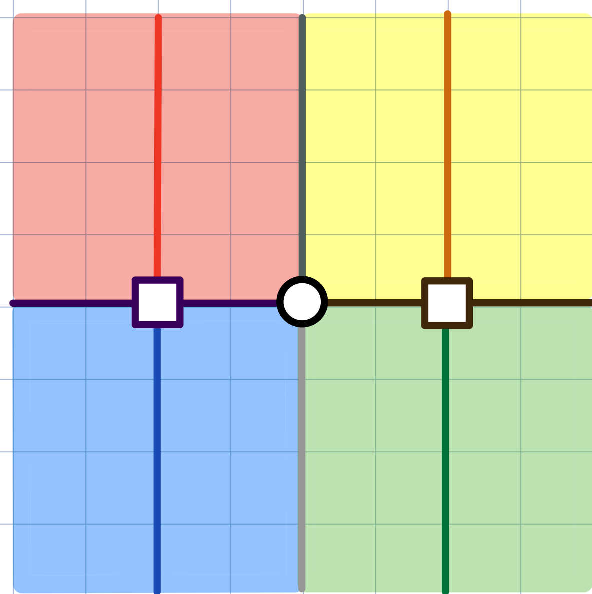
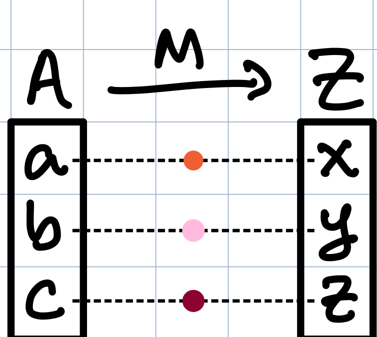
logic in color

monads & modules



We saw a relation as a 0/1-matrix, and realized: we can have many kinds of data — judgements generalize relations.



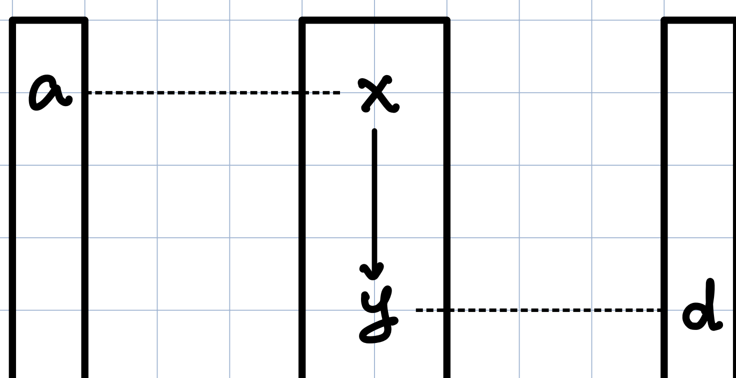
	x	y	z
a	●	○	○
b	○	●	○
c	○	○	●

Only one problem: sets don't "understand"!

In the "logic of distance", for example, types need distance too.

Otherwise, in a set, everything is infinitely far away from each other!

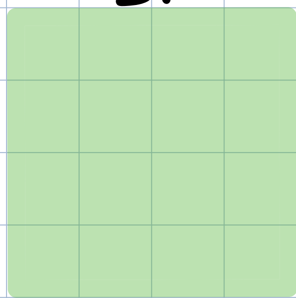
* Internalizing the notion of judgement allows logic to flow through types.

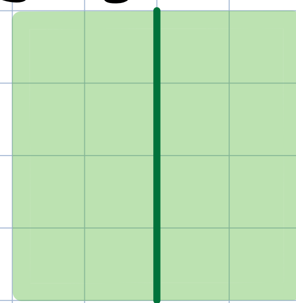


In other words, "as above, so below".

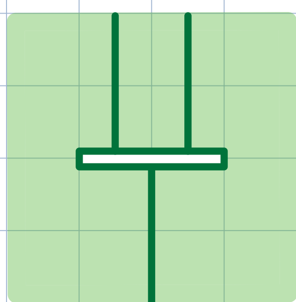
Monad

A type & judgement

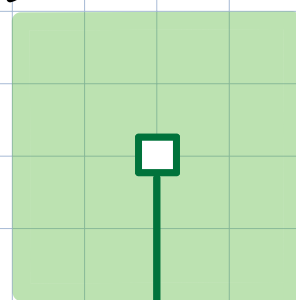




with inferences

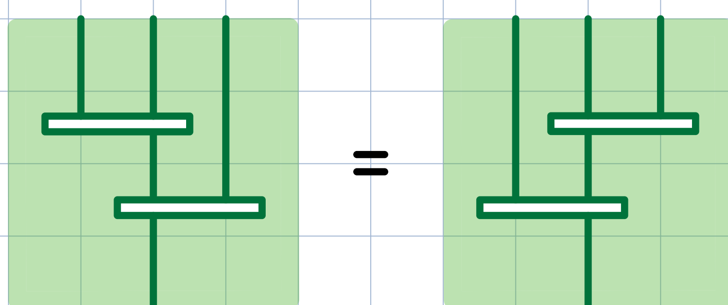


"join"

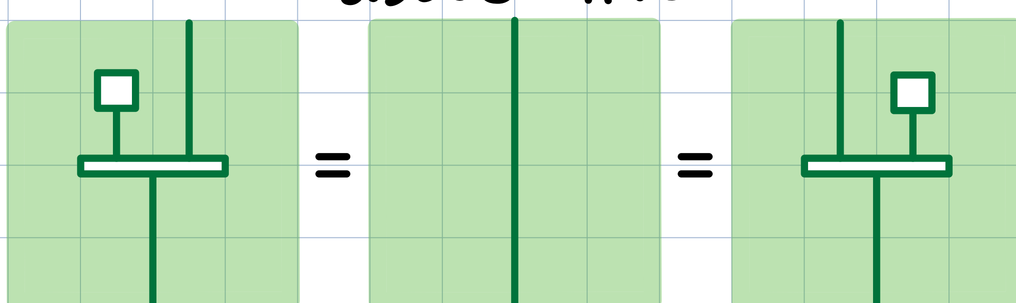


"unit"

which are



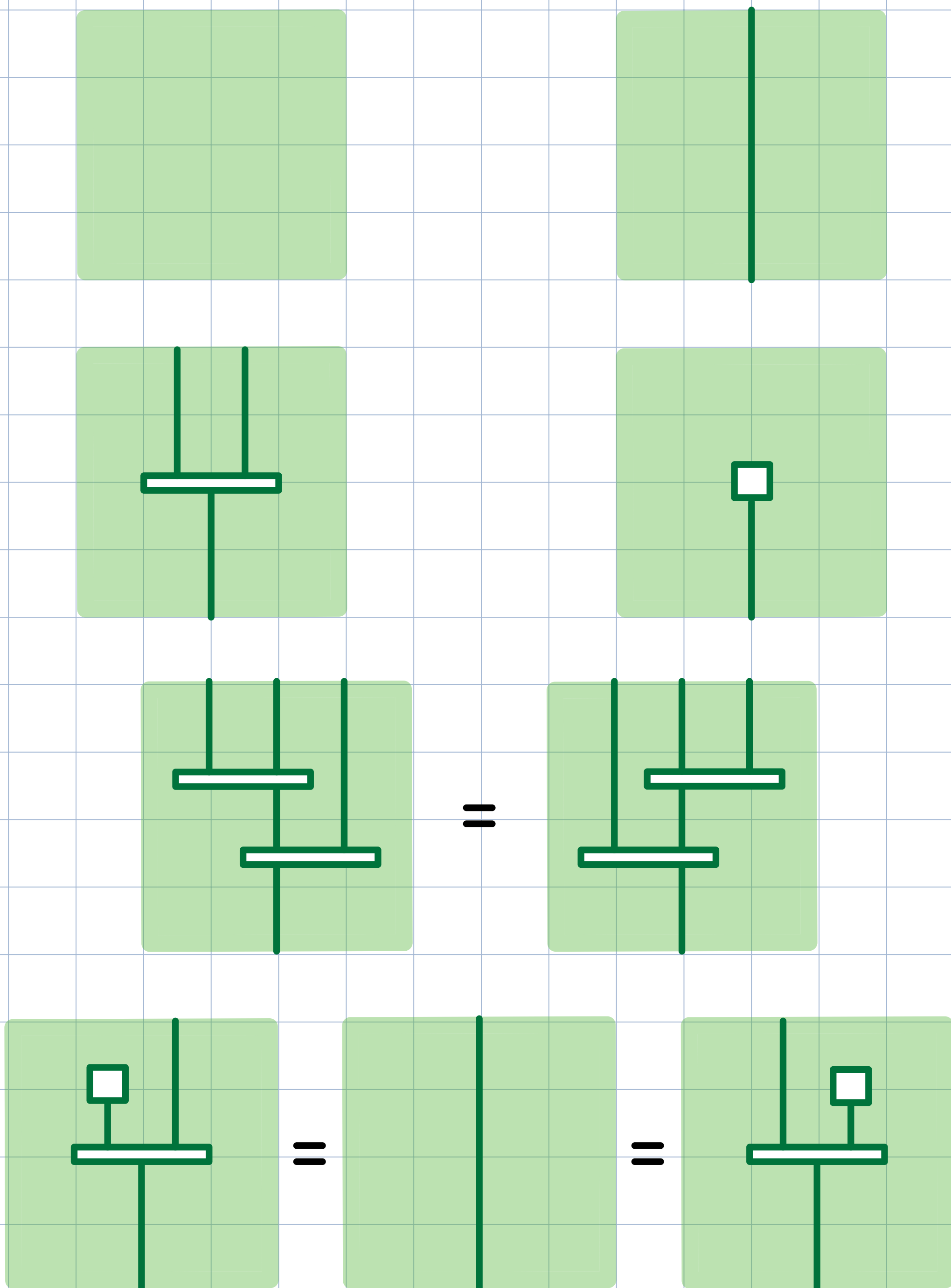
associative



& left & right unital.

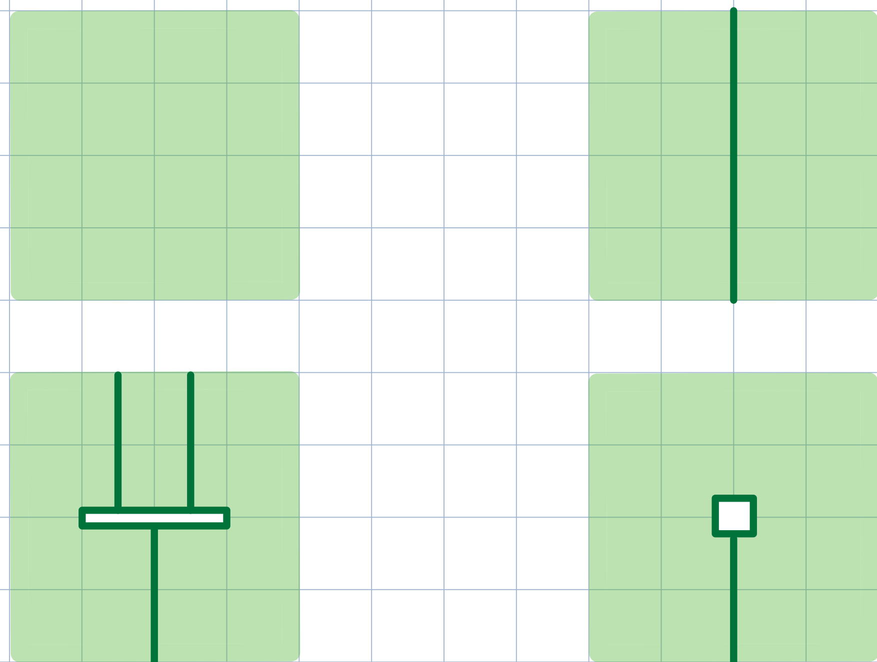
$(\mathbb{B}, \wedge, \exists)$

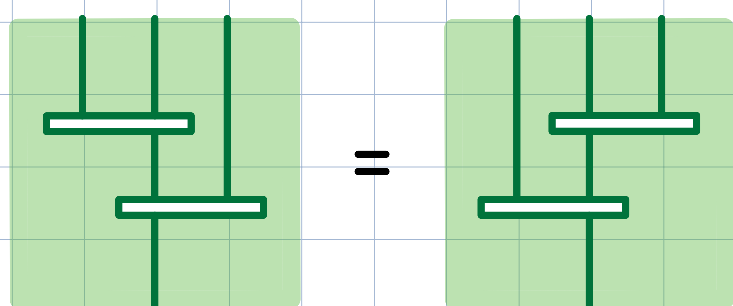
ex: matrices of $(Set, \times, \Sigma)$

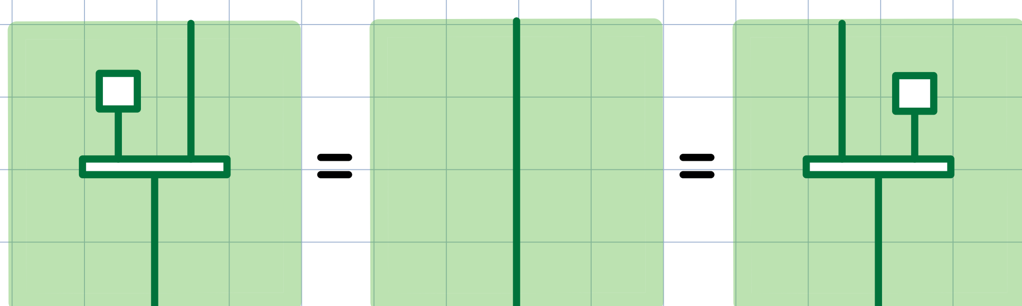


[note: identity is a monad.]

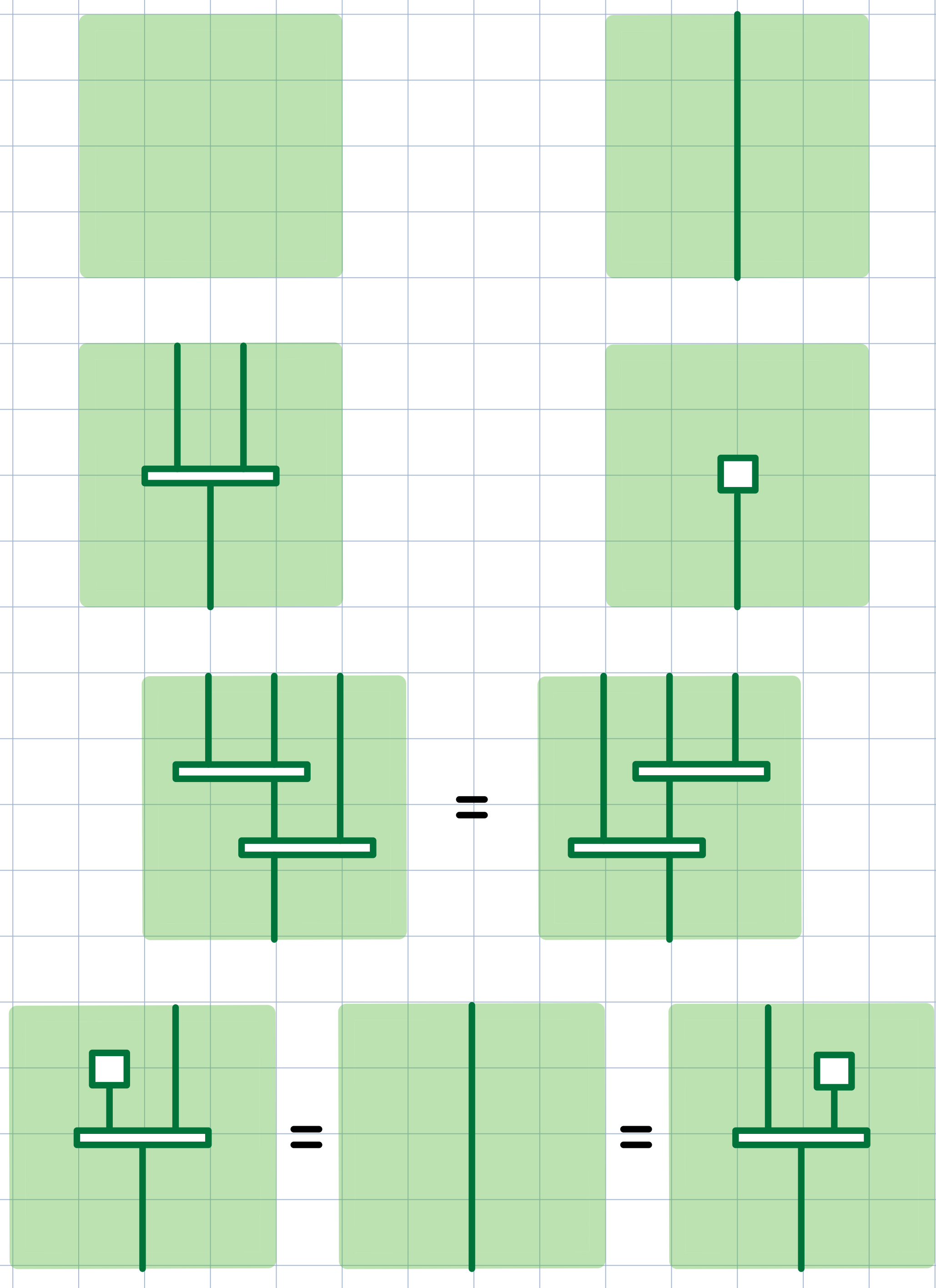
ex: matrices of $((\mathbb{R}, \geq), +, \inf)$







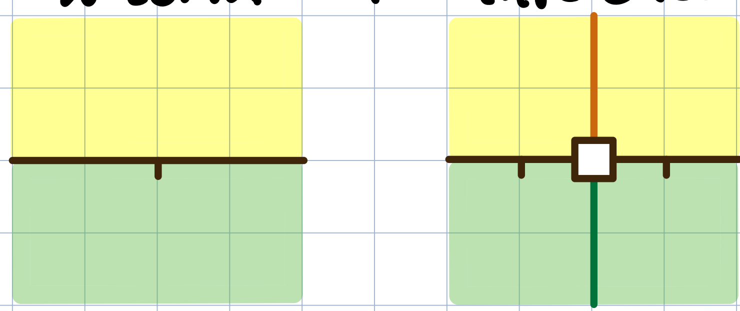
ex: matrices of $(Ab, \otimes, \Sigma)$



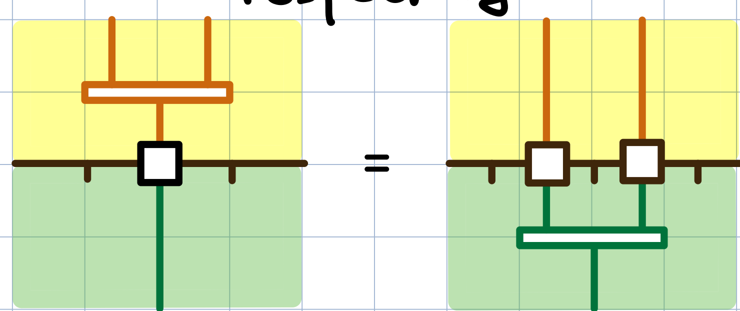
If monads are types, what are terms?

Morphism:

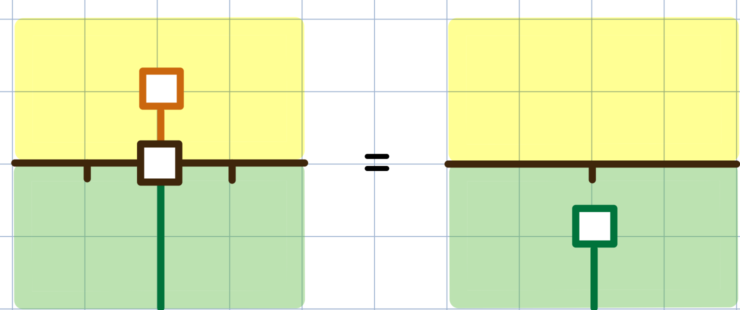
a term + inference



respecting



join



+ unit.

exs: $\mathbb{B}$

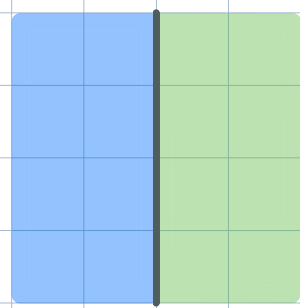
Set

$\mathbb{R}$

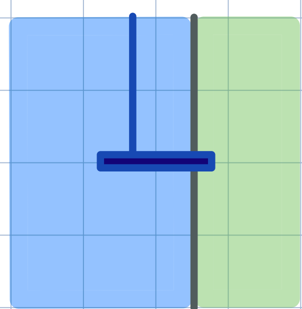
Ab

What is a judgement between monads?

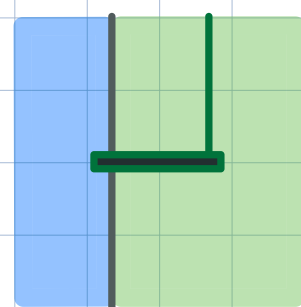
Module:
a judgement



with

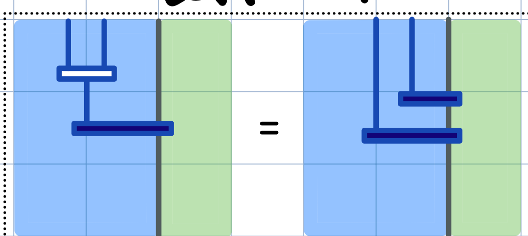


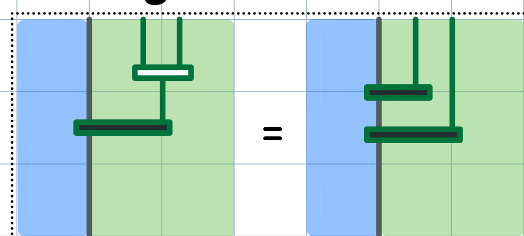
inferences



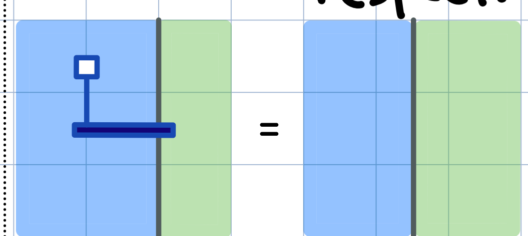
left action

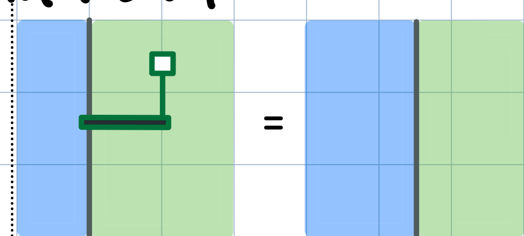
right action



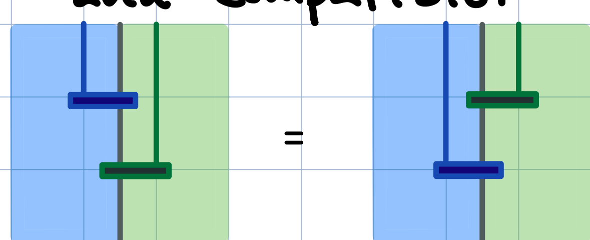


respecting join + unit

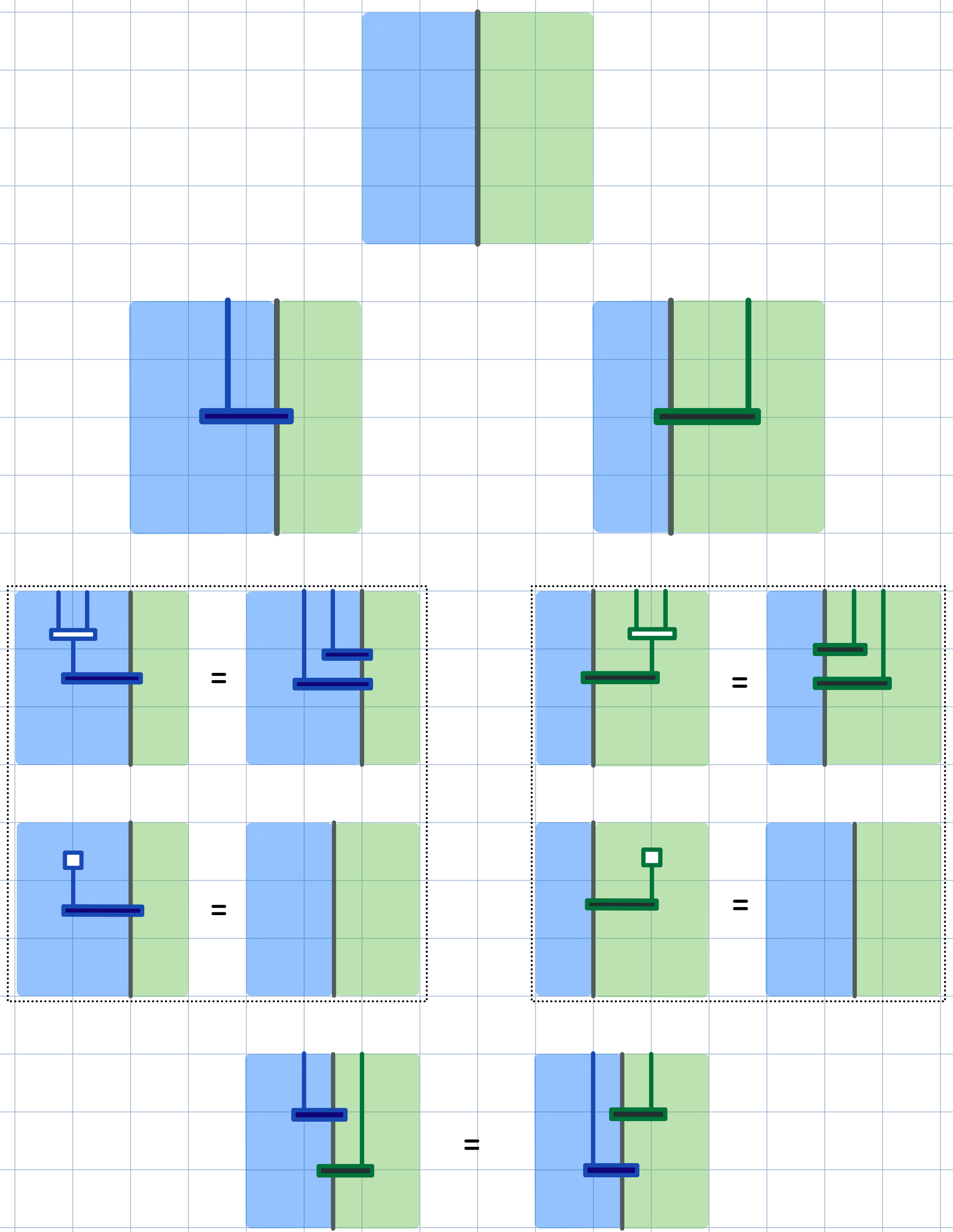




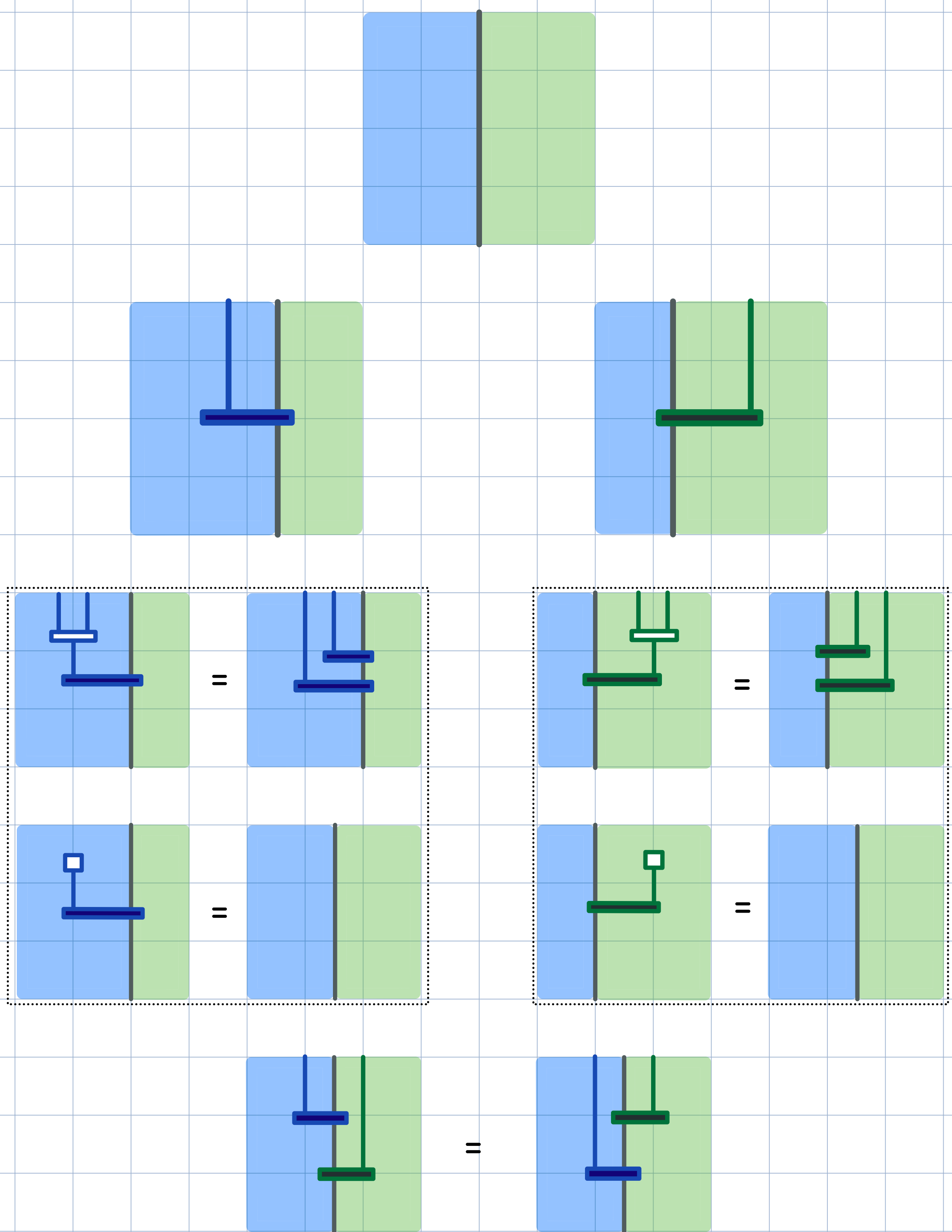
and compatible.



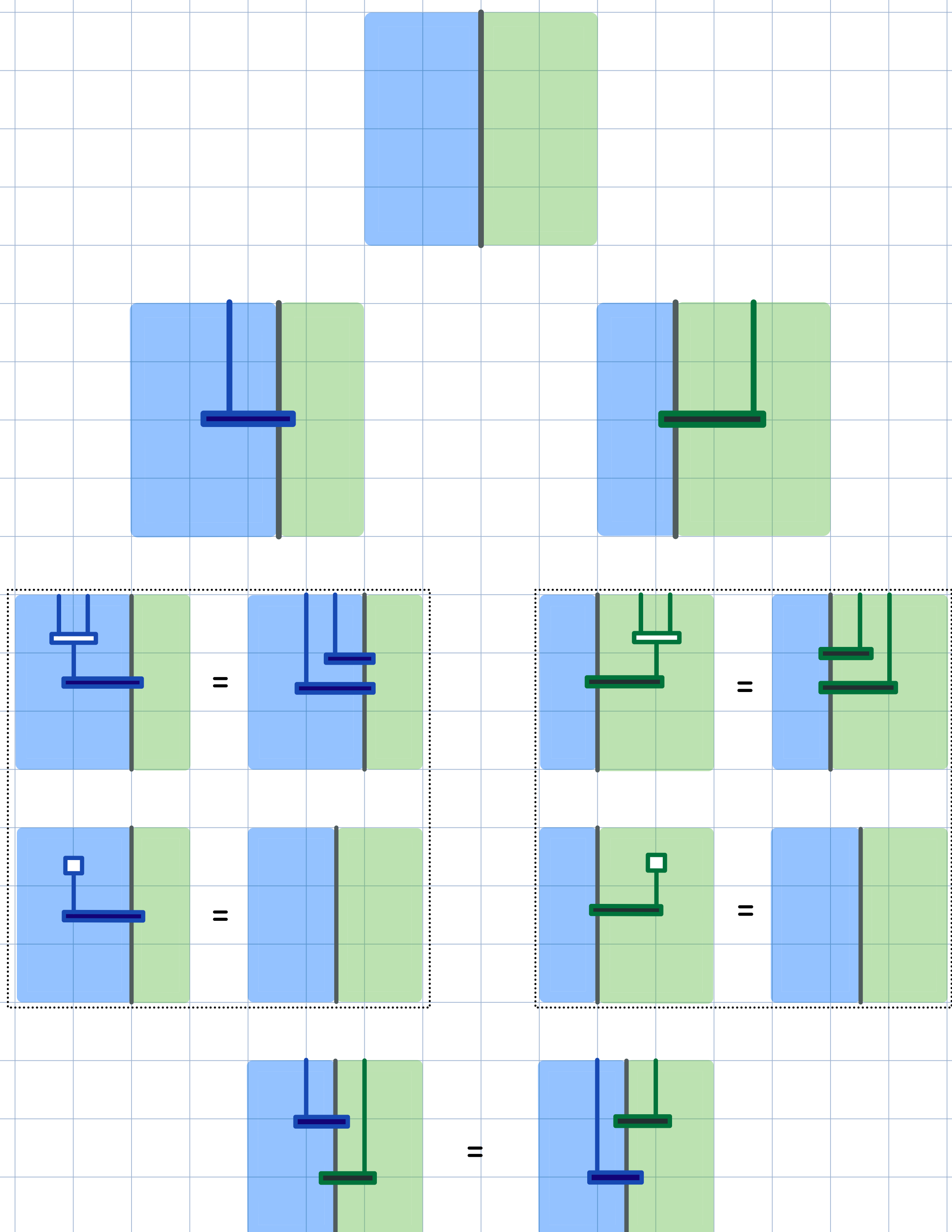
ex: $\mathbb{B}$



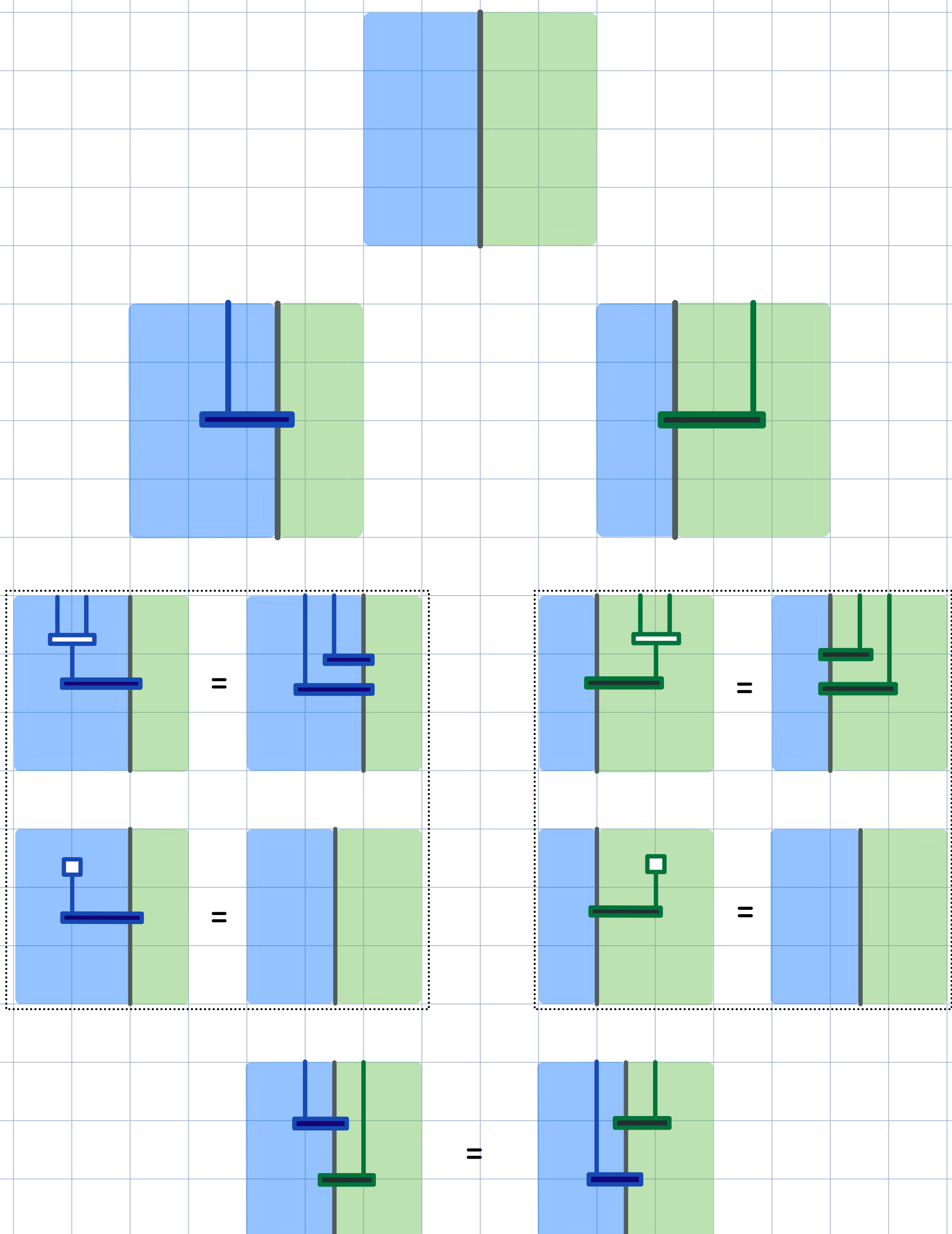
ex: Set



ex: $\mathbb{R}$

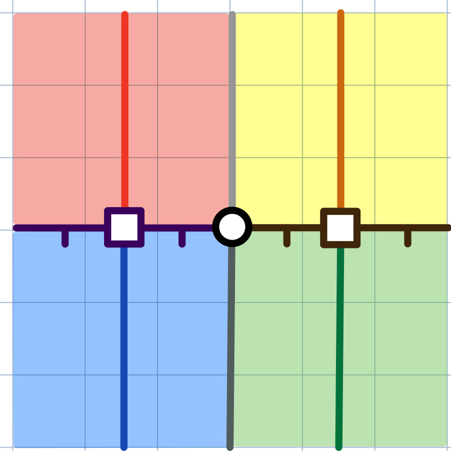


ex: Ab

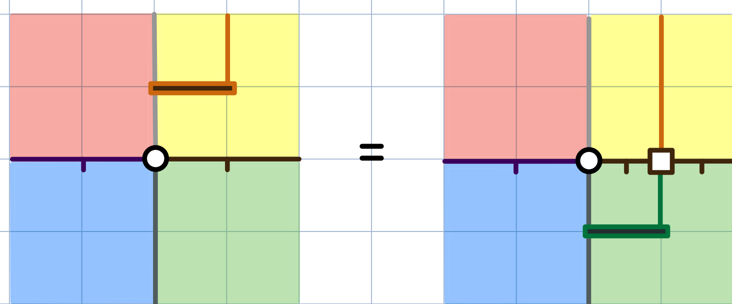


What is an <u>inference</u> between modules?

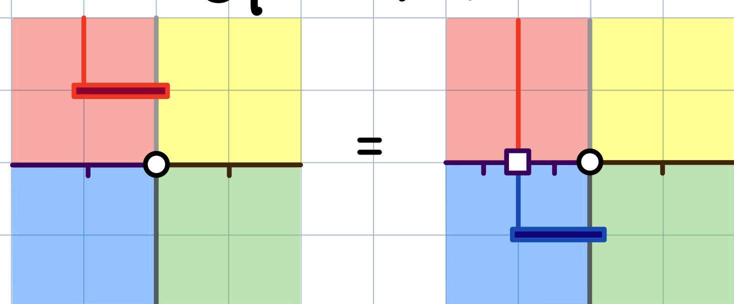
An inference



with



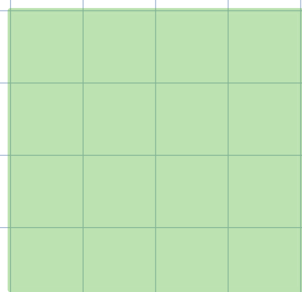
"equivariance"



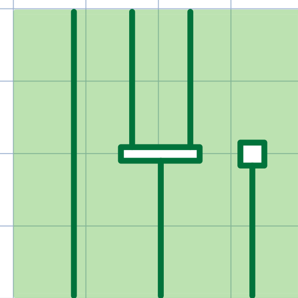
These are the squares of

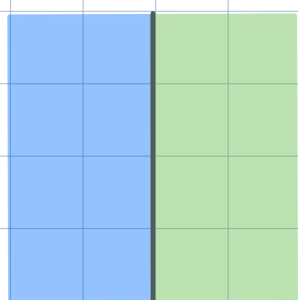
* $\mathrm{Mod}\mathbb{V}$. *

Mod IV

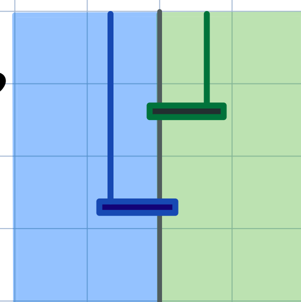


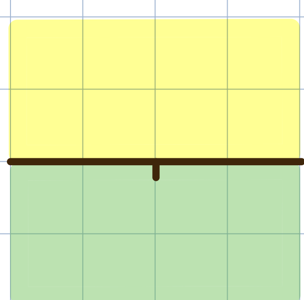
types := monads



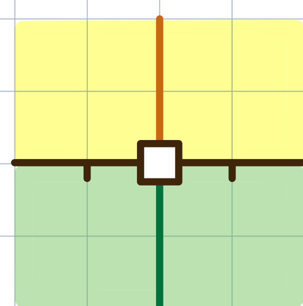


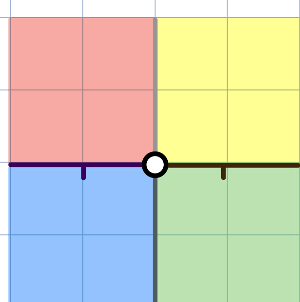
judgements := modules



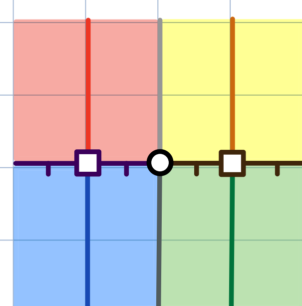


terms := morphisms



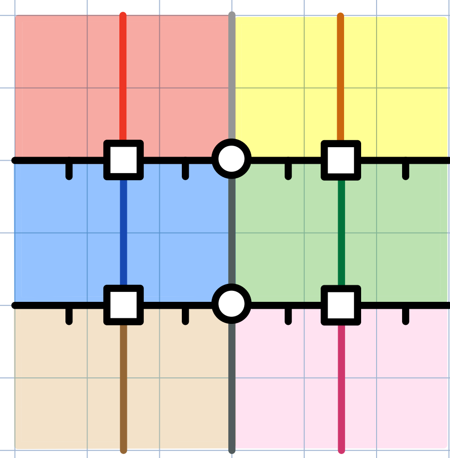


inferences := trans-formations



(note: color)

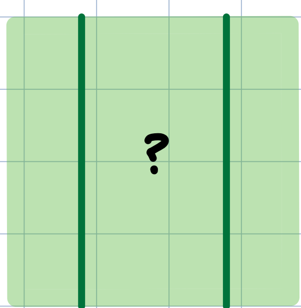
Composition:

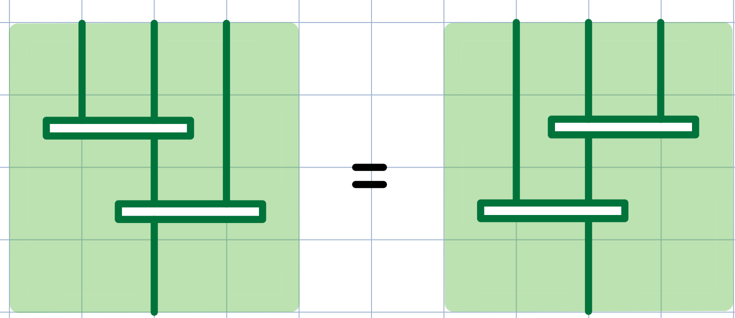


sequential is easy.

Parallel is more: now, types act.

First, what is the identity?



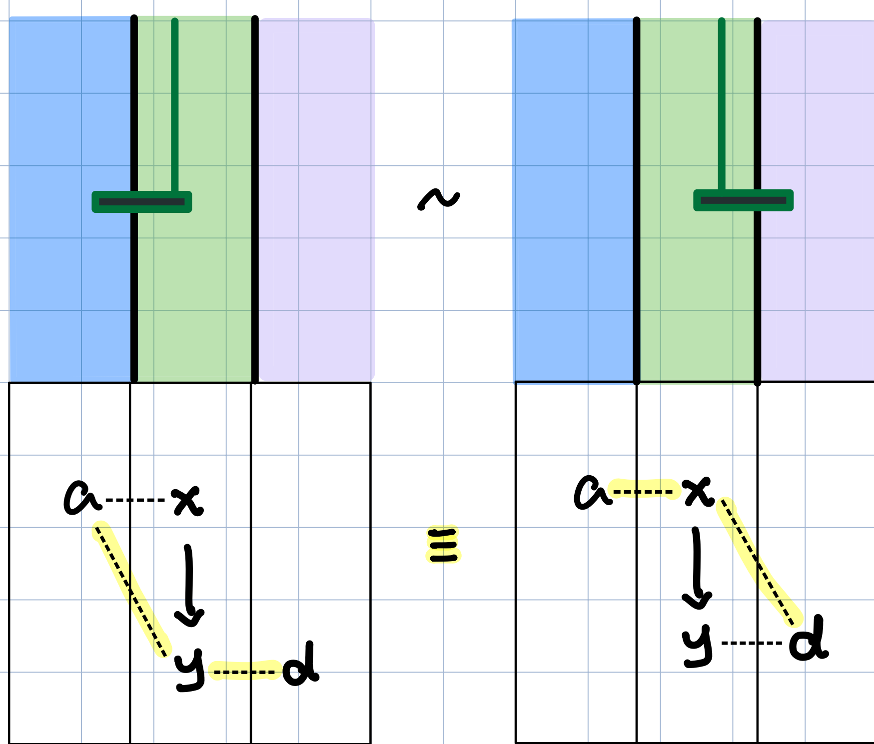


The monad itself!

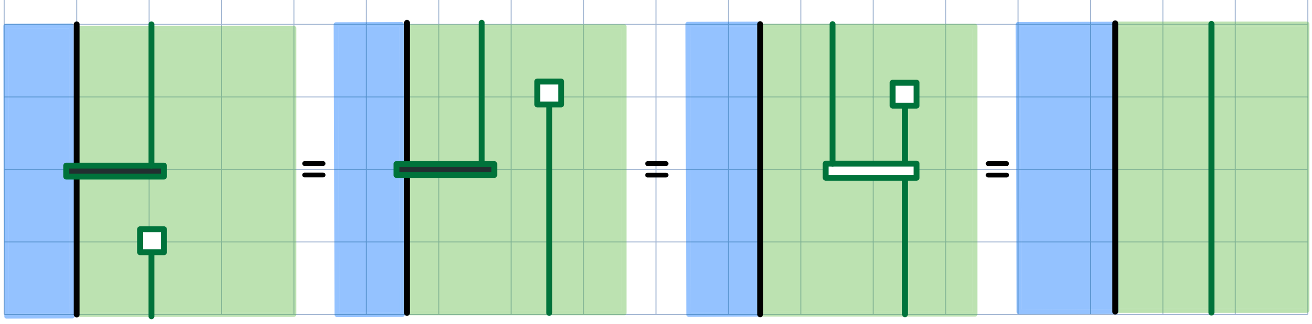
Because types act on judgements,

to compose $A \to B \to C$
we must equate the "inner actions" of B.

So, we quotient:

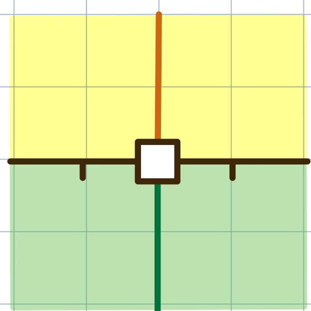
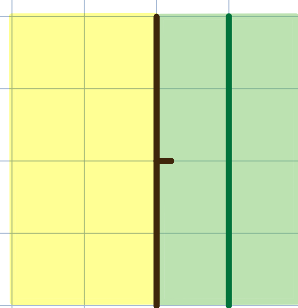


We need this for composition to be unital.

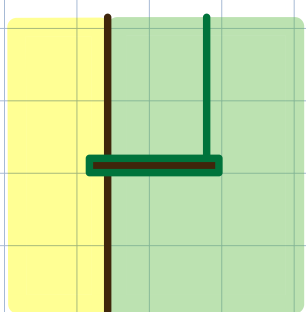
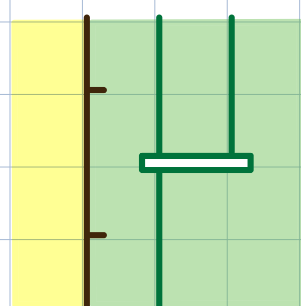


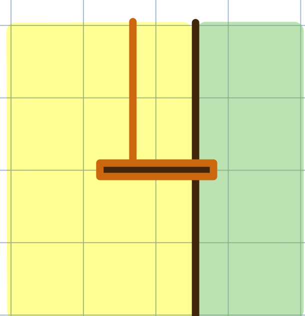
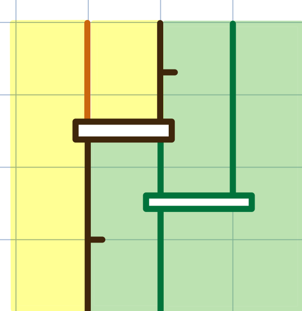
Remember: every function gives an adjoint pair of relations, graph & cograph.

Now, in ModV: term ⇝ judgement?

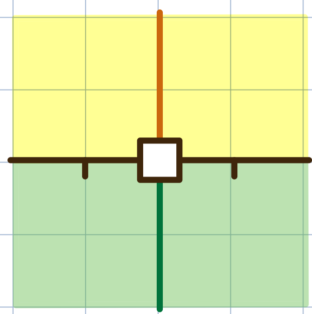
term  ⇝  positive judgement

aka "companion"

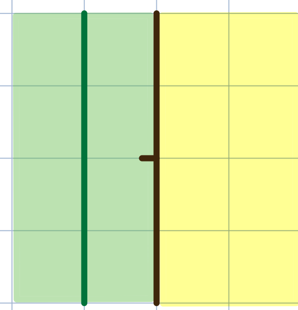
 := 

 := 

term

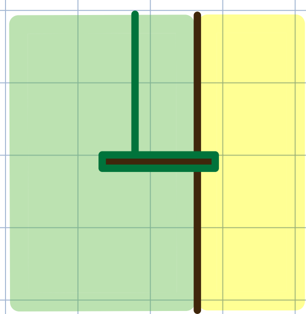


$\rightsquigarrow$

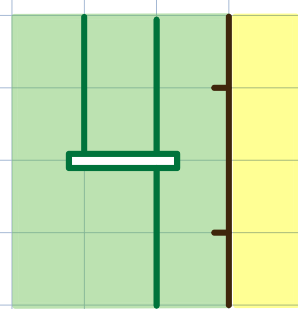


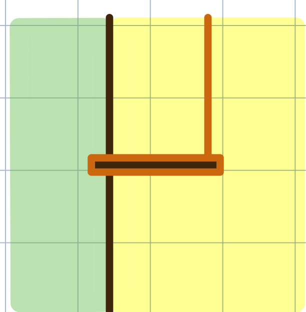
negative judgement

aka "conjoint"

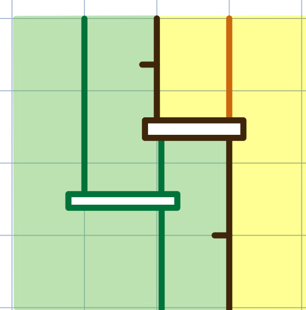


:=





:=



<u>puzzle</u>: show that
positive is left adjoint to negative.

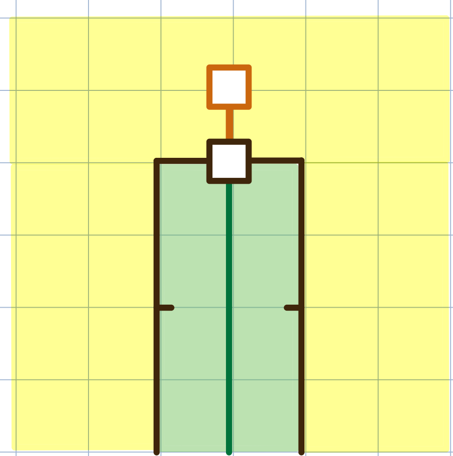
Mod expands logic:

formal $\subset$ actual

Questions / Thoughts ?

puzzle answer

unit



counit