf{2}_{Fr}))$$

which is defined for $a \in F$ as

$$a \quad \mapsto \quad \{f\colon F \longrightarrow \mathbf{2}_{Fr} : f(a) = 1\}.$$

- This means that a goes to all the frame maps that pick out a.

$\square$

Pointless Topology

The subcategories where the unit and counit of this adjunction are isomorphisms form an equivalence of categories. First some definitions.

Definition

- *A* **sober space** *is a topological space T such that every nonempty closed subset of T that cannot be separated is the closure of exactly one point.*
- *A space that is not sober will have some closed set that cannot be separated with two or more points.*
- *Perhaps we should call a space that is not sober a "Tequila space."*
- *Such a space has enough closed spaces.*
- *The category of all sober spaces and continuous maps between them is denoted as $\mathbb{Sober}$. A space is sober exactly when the above unit is an isomorphism (homeomorphism) of topological spaces.*

Pointless Topology

Definition

- *A frame F is a* **spatial frame** *if for any a and b in F with $a \not\leq b$, there is a $f\colon F \longrightarrow \mathbf{2}_{Fr}$ such that $f(a) = 1$ and $f(b) = 0$.*
- *In other words, there is a point of F that separates a and b.*
- *A frame is spatial exactly when the counit of the above adjunction is an isomorphism of frames.*
- *One can think of such frames as having enough objects.*
- *The collection of all spatial frames and frame morphisms is denoted $\mathbb{SFrame}$.*

Pointless Topology

Thus we have proven the following.

Theorem

The above adjunction induces an equivalence of categories between the category of spatial frames and the category of sober spaces. In summary:

$$
\begin{array}{ccc}
\mathbb{Frame}^{op} & \underset{O}{\overset{Pt}{\rightleftarrows}} \;\top & \mathbb{Top} \\
\uparrow & & \uparrow \\
\mathbb{SFrame}^{op} & \underset{O}{\overset{Pt}{\rightleftarrows}} \;\simeq & \mathbb{Sober}.
\end{array}
$$