Type-theoretic (foundations for) **mathematics**

(title is a WIP)

Thanos Tsouanas*

Spring semester 2025-2026

* UFRN, Brazil

⚠ pink means homework!

Buzzwords

analytic ~ synthetic

proposition ~ judgment

Bool ~ Prop

axiom ~ law

specification ~ implementation

intension ~ extension

type ~ set

static ~ dynamic

command ~ proposition

syntax ~ semantics

(?) proof ~ formal proof

$\beta \sim \eta$

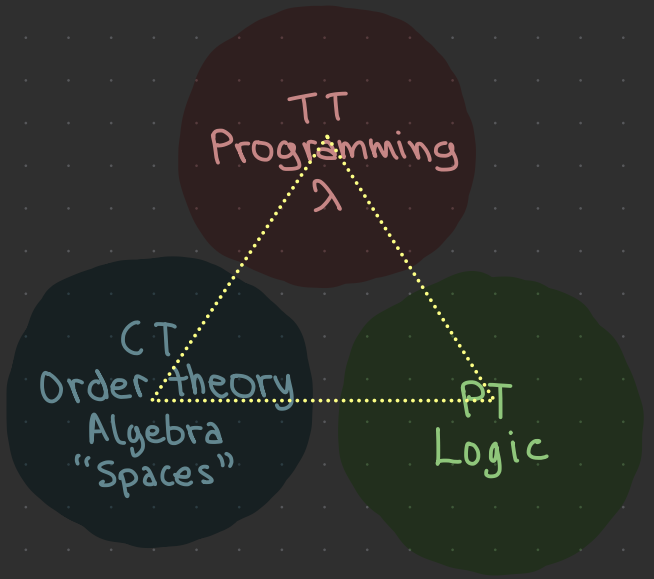
internal ~ external (internalization)

introduction ~ elimination

Introduction

lec01
2026-02-24

- Foundations
- Formalization
- Trinitarianism
- Type theory (TT)
- Category theory (CT)
- Order theory
- Proof theory (PT)



What is Set theory?

⚠ Naming inconsistency: « Set theory » vs « Group theory »
Ring
Monoid
Field
⋮

How do we read $(\forall x)(\exists y)[\ \ldots]$ in each.

we call these things «sets»

for every member of the universe x, there exists a member of the universe y, such that ...

set-

set-

for every member of the group x, there exists a member of the group y, such that ...

we don't have a name for these

⚠ $a : \alpha$ is a judgment

$a \in A$ is a proposition

Intension vs extension

$u \equiv v$	means:	u means the same thing as v
$P \Leftrightarrow Q$	means:	P, Q mean the same thing

(intensional)

same sense (Frege)

same reference

$u = v$	means:	u, v denote equal objects
$P \Leftrightarrow Q$	means:	P, Q are logically equivalent

(extensional)

aka: P iff Q;

P is sufficient and necessary for Q; ...

- $\{0,1\} = \{1,0\}$? $\{0,1\} \equiv \{1,0\}$?
- Does $a \equiv b$ imply $a = b$?
- Why did I spell out «imply» instead of «$\Rightarrow$»?

def

LHS $\overset{\text{def}}{\equiv}$ RHS ← We are defining LHS to mean RHS

If they are Props, use ($\overset{\text{def}}{\iff}$)

Examples:

$a \mid b \overset{\text{def}}{\iff} (\exists k)[ak = b]$: ?

vs

$a \mid b \iff (\exists k)[ak = b]$: ?

vs

$a \mid b \iff (\exists k)[ak = b]$: ?

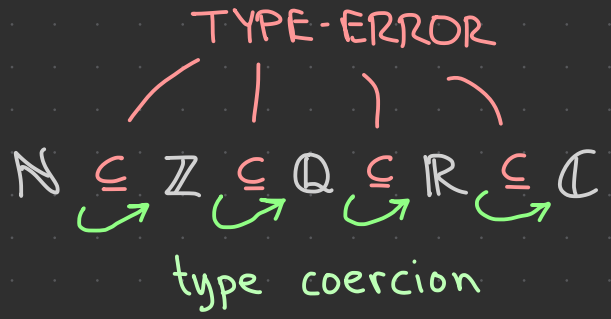
Some invisible mathematics

$\_\_ :\alpha \;=_\alpha\; \_\_ :\alpha$

$\_\_ :\alpha \;\in_\alpha\; \_\_ : \mathrm{Set}\ \alpha$ — "function application"

$\mathrm{Set}(\alpha)$

{ Θάνος, 42 } ← TYPE-ERROR (ill-typed)



$\mathbb{Z}_{\mathbb{N}}$ (the natural integers

$\mathbb{R}_{\mathbb{Q}}$ (the rational reals

. . .

$\mathbb{R}_{\mathbb{N}} \subseteq_{\mathrm{Real}} \mathbb{R}_{\mathbb{Z}} \subseteq_{\mathrm{Real}} \mathbb{R}_{\mathbb{Q}} \subseteq_{\mathrm{Real}} \mathbb{R}$

Implementation-agnostic

$\cos \in \sqrt{2}$?

$\langle 2,1 \rangle_{\text{Kuratowski}} \equiv \{\{2,1\},\{2\}\}$

$2 \in \langle 0,1 \rangle$?

$2 \in \langle 2,1 \rangle$?

$2 \in \bigcup \langle 1,2 \rangle$?

Claim: you may say that you use set theory but you think in type-theoretic terms!

Some typing statements

Sotiris : Person

mother of Θάνος : Person

mother of ____ : Person → Person

↑ hole (to be filled by something of the appropriate type)

⚠ these are four different 42s!

42_{Int} : Int

42_{Nat} : Nat

42_{Real} : Real

42_{Rat} : Rat

Manolis was born in Greece : Prop

Manolis was born in ____ : Country → Prop

____ was born in Greece : Person → Prop

____ was born in ____ : (Person × Country) → Prop

: Person → (Country → Prop)

: Country → (Person → Prop)

Currying

whatever this means

$$(+) : (\alpha \times \beta) \to \gamma \quad \cong \quad \alpha \to (\beta \to \gamma)$$

claim: you have known this since ~~elementary~~ high school!

Some more typing

lec02
2026-02-27

$3 + \underline{6} : \mathrm{Nat} \to \mathrm{Nat}$

$\underline{3} + \underline{6} : (\mathrm{Nat} \times \mathrm{Nat}) \to \mathrm{Nat}$ ← curried

$: \mathrm{Nat} \to (\mathrm{Nat} \to \mathrm{Nat})$ ← uncurried

$3 \underline{\leq} 6 : ((\mathrm{Nat} \times \mathrm{Nat}) \to \mathrm{Nat}) \to \mathrm{Nat}$

$\underline{\leq}$ ← divides

$\mathrm{BinOp}\ \mathrm{Nat}$

Operator, Predicate

$\mathrm{BinOp}\ \alpha \overset{\text{sug}}{\equiv} (\alpha \times \alpha) \to \alpha$

$\mathrm{Pred}\ \alpha \overset{\text{sug}}{\equiv} \alpha \to \mathrm{Prop}$

Props: using & proving

		How to use	How to attack (prove)
Conj	$P \wedge Q$ (wedge; $\&$)	Ext-L ___. Ext-R ___.	Split.
disj	$P \vee Q$ (vee)		
Imp	$P \Rightarrow Q$ ($\supset$ – horseshoe)		Suppose P.
	$\bot$		
forall	$(\forall x : \alpha)[\varphi(x)]$		
exists	$(\exists x : \alpha)[\varphi(x)]$		
eq	$u = v$		

Comparing three theorems

$\vdash (\forall a,b,c)[\cdots]$

0. For all a, b, c : Int, if a | b & b | c then a | c

   Prove

   $a : Int, \ldots, c : Int \vdash a|b \,\&\, b|c \Rightarrow a|c$

0. Let a, b, c : Int. : Cmd

   (actually 3 lines)

   Prove If a | b & b | c then a | c. : ~~Prop~~ Cmd

   $a : Int, \ldots, a|b \,\&\, b|c \vdash a|c$

0. Let a, b, c : Int. : Cmd

   Suppose a | b and b | c. : Cmd

   (actually 2 lines)

   Prove a | c. : Cmd

0. For all $a, b, c : \text{Int}$, if $a \mid b \;\&\; b \mid c$ then $a \mid c$

Proof.

Let $u, v, w : \text{Int}$

Suppose $u \mid v \;\&\; v \mid w$ (1).

Ext-L from (1).

Ext-R from (1).

⋮

∎ Halmos tombstone

(You have killed the monster)

Givens

$u : \text{Int}$

$v : \text{Int}$

$w : \text{Int}$

$u \mid v \;\&\; v \mid w$

$u \mid v \iff (\exists k)[u \cdot k = v]$

$v \mid w \iff (\exists k)[v \cdot k = w]$

Goal ← type inference $(\forall a : \text{Int})$.

~~$(\forall a)(\forall b)(\forall c)[(a \mid b \;\&\; b \mid c) \Rightarrow a \mid c]$~~

~~$(u \mid v \;\&\; v \mid w) \Rightarrow u \mid w$~~

$u \mid w \iff (\exists k)[u \cdot k = w]$

lec03

2026-03-03

		How to use	How to attack
conj	$P \wedge Q$ (wedge; &)	Ext-L $P \wedge Q$. Ext-R $P \wedge Q$.	Split.
disj	$P \vee Q$ (vee)	Separate in cases by $P \vee Q$.	Left. Right.
imp	$P \Rightarrow Q$ ($\supset$ – horseshoe)	Apply $P \Rightarrow Q$ to P to obtain Q.	Suppose P.
falsum	aka: falsehood, absurdity, bottom, boom, ... $\bot$	Boom! ↳or: Contradiction.	(none)
forall	$(\forall x : \alpha)[\varphi(x)]$	Apply $(\forall x \cdot \alpha)[\varphi(x)]$ to $a : \alpha$ to obtain $\varphi(a)$.	Let a : α.
exists	$(\exists x : \alpha)[\varphi(x)]$	Let $a : \alpha$ s.t. $\varphi(a)$.	Use $\_\_ : \alpha$
eq	$u = v$	(Compute?)	?

Other logical notions

$P \Leftarrow Q \quad \overset{\text{def or sug}}{\Longleftrightarrow} \quad Q \Rightarrow P$

$P \Leftrightarrow Q \quad \overset{\text{sug}}{\Longleftrightarrow} \quad (P \Rightarrow Q) \;\&\; (P \Leftarrow Q)$

$\neg P \quad \overset{\text{sug}}{\Longleftrightarrow} \quad P \Rightarrow \bot$

Previously...

Let $u, v, w : \mathrm{Int}$

Suppose $u \mid v \;\&\; v \mid w$ (1).

Ext-L from (1). (2)

Ext-R from (1). (3)

Givens

$u : \mathrm{Int}$

$v : \mathrm{Int}$

$w : \mathrm{Int}$

$u \mid v \;\&\; v \mid w$

$u \mid v \;\left(\Leftrightarrow\; (\exists k)[\, u \cdot k = v \,]\right)$

$v \mid w \;\left(\Leftrightarrow\; (\exists k)[\, v \cdot k = w \,]\right)$

Goal

$u \mid w \;\left(\Leftrightarrow\; (\exists k)[\, u \cdot k = w \,]\right)$

An illusion of using ∃ while doing nothing

failed attempt to use ∃

Since (2) thence there exists k : Int s.t. u · k = v (4) : Cmd

IF u | v then there exists k : Int s.t. u · k = v : Prop

Since (4) thence there exists k' : Int s.t. u · k' = v. (5)

pointless & bad

technically wrong word

Check: Scope, free & bound variables.

these did not really do anything:

« since u | v thence u | v. »

..just written with different words

Actually using ∃

$(\exists k : \mathrm{Int})[u\,k = v]$

(Since $u \mid v$ thence) let $a : \mathrm{Int}$ s.t. $u\,a = v$ (ha)

(Since $v \mid w$ thence) let $b : \mathrm{Int}$ s.t. $v\,b = w$ (hb)

«Compute.»

Compute:

$w = \overline{v}\,b$ [(hb)]

$= (u\,a)\,b$ [(ha)]

$= u\,(a\,b)$ [(.)-ass ...]

.., $w = u\,(a\,b)$ ⊢ ...

its effect on the state

Use $a\,b$

Immediate.

□

is there anything here, really?

(Isn't this "just" (.)-ass and nothing more?)

Proof-relevance

Suppose $u \mid v$. (huv)

Let huv : $u \mid v$

Let huv be a proof of $h \mid v$.
(or: an evidence for)

Givens

u · Int
v · Int
w Int
h · $u \mid v$ & $v \mid w$
hl · $u \mid v$ ($\Leftrightarrow$ $(\exists k)[u \cdot k = v]$)
hr : $v \mid w$ ($\Leftrightarrow$ $(\exists k)[v \cdot k = w]$)
ha : $u\,a = v$
hb : $v\,b = w$

No prejudice now!

Same rights! 

Proving "laws of logic"

Θ. ⊢ P ⇒ P

Proof.

Suppose P. -- P ⊢ P

Exactly P.

□

Θ. ⊢ (P ∨ Q) ⇒ (Q ∨ P)

Proof.

Suppose P ∨ Q -- P ∨ Q ⊢ Q ∨ P

~~Left.~~ ~~-- P ∨ Q ⊢ Q~~

Valid move, but bad move. 'Undo'

By cases on P ∨ Q -- two things to prove now!

CASE-L : -- P ⊢ Q ∨ P

    Right. -- P ⊢ P
    Exactly P.

CASE-R : -- Q ⊢ Q ∨ P

    Left. -- Q ⊢ Q
    Exactly Q.

«Math as if humans matter» interpretation

or: mathematics is a social activity.

A true means «I have a proof of A»

LEM:

For each proposition P,

$P \vee \neg P$ true

(Now pick your favorite open problem)

Double negation

lec04
2026-03-06

$\vdash P \Rightarrow \neg\neg P$

$\neg P \Rightarrow \bot$

Sup P. $\quad P \vdash \neg\neg P$

Sup $\neg P$. $\quad P, \neg P \vdash \bot$

Apply $\neg P$ on P $\quad P, \neg P, \bot \vdash \bot$

Boom.*

Contradiction.

means that somewhere in my givens I have both A and $\neg A$ so just do the obvious.

□

* A different way to finish this proof:

-- $P, \neg P, \bot \vdash \bot$

Exactly $\bot$.

$\vdash \neg\neg P \Rightarrow P$

Sup $\neg\neg P$ $\quad \neg\neg P \vdash P$

Claim: $\neg P$ $\quad \neg\neg P \vdash \neg P$ $\Leftrightarrow P \Rightarrow \bot$

| Sup P. $\quad \neg\neg P, P \vdash \bot$

| Claim: $\neg P$ $\quad \neg\neg P, P \vdash \bot$

| | Sup P. $\quad \neg\neg P, P, P \vdash \bot$

| | ⋱

× ×

Contra. × $\quad \neg\neg P, \neg P \vdash P$

□ ×

LEM

$\vdash P \vee \neg P$

Right $\vdash \neg P$

Sup P. $P \vdash \bot$

(dead end)

maybe we should have gone left:

$\vdash P \vee \neg P$

Left. $\vdash P$

(even deader end)

The irrefutability of LEM

bad name (dishonest)

honest: every instance of LEM is irrefutable

$\vdash \neg\neg(P \vee \neg P)$

Sup $\neg(P \vee \neg P)$. $\quad \neg(P \vee \neg P) \vdash \bot$

Claim $P \vee \neg P$. $\quad \neg(P \vee \neg P) \vdash P \vee \neg P$

| Right $\quad \neg(P \vee \neg P) \vdash \neg P$

| Sup P. $\quad \neg(P \vee \neg P), P \vdash \bot$

| Hence $P \vee \neg P$. $\quad \neg(P \vee \neg P), P, P \vee \neg P \vdash \bot$

| Contradiction.

-- $\neg(P \vee \neg P), P \vee \neg P \vdash \bot$

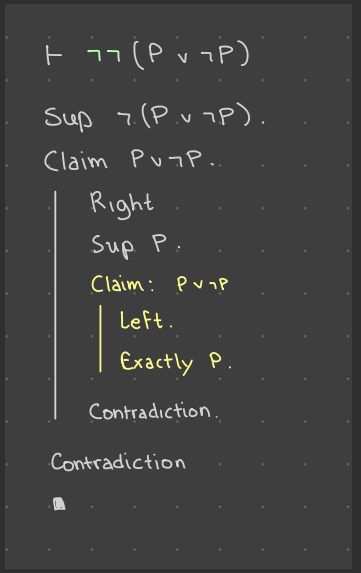
Contradiction

3 lines

Claim: $P \vee \neg P$

| Left.

| Exactly P.



Stable and decidable

A stable $\overset{\text{def}}{\Leftrightarrow}$ $\vdash \neg\neg A \Leftrightarrow A$

A decidable $\overset{\text{def}}{\Leftrightarrow}$ $\vdash A \vee \neg A$

Saving proofs

Somewhere, in some proof:

-- p : Int, p prime ⊢ G

Since $p \equiv 1 \pmod 4$, let k s.t. $p = 4k + 1$. ✗

⋮

∎ ✗

Similarly in some proof...

-- . . ⊢ G

Separate in cases. ← This is a use of disjunction! (you are using $A \vee \neg A$)

Do you have this disjunction?

If you do, cool. (You are using something you have.)

If you don't, not cool (same problem as above)

CASE A:

| ⋮

CASE ¬A:

| ⋮

∎

We can save the proofs by being honest in the statements:

just add the missing given(s) as hypotheses . . .

```
-- p : Int,  p prime,  p ≡ 1 (mod 4)  ⊢ G
⋮
Since  p ≡ 1 (mod 4), let k s.t. p = 4k + 1.  ✓
⋮
∎ ✓
```

```
..., A ∨ ¬A, ... ⊢ G
Separate in cases.  ✓
CASE A:
| ⋮
CASE ¬A:
| ⋮
∎ ✓
```

(...or prove them from your actual givens)

Rules of inference notation

I can reduce

the problem

of constructing C

(proving)

to the problems

of constructing $J_1, \ldots, J_n$

(proving)

from these premisses (ingrediends)

I infer this conclusion

(build, create, construct) (object)

construction of C

or: derivation tree

or: ...

$(=_\alpha)$

$(=)$ is an equivalence relation

- reflexive $\dfrac{}{a =_\alpha a}$ refl
- transitive $\dfrac{a = b \qquad b = c}{a = c}$ trans
- symmetric $\dfrac{a = b}{b = a}$ sym
- substitution $\dfrac{a = b}{\qquad\qquad}$ sub [which occurrences]

Multiply both sides by ... on the left

Compute:

$$
\begin{aligned}
a &= 1a && [\cdots] \\
&= (c^{-1}c)a && [\cdots] \\
&= c^{-1}(ca) && [\cdots] \\
&= c^{-1}(cb) && [ca = cb] \\
&= (c^{-1}c)b && [\cdots] \\
&= 1b && [\cdots] \\
&= b && [\cdots]
\end{aligned}
$$

Yikes!

Since $ca = cb$,

Hence $c^{-1}(ca) = c^{-1}(cb)$. $[(c^{-1}\cdot)]$

Hence $(c^{-1}c)a = (c^{-1}c)b$. [ass ...; ass ...]

Hence $1a = 1b$ [... ; ...]

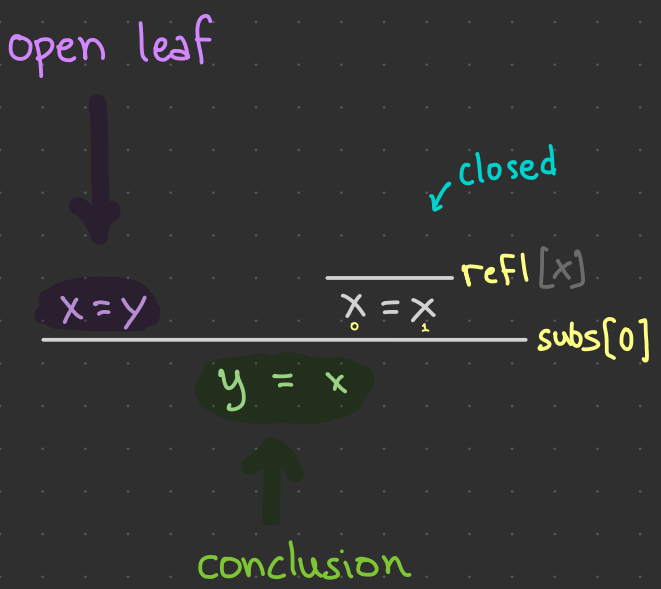
Hence $a = b$. [... ; ...]

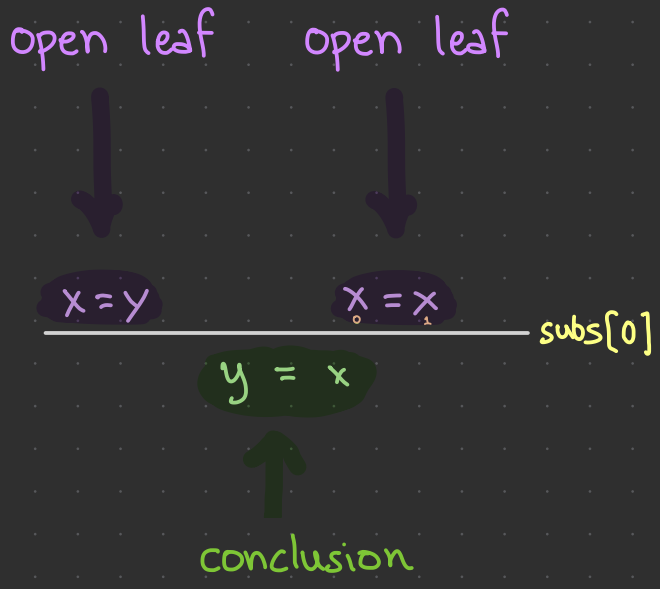
«Apply the same function to both sides.»

$$\dfrac{u =_{\alpha?} v \qquad \dfrac{}{f\,u_{\circ} =_{\beta?} f\,u_{1}}\ \text{refl}}{f\,u =_{\beta?} f\,v}\ \text{Sub}\ [1]$$

What does a tree accomplish?

lec05
2026-03-10





A construction/proof of

$x = y \vdash y = x$

A construction/proof of

$x = y,\ x = x \vdash y = x$

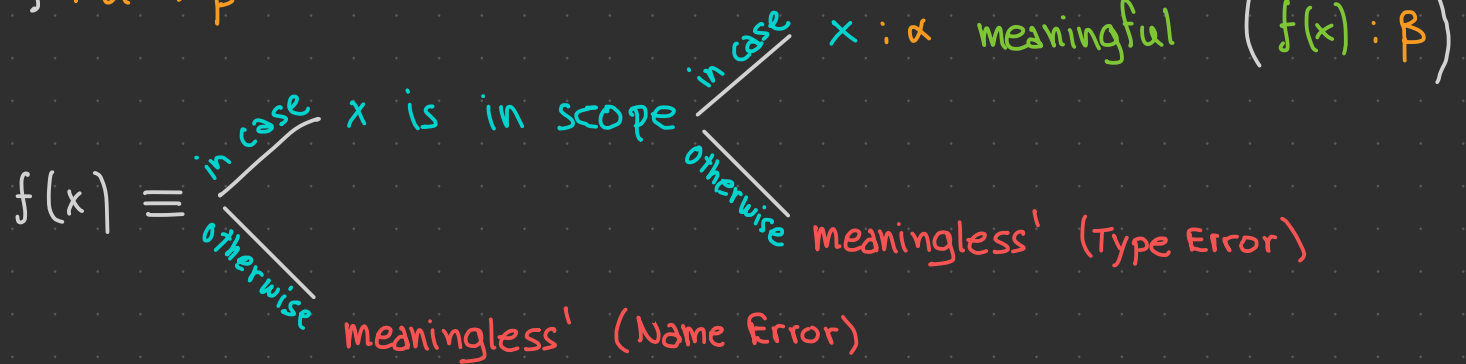
f vs $f(x)$

TYPE ERROR! $\cos(x)$ is not a function!

« The function $\cos(x)$ is periodic. » ✗

« The function $\cos$ is periodic. » ✓

$f : \alpha \to \beta$



Anonymous functions

why give it a name?

« Let $d \overset{\text{def}}{\equiv} x + y$.

The gcd of a, b is d. »

vs

« The gcd of a, b is $x + y$. »

why give it a name?

« Let $f(x) \overset{\text{def}}{\equiv} x^2 + 1$.

The function we seek is f. »

vs

« The function we seek is $(x \mapsto x^2 + 1)$. »

Defining functions

$f \overset{\text{def}}{\equiv} \lambda x.\ x^2 + 1$

$f\ x \overset{\text{def}}{\equiv} x^2 + 1$

directly

indirectly

Similar idea on sets:

$A \overset{\text{def}}{\equiv} \{ \dots \}$

$x \in A \overset{\text{def}}{\iff} \dots$

Let $f : \alpha \to \beta$ be the function defined by:

for each $a : \alpha$, $f\ a \overset{\text{def}}{\equiv} \dots\dots$

vs.

Let $f \overset{\text{def}}{\equiv} \lambda a. \dots\dots : \alpha \to \beta$.

λ, $\mapsto$

How to pronounce « $(x \mapsto x^2 + 1)$ »:

« the function that given x returns $x^2 + 1$ »

(silent)

$(x \mapsto x^2 + 1)$

We also write: $\lambda x . x^2 + 1$

Compare with « $\{ x \mid x \text{ is nice} \}$ »:

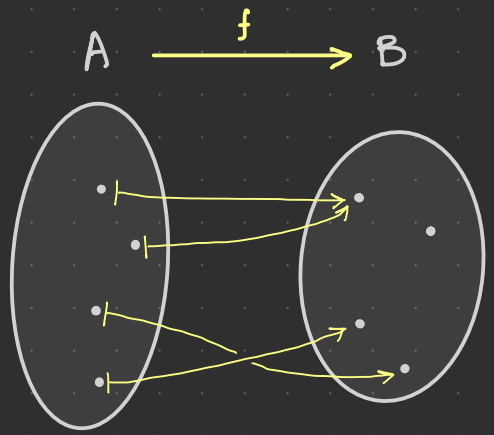
(silent)

« the set of all x such that x is nice »

⚠ ($\mapsto$) vs ($\to$)

$\mapsto$ goes from "points" to "points"

$\to$ goes from types / sets / spaces ⋮ to types / sets / spaces ⋮



Convention:

$$\lambda x . \lambda y . \lambda z . (x+y)^z \equiv \lambda x . (\lambda y . (\lambda z . ((x+y)^z)))$$

$(\to), (\times) : ?$

Attempt: $(\to) : (\text{Type} \times \text{Type}) \to \text{Type}$

$(\times) : (\text{Type} \times \text{Type}) \to \text{Type}$

(Binary) constructions on types

$$\frac{\alpha : \text{Type} \qquad \beta : \text{Type}}{\alpha \to \beta : \text{Type}} \; (\to)\text{-F}$$

formation

$$\frac{\alpha : \text{Type} \qquad \beta : \text{Type}}{\alpha \times \beta : \text{Type}} \; (\times)\text{-F}$$

Internal ops vs external constructions

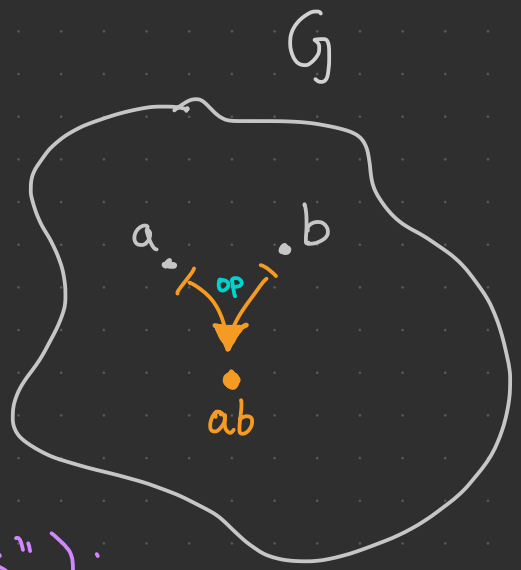
(internal) ops of groups

$(G\ ;\ op,\ inv,\ id)$

$op : G \times G \to G$

$id : G$

$inv : G \to G$



external "big" ops ("constructions"):

$G \times H,\quad G \oplus H,\quad Aut(G),\quad \ldots$

$\alpha \times \beta$ (product)

Formation product construction

$$\frac{A \text{ type} \qquad B \text{ type}}{A \times B \text{ type}} \; (\times)\text{-F}$$

↖ $(\times)$ is a binary construction on types.

What about n-ary?

($n \geq 3$ is easy $n := 1$? $n := 0$?)

Introduction pairing

$$\frac{a : A \qquad b : B}{\langle a, b \rangle_{A \times B} : A \times B} \; (\times)\text{-I}$$

$\langle a, b \rangle$ construtor of inhabitants (elements) of type $\alpha \times \beta$.

could have chosen a less culturally biased notation: pair a b or pair(a, b) ...

... and then set $\langle a, b \rangle \overset{\text{sug}}{\equiv}$ pair a b

Elimination

projections

$$\frac{w : A \times B}{w.\mathtt{l} : A}\ (\times)\text{-}E_1 \qquad \frac{w : A \times B}{w.\mathtt{r} : B}\ (\times)\text{-}E_2$$

alternative notations:

$w \cdot 1$	$w \cdot \mathtt{l}$	$\mathtt{outl}\ w$	$\pi_1\ w$	$\mathtt{fst}\ w$
$w \cdot 2$	$w \cdot \mathtt{r}$	$\mathtt{outr}\ w$	$\pi_2\ w$	$\mathtt{snd}\ w$

Equations

$$\langle a, b \rangle.\mathtt{l} = a$$
$$\langle a, b \rangle.\mathtt{r} = b \qquad (\beta)$$

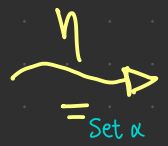
computation (reductions)

$$w = \langle w.\mathtt{l}, w.\mathtt{r} \rangle \qquad (\eta)$$

uniqueness (?)

β, η in sets

$a \in_\alpha \{ x \overset{\cdot\cdot\alpha}{|} \varphi(x) \}$  $\varphi(a)$

$\{ x \overset{\cdot\cdot\alpha}{|} x \in_\alpha A \}$  A

Functions: notation of application

$$\frac{f : \alpha \to \beta \qquad a : \alpha}{f\ a : \beta}\ (\to)\text{-E}$$

how we use elements of $\alpha \to \beta$ (functions)

other common notations: $f(a)$, a^f, af, ...

$f(a,b)$ ← nope!

$f\ \langle a,b \rangle$ ← yeap! ← uncurried $(f : \alpha \times \beta \to \gamma)$

invisible

$(f\ a)\ b$ ← curried $(f : \alpha \to \beta \to \gamma)$

...then what is the arity of a function?

Syntactic aspects

⚠ Do not confuse syntactic associativity (a syntactic decision) with associativity (a mathematical property)!

- Syntactic associativity (gives meaning) resolves this: $a \heartsuit b \heartsuit c$ (Proposition)
- Syntactic precedence resolves this: $a \heartsuit b * c$
- Default op (juxtaposition) resolves this: $a\,b$

set L-(syntactic) associativity for $\sqcup$. $f\ a\ b\ c \equiv ((f\ a)\ b)\ c$

set R-(syntactic) associativity for $\to$. $\alpha \to \beta \to \gamma \to \delta \equiv \alpha \to (\beta \to (\gamma \to \delta))$

set precedence of $\times$ higher than $\to$ $\alpha \times \beta \to \gamma \times \delta \equiv (\alpha \times \beta) \to (\gamma \times \delta)$

Hello Nat

Common human way

Def (Nat)

- O is a Nat.
- For each n (metavar)
- if n is a Nat, then S n is a Nat
  - yikes! looks like a Prop
- Nothing else.

Some elements of Nat:

O , S O , S (S O) , S (S (S O)) , . . . : Nat

Three neat ways to say this

Listing (introduction) rules

$$\frac{}{O : \mathsf{Nat}}\ \text{ZERO}$$

$$\frac{n : \mathsf{Nat}}{S\ n : \mathsf{Nat}}\ \text{SUCC}$$

(what is the formation rule?)

Listing constructors

type Nat

O : Nat

S : Nat → Nat

Listing forms

type Nat ≝ O | S Nat

it is common to break lines and write it like this:

type Nat ≝ O
| S Nat

Define Bool using all three neat ways!

Functions

lec06
2026-03-13

$$\frac{f \cdot \alpha \to \beta \qquad a : \alpha}{f\ a \cdot \beta}\ (\to)\text{-E}$$

β ?

η ?

x may occur here
(sometimes we write $b(x)$ for the same thing)

$$\frac{x \cdot \alpha \vdash b \cdot \beta}{\vdash \lambda x\ b : \alpha \to \beta}\ (\to)\text{-I}$$

Currying

$$((\alpha \times \beta) \to \gamma) \cong (\alpha \to (\beta \to \gamma))$$

curry (→), uncurry (←), prove, define

curry $f \overset{\text{def}}{\equiv} \lambda a : \alpha \,.\, \lambda b : \beta \,.\, f\langle a, b\rangle$

$$\frac{\dfrac{a : \alpha \qquad b : \beta}{\langle a, b\rangle : \alpha \times \beta}\,(\times)\text{-I} \qquad f : (\alpha \times \beta) \to \gamma}{f\langle a, b\rangle : \gamma}\,(\to)\text{-E}$$

$b : \beta$

$a : \alpha$

$f : \alpha \times \beta \to \gamma$

$$\text{curry} \equiv \lambda f \,.\, \lambda a\ \lambda b \,.\, f\langle a, b\rangle : ((\alpha \times \beta \to \gamma) \to (\alpha \to \beta \to \gamma))$$

$$\text{curry}\ f \equiv \lambda a\ \lambda b \,.\, f\langle a, b\rangle : (\alpha \times \beta \to \gamma) \to (\alpha \to \beta \to \gamma)$$

$$\text{curry}\ f\ a \equiv \lambda b \,.\, f\langle a, b\rangle : (\alpha \times \beta \to \gamma) \to \alpha \to (\beta \to \gamma)$$

$$\text{curry}\ f\ a\ b \equiv f\langle a, b\rangle : (\alpha \times \beta \to \gamma) \to \alpha \to \beta \to \gamma$$

Constructor vs function

It makes sense to ask.

«How much is fact 3?» (Ans. 6)

It does not make sense to ask:

«How much is `S (S O)`?»

↳ sounds like: «How much is 42?»

↳ this is already a fully computed, canonical value of type Nat.

What is Algebra? Algebraic structure? Algebraic theory?

Magma, Semigroup, Monoid, Group, Abel, ...

the following are <u>theorems</u>:

Herstein's group theory is an algebraic theory

Field theory is not algebraic (fields are not algebraic structures)

What is a group-homomorphism

Homomorphisms · respect the protagonists of the theory
the operations of the structure
the "groupness" of groups: op, id, inv

Short definitions are better... Right?

No! A good definition captures and communicates the essence of the concept you are defining

Prove and use **Criteria**!

↳ theorems that let you conclude that something is nice™ without having to verify all that the definition demands.

Eg. in group theory

Similar: what is a subgroup?

$\varphi : G \to H$ homo iff φ respects the op ← BAD!

$\varphi : G \to H$ homo iff φ respects the operations (op, id, inv)

CRITERION: $\varphi : G \to H \vdash \varphi$ respects op $\Rightarrow \varphi$ homo

About induction

$$\frac{\varphi(0) \qquad (\forall k{:}\mathrm{Nat})[\varphi(k) \Rightarrow \varphi(s)]}{(\forall n{:}\mathrm{Nat})[\varphi(n)]}\ \mathrm{IND}^{\mathrm{Nat}}_{\varphi}$$

-- $\Gamma \vdash (\forall n \cdot \mathrm{Nat})[\varphi(n)]$

Induction.

BASE: -- $\Gamma \vdash \varphi(0)$

INDUCTIVE STEP: -- $\Gamma \vdash (\forall k : \mathrm{Nat})[\varphi(k) \Rightarrow \varphi(s\,k)]$

probably we will proceed with:

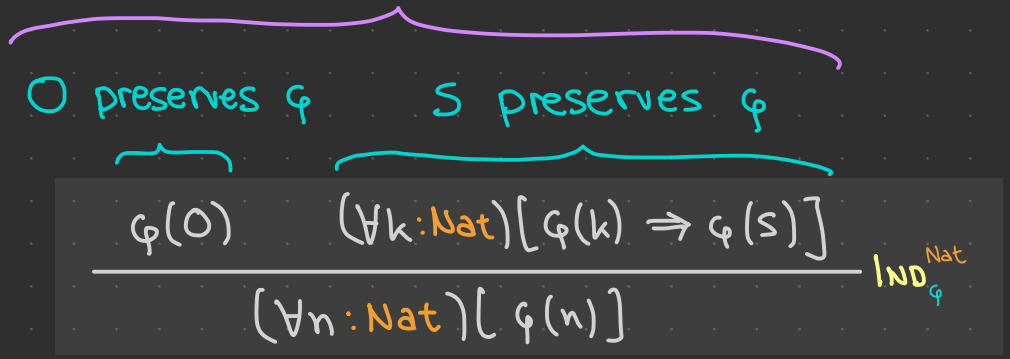
Let $t \cdot \mathrm{Nat}$. -- $\Gamma, t : \mathrm{Nat} \vdash \varphi(t) \Rightarrow \varphi(s\,t)$

Suppose $\varphi(t)$. -- $\Gamma, t : \mathrm{Nat}, \varphi(t) \vdash \varphi(s\,t)$

abbrev: Let $t \cdot \mathrm{Nat}$ such that $\varphi(t)$.

How to ♡-understand the $\text{IND}_{\varphi}^{\text{Nat}}$ rule

all constructors preserve $\varphi : \text{Nat} \to \text{Prop}$



Hello, Bool!

What is the induction rule for Bool?

(if there is one)

How can we construct bools?

$$\frac{}{\mathit{ff} : \mathsf{Bool}}\ \text{Bool-I}_F \qquad \frac{}{\mathit{tt} : \mathsf{Bool}}\ \text{Bool-I}_T$$

Or:

```
type Bool
  ff : Bool
  tt : Bool
```

How can we use bools? (E)

$$\frac{b : \mathsf{Bool} \qquad y : \alpha \qquad n : \alpha}{\text{if } b \text{ then } y \text{ else } n : \alpha}\ \text{Bool-E}$$

```
branch (y, n ; b)
ifthenelse b y n
```

⋮

β

if tt then y else n = y

if ff then y else n = n

η

if b then tt else ff = b

Lists

lec07
2026-03-17

Examples.

$[3, 2, 2, 3, 8] : \text{List Nat}$

$[] : \text{List Nat}$

$[] : \text{List Person}$

$[[\text{Gentzen}, \text{Brouwer}], [], [\text{Galois}]] : \text{List (List Person)}$

$[] : \text{List (List (Nat} \times \text{Nat} \to \text{Prop))}$

$[(+)] : \text{List (Nat} \times \text{Nat} \to \text{Nat)}$

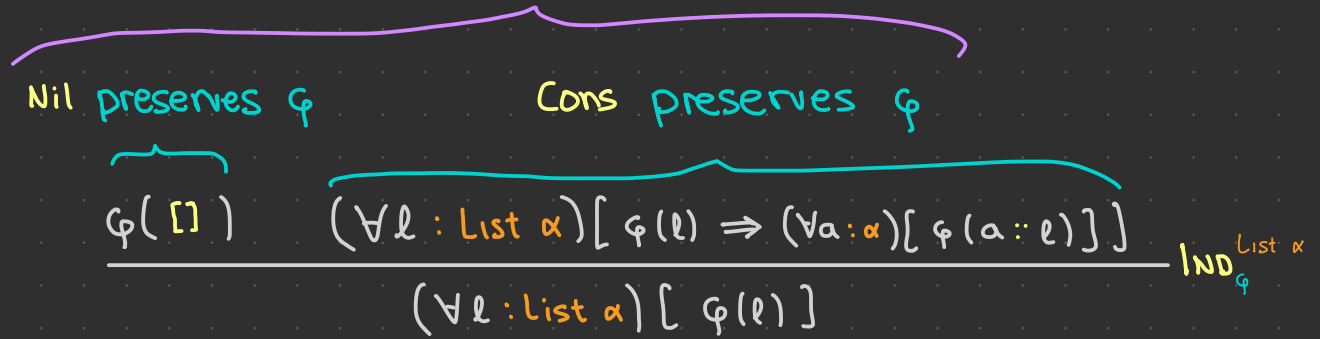
$$\text{List} : \text{Type} \to \text{Type} \qquad \frac{\alpha : \text{Type}}{\text{List } \alpha : \text{Type}}\ \text{List-F}$$

$\text{Nil} : \text{List } \alpha$ (also: $[]$)

$\text{Cons} : \alpha \to \text{List } \alpha \to \text{List } \alpha$ (also: $\_::\_$)

Induction on lists

(♡) all constructors preserve $\varphi : \text{List } \alpha \to \text{Prop}$



Type arithmetic

$$(\alpha + \beta) + \gamma \overset{?}{\cong} \alpha + (\beta + \gamma)$$

$$\mathbb{0} + \alpha \overset{?}{\cong} \alpha$$

$$\alpha + \beta \overset{?}{\cong} \beta + \alpha$$

$$(\alpha \times \beta) \times \gamma \overset{?}{\cong} \alpha \times (\beta \times \gamma)$$

$$\mathbb{1} \times \alpha \overset{?}{\cong} \alpha$$

$$\alpha \times \beta \overset{?}{\cong} \beta \times \alpha$$

Distributivities

$$\delta \times (\alpha + \beta) \cong ?$$

$$\delta \times \mathbb{0} \cong ?$$

(future?)

$$\delta^{(\alpha \times \beta)} \cong ?$$

$$\delta^{(\alpha + \beta)} \cong ?$$

$$(\alpha \times \beta)^{\delta} \cong ?$$

$$(\alpha + \beta)^{2} \cong ?$$

what is that?!

Natural deduction (PT)

Two forms of judgement:

- A prop (means A is a meaningful proposition)
- A true (means I have a proof of A)

Entailment ($\vdash$): $\Gamma \vdash J$ (J: judgement; Γ: list of judgements)

$$\frac{A \text{ prop} \qquad B \text{ prop}}{A \wedge B \text{ prop}}\ F$$

$$\frac{A \text{ true} \qquad B \text{ true}}{A \wedge B \text{ true}}\ \text{Intro}$$

$$\frac{A \wedge B \text{ true}}{A \text{ true}}\ E \qquad \frac{A \wedge B \text{ true}}{B \text{ true}}\ E$$

$(\wedge)$

$$\frac{A \text{ prop} \qquad B \text{ prop}}{A \wedge B \text{ prop}}\ (\wedge)\text{-F}$$

$$\frac{A \text{ true} \qquad B \text{ true}}{A \wedge B \text{ true}}\ (\wedge)\text{-I}$$

$$\frac{A \wedge B \text{ true}}{A \text{ true}}\ (\wedge)\text{-E}_L \qquad \frac{A \wedge B \text{ true}}{B \text{ true}}\ (\wedge)\text{-E}_R$$

Here (∧) we also have β, η →

← What are the β, η here? (×)

Example: $A \wedge B$ true $\vdash$ $B \wedge A$ true

open leaves

$$\dfrac{\dfrac{A \wedge B \text{ true}}{B \text{ true}}\ \wedge\text{-E}_R \qquad \dfrac{A \wedge B \text{ true}}{A \text{ true}}\ \wedge\text{-E}_L}{B \wedge A \text{ true}}\ \wedge\text{-I}$$

Compare with (×):

Formation product construction

$$\frac{A \text{ type} \qquad B \text{ type}}{A \times B \text{ type}}\ (\times)\text{-F}$$

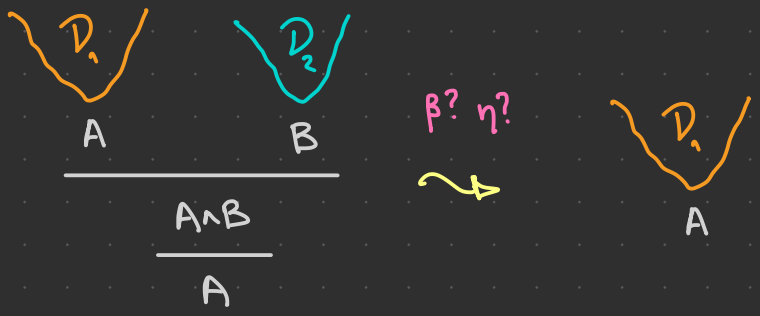
Introduction pairing

$$\frac{a : A \qquad b : B}{\langle a, b \rangle_{A \times B} : A \times B}\ (\times)\text{-I}$$

Elimination projections

$$\frac{w : A \times B}{w.\mathrm{l} : A}\ (\times)\text{-E}_L \qquad \frac{w : A \times B}{w.\mathrm{r} : B}\ (\times)\text{-E}_R$$

Proofs compute! You can run them!



Propositions – types

PT	TT
$\wedge$	$\times$
$\supset$	$\to$
$\bot$	0
$\top$	1
$\vee$	$+$
$\forall$	?
$\exists$	?
$=$	?

also 0

```
Empty : Type
```

nothing to see here

no constructors!

```
Unit : Type
```

only one constructor → ★ : Unit

(therefore): no information

Who is void?

```
int f(void)
    return 42;
```

```
void g(int x)
    return;
```

$$\frac{f : \text{void} \to \text{int} \qquad a \cdot \text{void}}{f(a) : \text{int}}$$

$$\frac{g : \text{int} \to \text{void} \qquad 42 \cdot \text{int}}{g(42) : \text{void}}$$

what the programmer perceives as argument(s) is in orange as "function applicator" in cyan:

$f(a_1, \dots, a_n)$ — bad notation

$f\,\langle a_1, \dots, a_n \rangle$ — n-ary, good notation

$f\,a_1\,\dots\,a_n$ — n-ary, good notation

n-ary uncurried

n-ary curried

$f()$ ← gives the fall impression that f received nothing (and yet somehow it got used/called ?!?)

$f\,()$ ← makes it clear that f received $()$ (different notation for our $\star$)

the only inhabitant of $\mathbb{1}$

$(\vee)$

$$\frac{A \text{ prop} \qquad B \text{ prop}}{A \wedge B \text{ prop}}\,(\vee)\text{-F}$$

$$\frac{A \text{ true}}{A \vee B \text{ true}}\,(\vee)\text{-I}_L \qquad \frac{B \text{ true}}{A \vee B \text{ true}}\,(\vee)\text{-I}_R$$

$$\frac{A \vee B \text{ true} \qquad A \text{ true} \vdash G \text{ true} \qquad B \text{ true} \vdash G \text{ true}}{G \text{ true}}\,(\vee)\text{-E}$$

β, η ?

(+)

$$\frac{\alpha : \mathrm{Type} \qquad \beta : \mathrm{Type}}{\alpha + \beta : \mathrm{Type}}\ (+)\text{-F}$$

$$\frac{a : \alpha}{l \cdot a : \alpha + \beta}\ (+)\text{-I}_L$$

also: inl a

$$\frac{b : \beta}{r \cdot b : \alpha + \beta}\ (+)\text{-I}_R$$

also: inr b

What's missing here?

(Discover it in multiple ways!)

lec08

2026-03-20

(+)

$$\frac{s : \alpha + \beta \qquad f : \alpha \to \delta \qquad g : \beta \to \delta}{\begin{pmatrix} \texttt{match}\ s\ \texttt{with} \\ l \cdot a \rightsquigarrow f\ a \\ r \cdot b \rightsquigarrow g\ b \end{pmatrix} : \delta}\ (+)\text{-E}^*$$

also:

$$\begin{pmatrix} \texttt{case}\ s\ \texttt{of} \\ \vdots \end{pmatrix}$$

this relies on $(\to)$

HW: write an independent $(+)$-E

Hint: work your way one step up from the leaves which use $(\to)$.

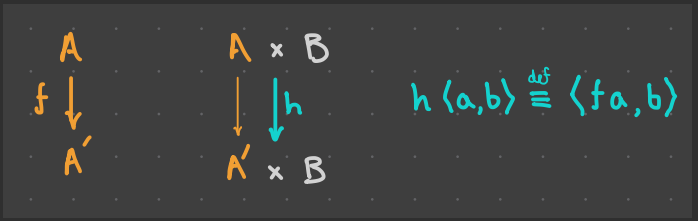
β

$$[f \mid g]\,(l \cdot a) = f\ a$$
$$[f \mid g]\,(r \cdot b) = g\ b$$

η

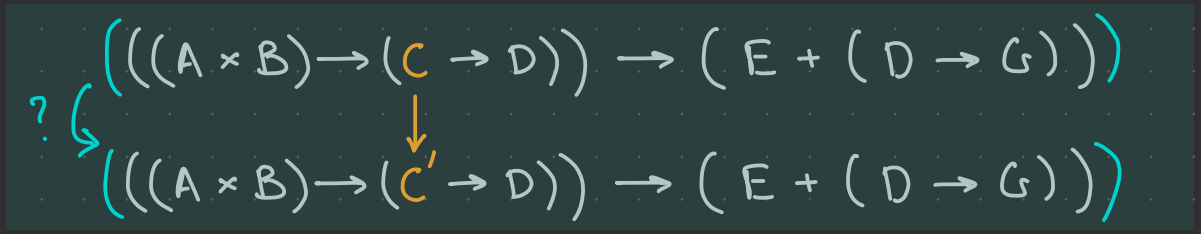
$$s = [\,\mathrm{inl} \mid \mathrm{inr}\,]\ s$$

Functions for free

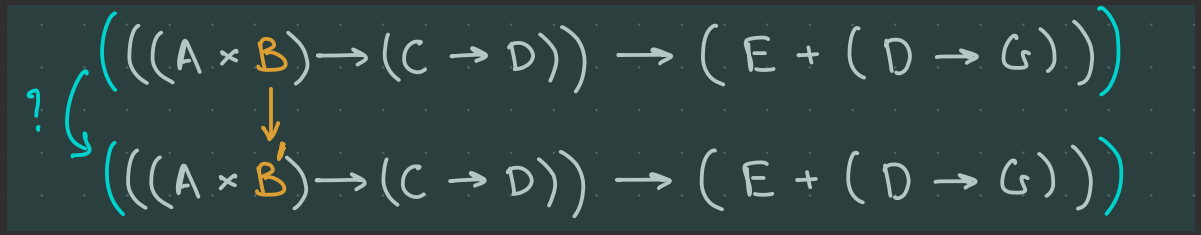


given: can we obtain for free...:

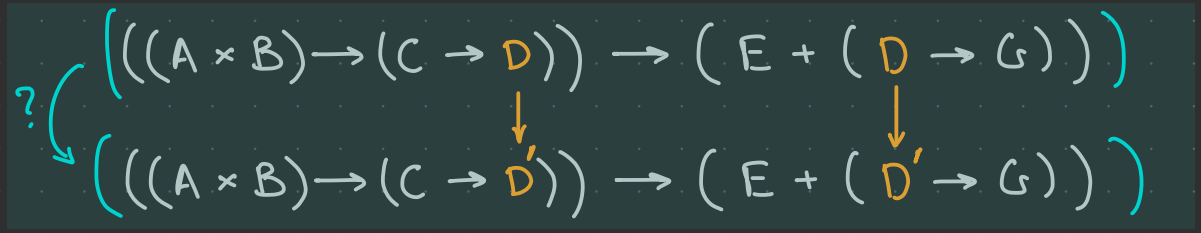
C ↓ C′



B ↓ B′



D ↓ D′



Dependent types / Families

"$D : I \to \mathrm{Type}$"

(I: indexing)

$i : I \vdash D(i)$ type

$i : I \vdash D(i)$ type

D is a family of types (indexed by $i : I$)

D is a dependent type (on $i : I$)

Examples:

"$\mathbb{R}^{\square} : \mathbb{N} \to \mathrm{Type}$"

$n : \mathbb{N} \vdash \mathbb{R}^n$ type

Contrast with $\mathbb{R}^3$

"$A : \mathbb{B} \to \mathrm{Type}$"

$A(\#) \overset{\text{def}}{\equiv} \mathbb{N}$

$A(\#) \overset{\text{def}}{\equiv} \mathbb{N} \times (\mathbb{N} \to \mathbb{N})$

$$b : \mathbb{B} \vdash A(b) \text{ type}$$

Contrast with $\mathcal{C}[0,1]$

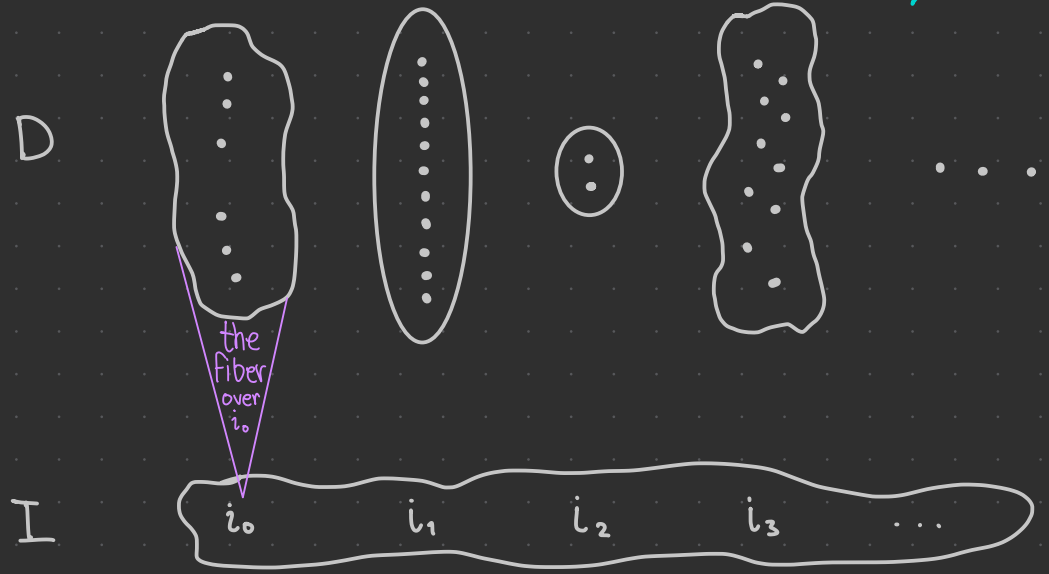
$a : \mathbb{R},\ b : \mathbb{R} \vdash \mathcal{C}[a,b]$ type

Depicting $D : I \to \mathrm{Type}$

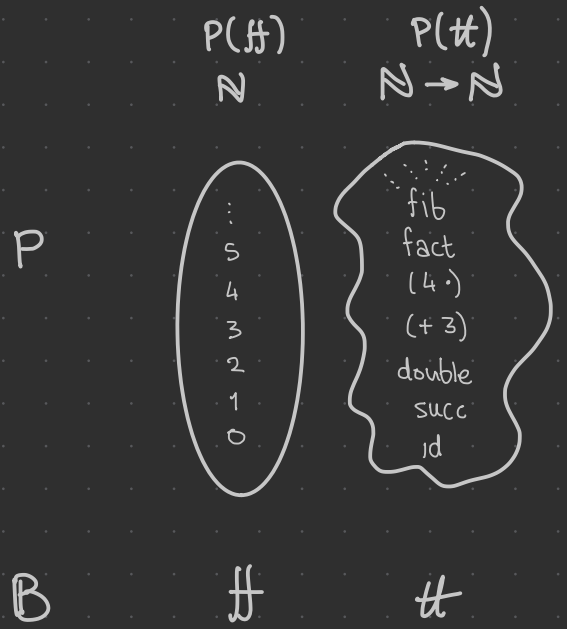
$$i : I \vdash D(i)\ \mathrm{type}$$

How many?

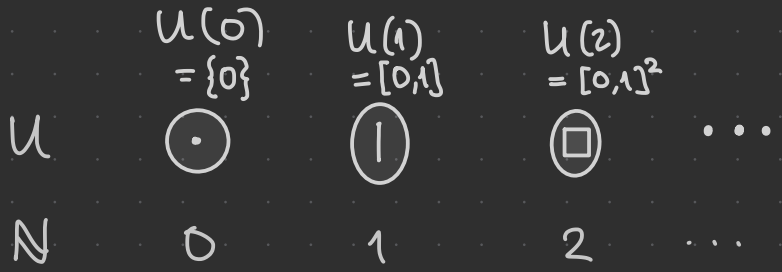
I-many types



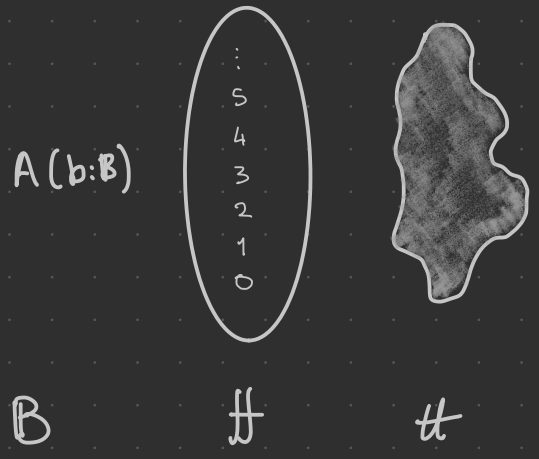
$b : \mathbb{B} \vdash P(b)$ type



$n : \mathbb{N} \vdash U(n)$ type



$b : \mathbb{B} \vdash A(b)$ type



Contexts

implicit : $a : A \vdash B$ type

$a : A$, $b : B$, $c : C$, ...

implicit : $\vdash A$ type

implicit : $a : A, b : B \vdash C$ type

$a_1 : A_1$, $a_2 : A_2(a_1)$, $a_3 : A_3(a_1, a_2)$, ...

Writing : $n : \mathbb{N}$, $u : \mathbb{R}^n$, $v : \mathbb{R}^n$, $m : \mathbb{N}$, $A : \mathbb{Z}^{n \times m}$

Implicit : $\vdash \mathbb{N}$ type

$n : \mathbb{N} \vdash \mathbb{R}^n$ type

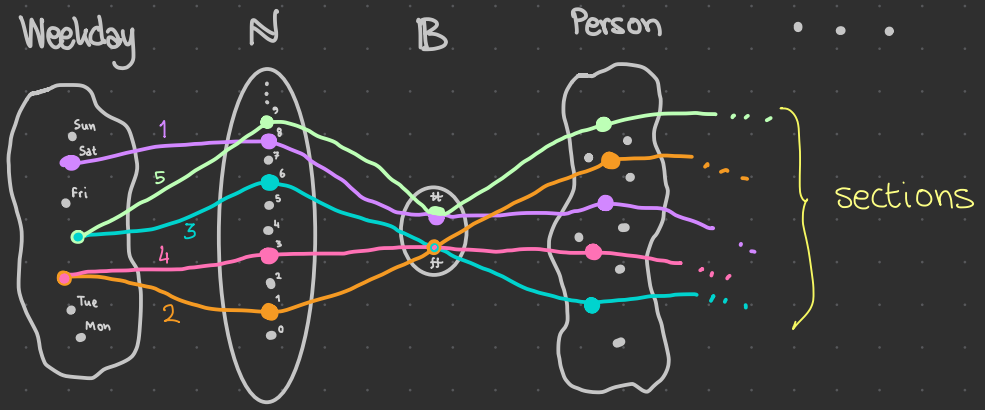
$n : \mathbb{N}, u : \mathbb{R}^n \vdash \mathbb{R}^n$ type

$n : \mathbb{N}, u : \mathbb{R}^n, v : \mathbb{R}^n \vdash \mathbb{N}$ type

$n : \mathbb{N}, u : \mathbb{R}^n, v : \mathbb{R}^n, m : \mathbb{N} \vdash \mathbb{Z}^{n \times m}$ type

5 inhabitants of $\mathbb{T}\mathbb{D}$

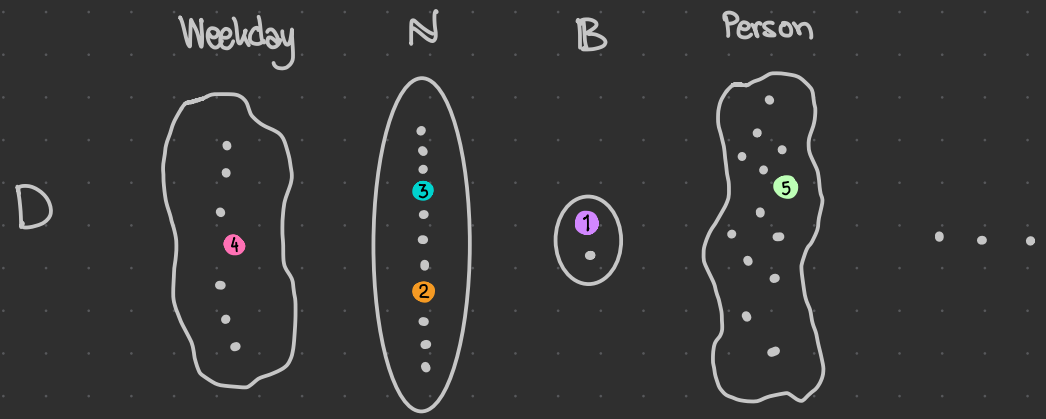
D



I

How to use: $f_3\ i_1 = 6$ $f_3\ i_2 = ff$ $f_1\ i_0 = Sat$

5 inhabitants of ΣD



How to use: $w_3.l = i_1$ $\quad$ $\mathrm{out}\,l\ w_1 = i_2$

$w_3.r = 7$ $\quad$ $\mathrm{out}\,r\ w_1 = tt$

we may choose more indicative names than left/right.
More on this later:
when we define the notion of PRODUCT.

Π, Σ : formation rules

D is a dependent type

$$\frac{i : I \vdash D(i)\ \text{type}}{\vdash \prod(i:I).\ D(i)\ \text{type}}\ (\Pi)\text{-F}$$

also: $\prod_{i:I} D(i)$

also: $(i:I) \to D(i)$

$$\frac{i : I \vdash D(i)\ \text{type}}{\vdash \sum(i:I).\ D(i)\ \text{type}}\ (\Sigma)\text{-F}$$

also: $\sum_{i:I} D(i)$

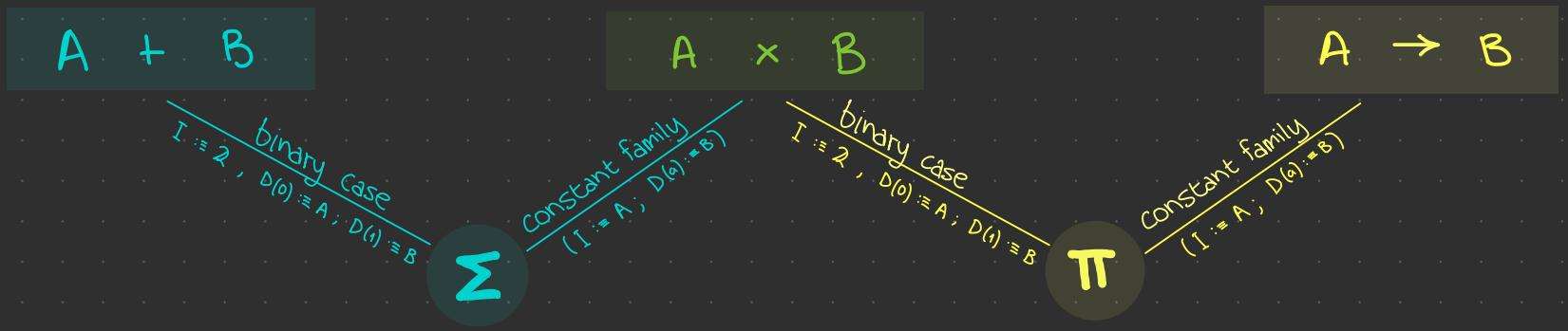
also: $(i:I) \times D(i)$

E, I & β, η

(like) functions

(like) pairs

Special cases of Σ, Π



constant family : $I :\equiv A \; ; \; D(a) :\equiv B$

$$\sum_{(a:A)} D(a) \equiv (a:A) \times B \equiv \boxed{A \times B} \qquad \prod_{(a:A)} D(a) \equiv (a:A) \to B \equiv \boxed{A \to B}$$

binary case : $I :\equiv 2 \; ; \quad D(0) :\equiv A \; ; \quad D(1) :\equiv B$

$$\sum_{(i:2)} D(i) \cong D(0) + D(1) \equiv \boxed{A + B} \qquad \prod_{(i:2)} D(i) \cong D(0) \times D(1) \equiv \boxed{A \times B}$$

(⊢) Entailment

Properties: Are these like axioms or inferable (provable)?

list of hypotheses

$$\Gamma \vdash J$$

hypothetical judgement

use of hypothesis

$$\frac{A \text{ prop}}{A \text{ true} \vdash A \text{ true}}\ (\text{AXIOM})\ \text{ID REFL}$$

Lemma

use of lemma

$$\frac{\Gamma \vdash L \text{ true} \qquad L \text{ true} \vdash A \text{ true}}{\Gamma \vdash A \text{ true}}\ (\text{CUT})\ \text{COMP TRANS}$$

you may leave hypotheses unused

$$\frac{\Gamma \vdash A \text{ true}}{\Gamma, H \text{ true} \vdash A \text{ true}}\ (\text{WEAKENING})$$

you may use your hypotheses as many times as you please

$$\frac{\Gamma, H \text{ true}, H \text{ true} \vdash A \text{ true}}{\Gamma, H \text{ true} \vdash A \text{ true}}\ (\text{CONTRACTION})$$

$$\frac{\Gamma \vdash A \text{ true}}{\Gamma^* \vdash A \text{ true}}\ (\text{PERM})\ \text{EXCHANGE}$$

$\ldots, A, B, \ldots \vdash$
$\ldots, B, A, \ldots \vdash$

Γ^*: a valid permutation of Γ

Order-theoretic view

Write: $A \leq B \overset{\text{def}}{\iff} A \text{ true} \vdash B \text{ true}$

defend this name (order: refl & trans & antisym)

We investigate the world of propositions ordered by ($\leq$)

Pre.

↳ this needs proof: (refl & trans)

Verify

- Has (binary) meets given by ($\wedge$)

$$\frac{}{A \wedge B \leq A} \qquad \frac{}{A \wedge B \leq B}$$

} $A \wedge B$ is a lower bound of A, B

some candidate/impostor

$$\frac{C \leq A \qquad C \leq B}{C \leq A \wedge B}$$

} ... is the best.

- Has top given by ($\top$)

- Has (binary) joins given by $(\wedge)$

$$\frac{}{A \wedge B \leq A} \qquad \frac{}{A \wedge B \leq B}$$

} $A \wedge B$ is a lower bound of A, B

$$\frac{C \leq A \qquad C \leq B}{C \leq A \wedge B}$$

(some candidate/impostor)

} ... is the best.

- Has bottom given by $(\bot)$

Another example: the world of integers pre-ordered by $(\mid)$

(How does their Hasse diagram look like?

Does it have: meets, top, joins, bottom?)

- ~~the~~ a greatest common divisor

→ the common $(\leq)$ order is completely foreign and irrelevant here!

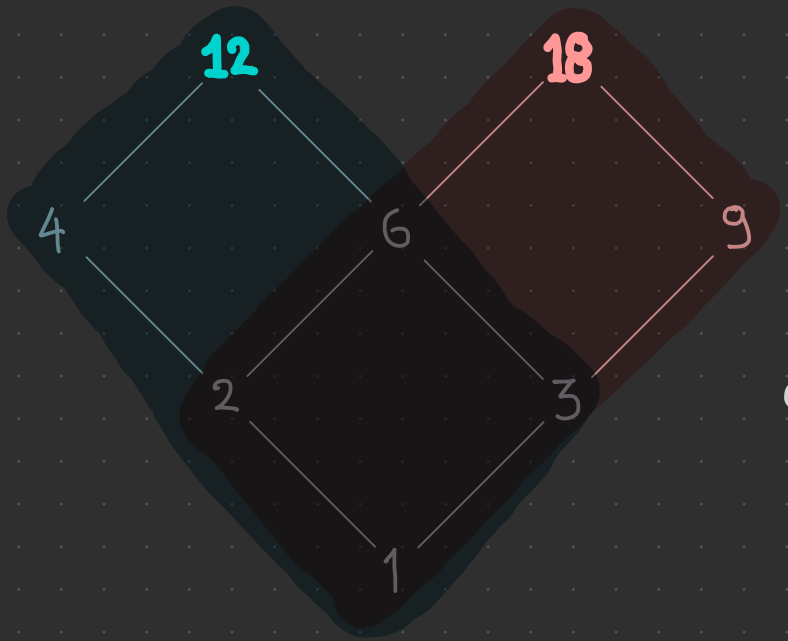
$$\begin{aligned} \text{divs } 6 &= \{ \pm 1, \pm 2, \pm 3, \pm 6 \} \\ \text{divs } 9 &= \{ \pm 1, \pm 3, \pm 9 \} \\ \text{comdivs } 6\ 9 &= \{ -1, 1, -3, 3 \} \end{aligned}$$

(-3 has every right to be called a g.c.d.!)

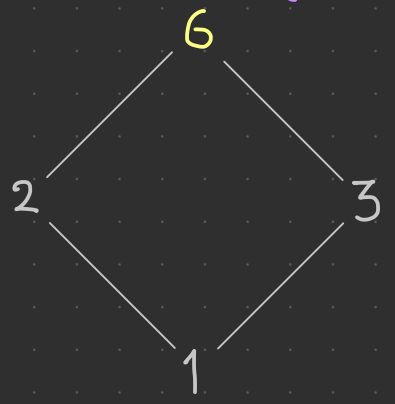
spell-out the Correct Definition™

What if a student arrived late..?

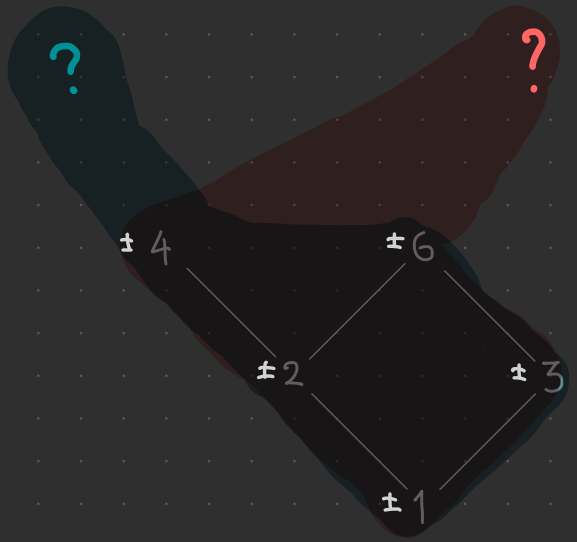
lec09
2026-03-24



comdivs =

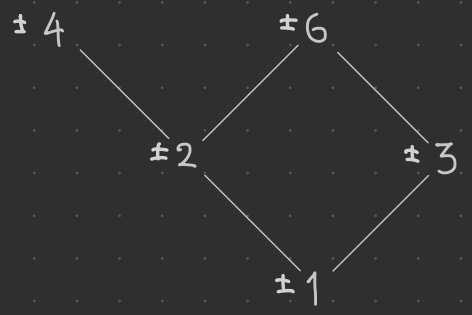


But if it was like that?



comdivs =

+6 (≤)-dominates.
BUT: Nobody cares about (≤) here.
No element manages to (1)-dominate!



↑

Euclid proves (constructively!) this situation is impossible! **How?**

gcd : best common divisor

Def Let $m, a, b : \mathrm{Int}$.

m is a greatest common divisor (g.c.d.) of a, b

$\Leftrightarrow$ (def)

$$\begin{cases} \underbrace{m \text{ is a common divisor}}_{m \mid a \;\&\; m \mid b} \\ \& \\ \underbrace{(\forall c)[\, c \text{ com div. of } a, b \Rightarrow c \mid m \,]} \\ \gcd(a,b) \quad m \text{ is the best} \end{cases}$$

$\wedge \vee \Rightarrow \forall \exists =$

similar with $\underline{\ell} = \lim (a_n)_n$ ($\forall$-prop in $(=)$-clothing)

We write $m = (a, b)$. → what kind of Prop is this?

deceiving notation: $\wedge$-prop in $(=)$-clothing

$\underline{m} = (\underline{a}, \underline{b}) : \mathrm{Int} \times \mathrm{Int} \times \mathrm{Int} \to \mathrm{Prop}$

P(r)oset ← partially ordered set (preordered set)

$\mathcal{P} \equiv (P : \text{carrier}; \leq)$

$\_\leq\_ : \underbrace{\text{Rel } P}_{P \times P \to \text{Prop}}$

Laws:

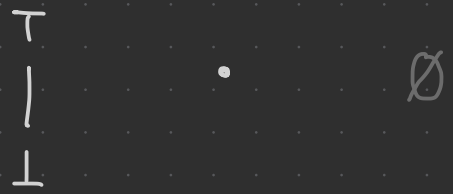
- $(\leq)$ - refl
- $(\leq)$ - trans

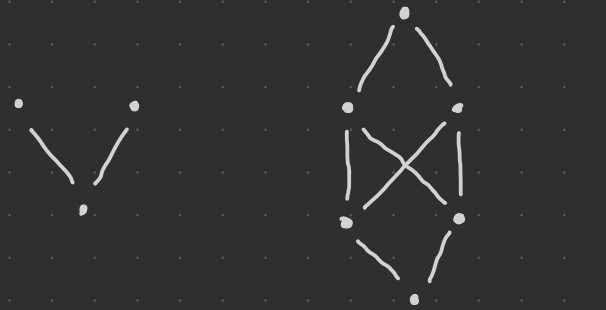
} pre-order

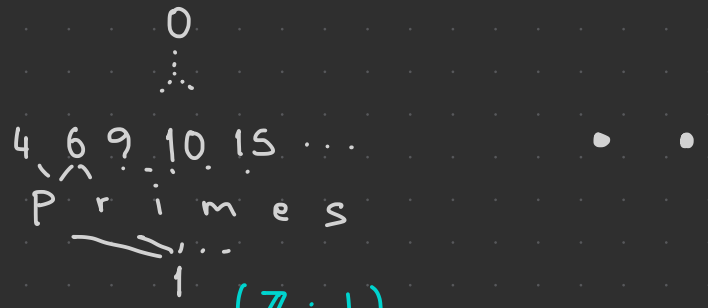
- $(\leq)$ - antisym

} (partial) order

Examples:







What are the transportation means?

$P \equiv (P ; \leq_P) \xrightarrow{\varphi} Q \equiv (Q ; \leq_Q)$

$|\varphi| : P \to Q$ st.:

- φ preserves $(\leq_P) \iff x \leq_P y \Rightarrow \varphi x \leq_Q \varphi y$

  also: φ $(\leq)$-monotone

  φ $(\leq)$-isotone

Bonus vocabulary:

φ reflects $(\leq_Q) \iff x \leq_P y \Leftarrow \varphi x \leq_Q \varphi y$

φ $(\leq)$-antitone $\iff x \leq_P y \Rightarrow \varphi y \leq_Q \varphi x$

order-theoretic O-Lattices

$P \equiv (P; \leq)$: Poset

P O-Lattice $\overset{\text{def}}{\iff}$ P has (all) binary lubs & glbs

(bounded $\overset{\text{def}}{\iff}$ ... & has bottom & top.)

0. In any poset, lubs & glbs are unique.

Therefore in any O-lattice we obtain binary operations: $(\vee)$, $(\wedge)$.

$a \vee b \overset{\text{def}}{=}$ the unique lub $\{a, b\}$

$a \wedge b \overset{\text{def}}{=}$ the unique glb $\{a, b\}$

a-Lattice (algebraic)

↝ o-lattice

$\mathcal{L} \equiv (L; \vee, \wedge)$

Define: $x \leq y \overset{\text{def}}{\iff} x \vee y = y$ & $x \wedge y = x$ — either one implies the other

$\mathcal{L} \equiv (L; \vee, \wedge, \bot, \top)$ ($\{\bot, \top\}$-lattice)

$\_\vee\_ : L \times L \to L$
$\_\wedge\_ : L \times L \to L$
$\bot : L$
$\top : L$
} ops on L

∴ Algebra! Yay!

EQUATIONS:

$\vee$-ass	$\wedge$-ass
$\vee$-com	$\wedge$-com
$\bot$-id-$\vee$	$\top$-id-$\wedge$
$\vee$-idem	$\wedge$-idem
$\vee,\wedge$-abs	$\wedge,\vee$-abs
⇕ def	⇕ def
$(a \wedge b) \vee a = a$	$(a \vee b) \wedge a = a$

Θ. $(L; \leq)$
... is an o-lattice — case $\mathcal{L} \equiv (L; \vee, \wedge)$
.. is a bounded o-lattice — case $\mathcal{L} \equiv (L; \vee, \wedge, \bot, \top)$

Spell-out what (bounded) a-lattice homo means.

Spell-out what sub-a-lattice means.

Θ? Let $\mathcal{L} \equiv (L; \vee, \wedge)$.
Let $S \subseteq L$
Suppose S equipped with the order inherited from $\mathcal{L}$ (def?) is a lattice.
Is S a sublattice of $\mathcal{L}$?

kernels

Let $f : A \to B$.

Define $\ker f : \overbrace{A \times A \to \mathrm{Prop}}^{\mathrm{Rel}_2\, A}$ by

$$x \ (\ker f)\ y \iff_{\text{def}} f\,x = f\,y$$

f identifies x, y: $x \mapsto \bullet \leftarrow y$

▷ Check: $\ker f$ is an equivalence relation

▷ Let $\varphi : \mathcal{L} \to \mathcal{L}'$.

Investigate the following subsets of L:

- $\varphi^{-1}(\{\bot\})$
- $\varphi^{-1}(\{\top\})$

Existence $(\exists_{\geq 1})$ & Uniqueness $(\exists_{\leq 1})$

lec 10
2026-03-27

is just $(\exists x)$ ↙ — is actually a $(\forall)$ ↙

$$(\exists! x : \alpha)[\varphi(x)] \overset{\text{def}}{\iff} \begin{cases} \text{at least one:} & (\exists x)[\varphi(x)] \quad \text{(existence)} \\ & \& \\ \text{at most one:} & (\forall u, v)[\varphi(u) \,\&\, \varphi(v) \Rightarrow u = v] \quad \text{(uniqueness)} \end{cases}$$

How do we prove uniqueness? (proof by Fight Club)

$\vdash (\forall u, v : \alpha)[\varphi(u) \,\&\, \varphi(v) \Rightarrow u = v]$

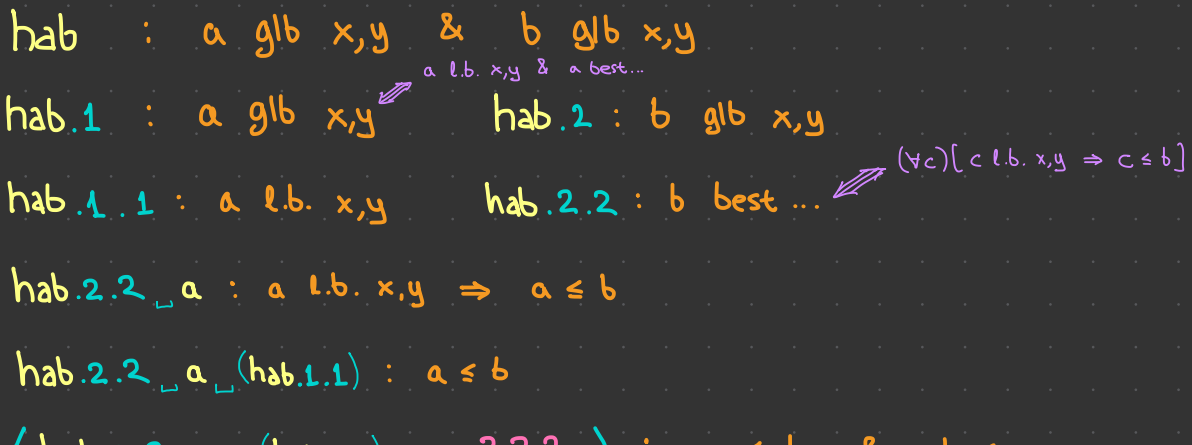
Let $u, v : \alpha$ such that $\varphi(u) \,\&\, \varphi(v)$. $u, v : \alpha,\ \varphi(u) \,\&\, \varphi(v) \vdash u = v$

-- Goal: $u = v$

⋮

Σε annoyingly low-level: a, b, x, y · P ⊢ a glb x,y & b glb x,y ⇒ a=b

Suppose a, b are glbs of x, y. (hab) …, hab : a glb x,y & b glb x,y ⊢ a=b
It is sufficient to prove a ≤ b & b ≤ a. [(≤)-antisym] … ⊢ a≤b & b≤a
Part a ≤ b: .. ⊢ a≤b
  Ext-L from (hab) to obtain: a is a glb of x, y (ha). …, ha · a glb x,y ⊢ a≤b
  Ext-R from (hab) to obtain: b is a glb of x, y (hb). …, hb : b glb x,y ⊢ a≤b
  -- Note that (ha) is itself a conjuction: a l.b. x,y ∧ a best l.b. x,y.
  -- Same goes for (hb).
  Ext-L from (ha) to obtain: a l.b. x, y (hal) …, hal : a l.b. x,y ⊢ ..
  Ext-R from (hb) to obtain: (∀c)[c l.b. x, y ⇒ c ≤ b] (hbg) …, hbg · (∀c)[c l.b.x,y ⇒ c≤b] ⊢ …
  Apply ~~(hb)~~ hbg to a to obtain: a l.b. x, y ⇒ a ≤ b (hi). …, hi : a l.b. x,y ⇒ a≤b ⊢ …
  Apply (hi) to (hal) to obtain: a ≤ b.
Part b ≤ a: .. ⊢ b≤a
  (Similar.)


hab : a glb x,y & b glb x,y
a l.b. x,y & a best...
hab.1 : a glb x,y
hab.2 : b glb x,y
(∀c)[c l.b. x,y ⇒ c ≤ b]
hab.1.1 : a l.b. x,y
hab.2.2 : b best ...
hab.2.2 _ a : a l.b. x,y ⇒ a ≤ b
hab.2.2 _ a _ (hab.1.1) : a ≤ b
⟨ hab.2.2 _ a _ (hab.1.1) , ??? ⟩ : a ≤ b & b ≤ a

BHK interpretation

$A \wedge B$ $\quad (a:A,\ b:B) \quad : \quad A \times B$

$A \vee B$ $\quad \begin{cases} (0, a:A) \\ (1, b:B) \end{cases} \quad : \quad A + B$

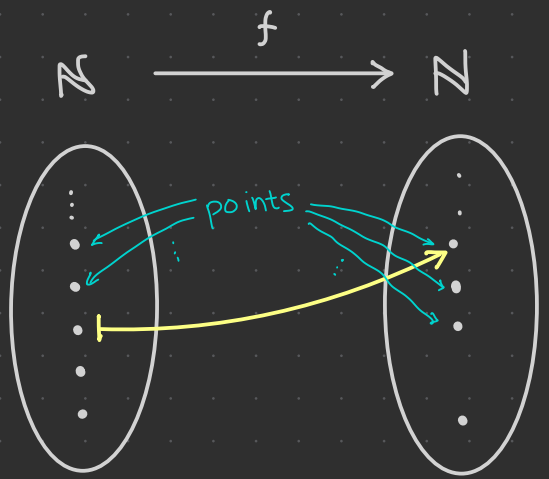
$A \Rightarrow B$ $\quad p \quad : \quad A \to B$

$(\forall a:\alpha)[\varphi(a)]$ $\quad \overbrace{p}^{\text{dependent function}} \quad : (a:A) \to \varphi(a)$ $\quad \prod_{a:\alpha} \varphi(a)$

$(\exists a:\alpha)[\varphi(a)]$ $\quad \overbrace{(\overset{\text{witness}}{w}:\alpha,\ \varphi(w))}^{\text{dependent pair}} \quad : (w:A) \times \varphi(w)$ $\quad \sum_{w:\alpha} \varphi(w)$

Working point-free

aka: "pointless"



points

$f\ n \overset{def}{\equiv} 2\,(n^2+1)$

$f \overset{def}{\equiv} \lambda n.\ 2\cdot(n^2+1)$

points

composition in diagrammatic order

$f;g \overset{def}{\equiv} g \circ f$

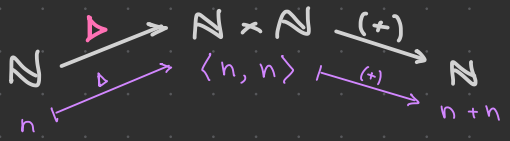
$f \overset{def}{\equiv} \text{double} \circ \text{succ} \circ \text{square}$

$\equiv \text{square} ; \text{succ} ; \text{double}$

$$\text{double} : \mathbb{N} \to \mathbb{N}$$

$$\text{double}\ n \overset{\text{def}}{=} n + n$$

$$\text{double} \overset{\text{def}}{=} (+) \circ \Delta$$



$$\Delta : \alpha \to \alpha \times \alpha$$

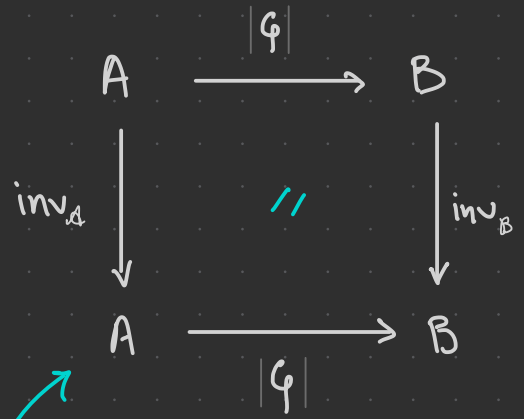
$$\Delta\ a \overset{\text{def}}{=} \langle a, a \rangle$$

should we be satisfied with this def or can we do better (i.e. define Δ point-free)?

Respecting ops expressed via com. diags

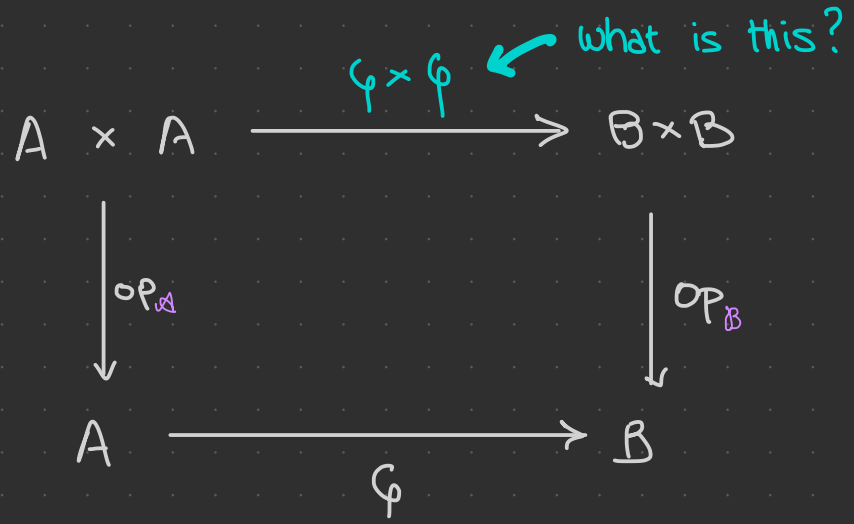
	arities
e	0
inv	1
op	2

φ respects inv ⟺

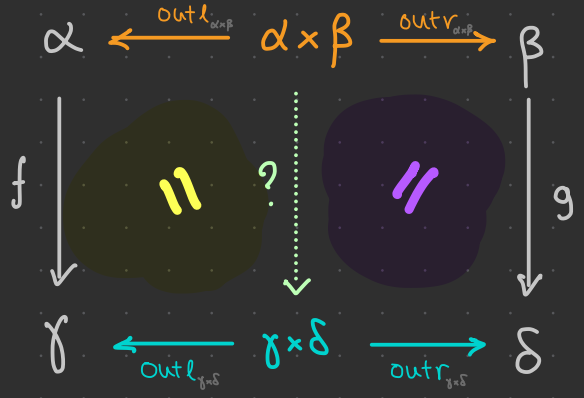


this diagram commutes ⟺ $\varphi \circ inv_A = inv_B \circ \varphi$

e	0
inv	1
op	2



Functions for free: (×)



$$f \times g : \alpha \times \beta \to \gamma \times \delta$$

$$(f \times g)\ w \stackrel{def}{=} \langle f\ (w.l),\ f\ (w.r) \rangle$$

$$(f \times g)\ (a,b) \stackrel{def}{=} \langle f\ a,\ g\ b \rangle$$

$$f \circ outl_{\alpha \times \beta} = outl_{\gamma \times \delta} \circ (f \times g)$$

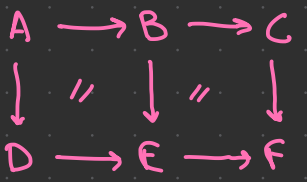
$$g \circ outr_{\alpha \times \beta} = outr_{\gamma \times \delta} \circ (f \times g)$$

Commutative diagrams

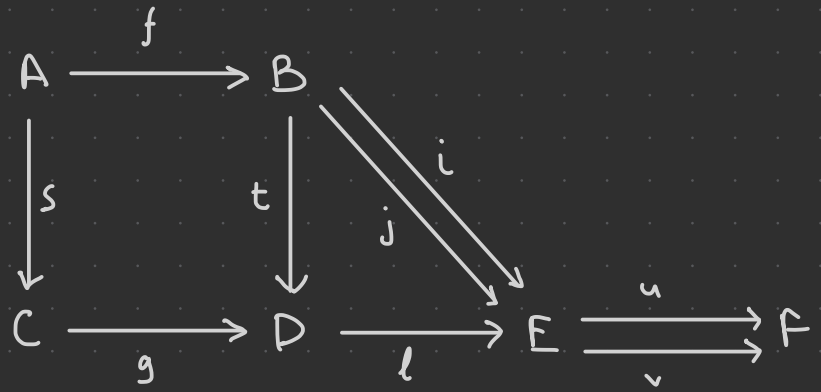
What does it mean that a diagram commutes?

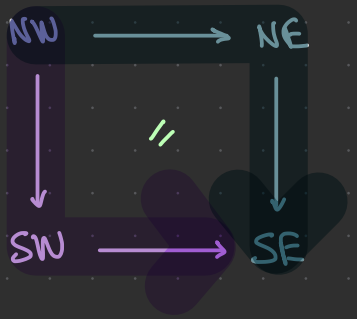
any (same source & target) comparable paths
(one of which is at least of length > 1)
are equal

Check:



the two squares commute
⇓
the whole diag. commutes





def ⇔ sug

$S \circ E = E \circ S$

Constructions on Posets - Posets for free

P^{op} ⇝ turns P upside-down ⇝ $P^{op} \overset{def}{\equiv} (P^{op}; \leq^{op})$

where $P^{op^{op}} \equiv P$

$x \leq^{op} y \iff y \leq x$

$P + Q$ ⇝ places P, Q side-by-side ⇝ ?

$P \oplus Q$ ⇝ places P below Q ⇝ ?

$P \times^{cw} Q$ ⇝ ? (component-wise) ⇝ ?

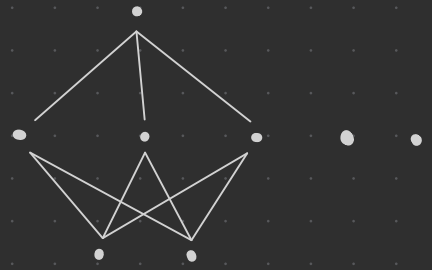
$P \otimes^{lex} Q$ ⇝ ? (lexicographic) ⇝ ?

$P \otimes^{antilex} Q$ ⇝ ? (antilexicographic) ⇝ ?

$n : \mathbb{N} \vdash$

$\overline{n} \equiv$ 

$n \equiv$ 

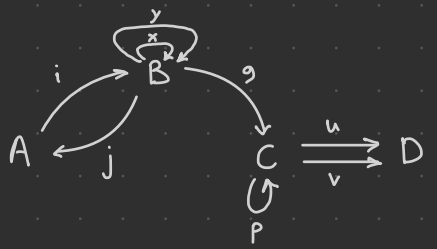
$\left(1 \oplus \overline{3} \oplus \overline{2}\right)^{op} + 2 =$ 

Categories (mathematical worlds)

lec11

2026-03-31

A category $\mathbb{C}$ consists of the following data:



$\mathbb{C} \equiv$

- a collection of "objects":
  - $\mathrm{Obj}(\mathbb{C})$ also $\mathbb{C}_0 :\equiv \{ A, B, C, \dots \}$
- a collection of "arrows":
  - $\mathrm{Arr}(\mathbb{C})$ also $\mathbb{C}_1 :\equiv \{ f, g, h, \dots, a, b, \dots \}$
- for each object, a designated "identity arrow":
  - id also 1
- for each pair of "consecutive arrows", a designated "composition arrow":
  - comp also $(\circ)$

Notation

- $\mathbb{C}(A, B) \equiv \{ \text{all arrows from } A \text{ to } B \}$

- $f : A \to B$

  or

  $A \xrightarrow{f} B$

  means "f is an arrow from the object A to the object B."

- diagrammatic $(A \xrightarrow{f} B \xrightarrow{g} C)$

- $f \,;\, g \overset{\text{Sug}}{\equiv} g \circ f$

- juxtaposition $\equiv (\circ)$

  means:

  $gf \overset{\text{means}}{\equiv} g \circ f$

What is a cat's interface? (Attempt)

$$\begin{aligned} \text{src}, \text{tgt} &: \text{Arr}(\mathbb{C}) \to \text{Obj}(\mathbb{C}) \\ \text{id} &: \text{Obj}(\mathbb{C}) \to \text{Arr}(\mathbb{C}) \\ (\circ) &: \text{Arr}(\mathbb{C}) \times \text{Arr}(\mathbb{C}) \to \text{Arr}(\mathbb{C}) \end{aligned}$$

Yikes! Like this it will need extra laws

partial

same yikes!

HW: figure out these "Yikes!" laws

SPOILER

$f : A \to B$ 

$$\left\{ \begin{array}{l} f \in \text{Arr}(\mathbb{C}) \\ \& \; A \in \text{Obj}(\mathbb{C}) \\ \& \; B \in \text{Obj}(\mathbb{C}) \\ \& \; \text{src}\, f = A \\ \& \; \text{tgt}\, f = B \end{array} \right.$$

brrr...

How about..?:

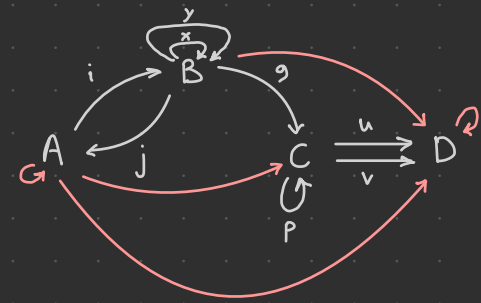
$\mathbb{C} \equiv$

$\mathbb{C}_0 \equiv$ objects $A, B, C, \ldots$

$\mathbb{C}(A, B) \equiv$ arrows $A \to B$

$id_A : \mathbb{C}(A, A)$

$comp_{A \to B \to C} : \mathbb{C}(B, C) \times \mathbb{C}(A, B) \to \mathbb{C}(A, C)$



Using dependent types:

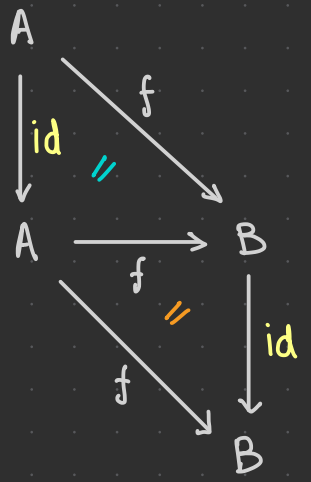
$id : (A : \mathbb{C}_0) \to \mathbb{C}(A \to A)$

$comp : (A : \mathbb{C}_0) \to (B : \mathbb{C}_0) \to (C : \mathbb{C}_0)$

$\to \mathbb{C}(B, C) \times \mathbb{C}(A, B) \to \mathbb{C}(A, C)$

Cat laws

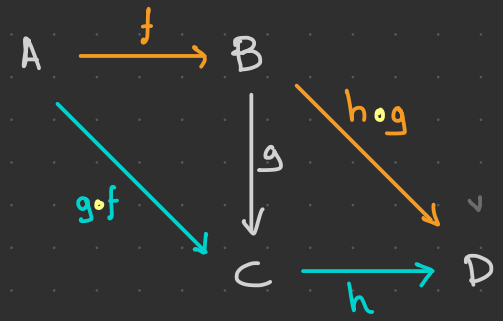
id-unit



$$f \circ id_A = f$$
$$id_B \circ f = f$$

comp-ass

T.f.d.c.:* — The following diagram commutes



$$(h \circ g)_{B \to C \to D} \circ f_{A \to B} = h \circ_{A \to C \to D} (g \circ f)_{A \to B \to C}$$

Examples

$\mathbb{1}$ $\quad$ $\mathbb{2}$ $\quad$ $\mathbb{0}$ $\quad$ $\overline{\mathbb{2}}$ $\quad$ $\mathbb{2}_{\text{Set}}$

Q. What is the freely generated cat from  ?

A. The most economical™ cat which has at least this

⚠ NOT in terms of how many objects, arrows, elements!

In terms of equations $\quad (\ fg \stackrel{?}{=} \mathrm{id}\ ;\ \ fgf \stackrel{?}{=} f\ ;\ \cdots$

$\qquad\qquad\qquad\qquad\qquad\qquad\quad gf \stackrel{?}{=} \mathrm{id}\ ;\ \ gfg \stackrel{?}{=} g\ ;\ \cdots\ )$

"Reminder": freely generated monoid

Consider the symbol s.

What is the freely generated monoid from s?

WRONG: The monoid $(\{s\}; \cdot, s)$.

↑ it only has one element, but nobody cares!

In it s is not free at all!
It is required to satisfy all of:

$$\left.\begin{aligned} s &= 1 \\ s^2 &= s \\ s^3 &= s \\ &\vdots \end{aligned}\right\}$$ none of these are monoid laws/consequences

s will refuse! (And has every right to!)

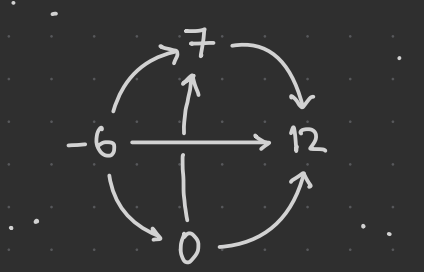
CORRECT: The monoid $(\{1, s, s^2, s^3, \ldots\}; \cdot, 1)$.

↑ some fresh object (distinct from s)

Cats of integers

$(\mathbb{Z}; \leq)$

- $\mathrm{Obj} \equiv \mathbb{Z}$
- $\mathrm{Arr}(A,B) \equiv \{ \checkmark_{A,B} \mid A \leq B \}$

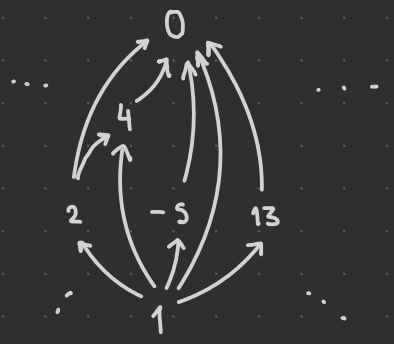


$(\mathbb{Z}; \mid)$

- $\mathrm{Obj} \equiv \mathbb{Z}$
- $\mathrm{Arr}(A,B) \equiv \{ \checkmark_{A,B} \mid A \mid B \}$

such that

divides



$\mathbb{C}[\mathcal{P}] : \mathbf{Poset} \to \mathbf{Cat}$

$\mathcal{P}$: Poset

$\mathcal{P} \equiv (P\,;\, \leq) \rightsquigarrow \mathbb{C}[\mathcal{P}] \equiv$

- $.\mathrm{Obj} :\equiv P$
- $.\mathrm{Arr}$ "$\equiv$"
- $.\mathrm{Arr}(A,B) :\equiv \{ \checkmark_{A,B} \mid A \leq_P B \} = \begin{cases} \emptyset, & \text{case } A \not\leq_P B \\ \{\checkmark\}, & \text{case } A \leq_P B \end{cases}$

Is this a cat?

We must verify:

- $\mathrm{id}_A : \mathbb{C}(A,A)$ ✓ [thanks to $\mathcal{P}.\mathrm{refl}$]
- $\mathrm{comp}_{A\to B\to C} : \mathbb{C}(B,C) \times \mathbb{C}(A,B) \to \mathbb{C}(A,C)$ ✓ [thanks to $\mathcal{P}.\mathrm{trans}$]

We get the laws for free:

all $\mathbb{C}(A,B)$ are subsingletons (Hence we have no room to break any law!)

thin cat

Of models of mathematical structures

.Obj ≡ models of such structure

Magma

Semigroup

Monoid

Group

Abel

Ring

Lattice

BoundedLattice

⋮

Algebraic structures

.Arr ≡ homomorphisms

Top .Arr ≡ continuous maps

Proset .Arr ≡ monotone, isotone order-preserving maps

Poset .Arr ≡

⋮

$\mathbb{C}[\mathcal{M}]$: Monoid → Cat

↑ Monoid

$\mathcal{M} \equiv (M; \cdot, u) \rightsquigarrow \mathbb{C}[\mathcal{M}] \equiv$

.Obj ≡ { ☺ }

.Arr ≡ M

$id_{☺} \equiv u_{\mathcal{M}} : \mathbb{C}(☺, ☺)$

$f \circ g \equiv f \cdot_{\mathcal{M}} g : \mathbb{C}(☺, ☺)$

Is this a cat?

We get for free:

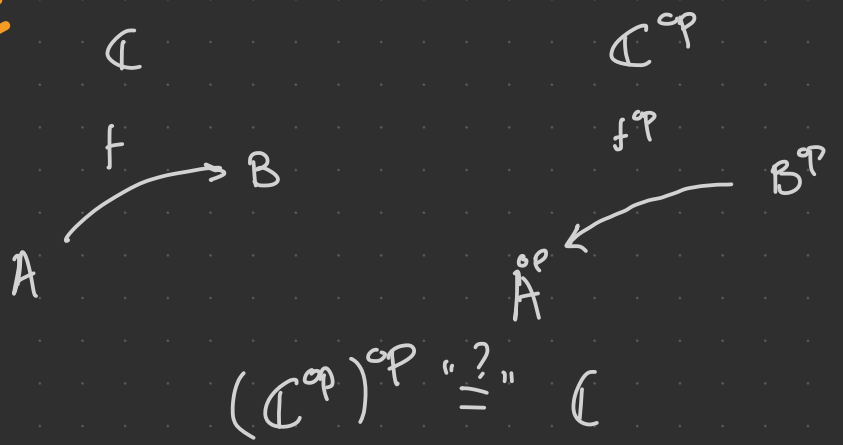
- $id_A : \mathbb{C}(A, A)$
- $comp_{A \to B \to C} : \mathbb{C}(B,C) \times \mathbb{C}(A,B) \to \mathbb{C}(A,C)$

We must verify the laws:

- 1-unit ✓ [thanks to $\mathcal{M}$.unit]
- (∘)-ass ✓ [thanks to $\mathcal{M}$.ass]

$\mathbb{C}^{op} : Cat \to Cat$

?



Non-categorial definitions

lec 12
2026-04-03

Inside Set ...:

$f : A \to B$ is injective

$$(\forall x, y \in A)[\ f x = f y \Rightarrow x = y\ ]$$

S is a singleton

$$(\exists s)[\ S = \{s\}\ ]$$

$A \subseteq B$

$$(\forall x \in A)[\ x \in B\ ]$$

f is onto B

$$(\forall b \in B)(\exists a \in A)[\ f a = b\ ]$$

⋮

none of these make sense in general (they rely on specifics that may not be available)

Categorial definitions

concepts we can define using only cat-vocabulary / interface.

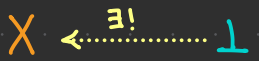
$\top$ terminal

$\overset{\text{def}}{\Longleftrightarrow}$ (categorial)

dualize →

$\bot$ coterminal
(also: initial)

$\overset{\text{def}}{\Longleftrightarrow}$ (categorial)



How does this concept manifest in ...:

$\mathbb{1}$: • has this property
$\mathbb{2}$: • → ⋆ ⋆ is terminal

$\mathbb{C}[(\mathbb{Z}; \leq)]$: has no terminal

$\mathbb{C}[(\mathbb{Z}; 1)]$: 0 is terminal
$\mathbb{C}[\mathcal{P}]$: $\top$ is terminal (if it exists)
Set : all singletons are terminal

$\mathbb{C}[\mathcal{P}]$: $\bot$ is initial (if it exists)
Set : $\emptyset$ is initial

the object T is terminal

$\iff$ for any object X, there is a unique $X \xrightarrow{!} T$

$\mathbb{C}[(\mathbb{Z};\leq)]$

integer

the ~~object~~ T is terminal

$\iff$ for any ~~object~~ integer X, ~~there is a unique $X \xrightarrow{!} T$~~ $X \mid T$

$\mathbb{C}[(\mathbb{Z};1)]$

integer

the ~~object~~ T is terminal

$\iff$ for any integer ~~object~~ X, ~~there is a unique $X \xrightarrow{!} T$~~ $X \leq T$

Set

set

the ~~object~~ T is terminal

$\iff$ for any set ~~object~~ X, there is a unique function $X \xrightarrow{!} T$

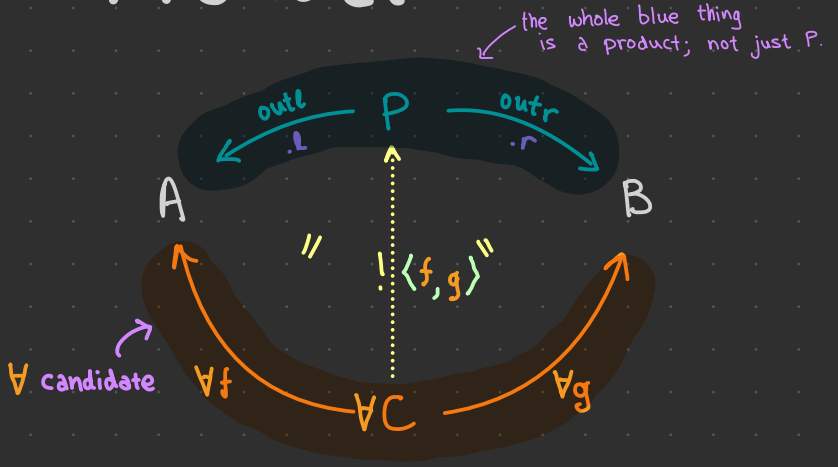
"Uniqueness" and iso

T, T' terminals $\Rightarrow T \cong T'$

$A \cong B \overset{\text{def}}{\iff} (\exists\, A \xrightarrow{f} B)[\, f \text{ iso} \,]$

$A \xrightarrow{f} B$ iso $\overset{\text{def}}{\iff} (\exists\, A \xleftarrow{f'} B)[\, f'f = 1_A \;\&\; ff' = 1_B \,]$

Product



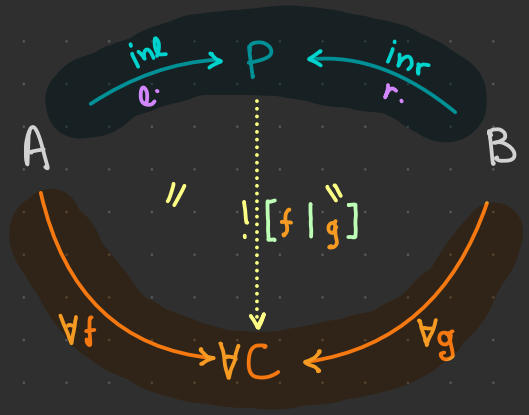
$\mathbb{C}[(\mathbb{Z};\ 1)] \rightsquigarrow$ g.c.d

$\mathbb{C}[(\wp A;\ \subseteq)] \rightsquigarrow \cap$

Set $\rightsquigarrow$ $A \times B$ $(\equiv \{\langle a,b\rangle \mid a \in A,\ b \in B\})$

$\llcorner\ h\,c = \langle f\,c,\ g\,c\rangle$

Coproduct (Sum)



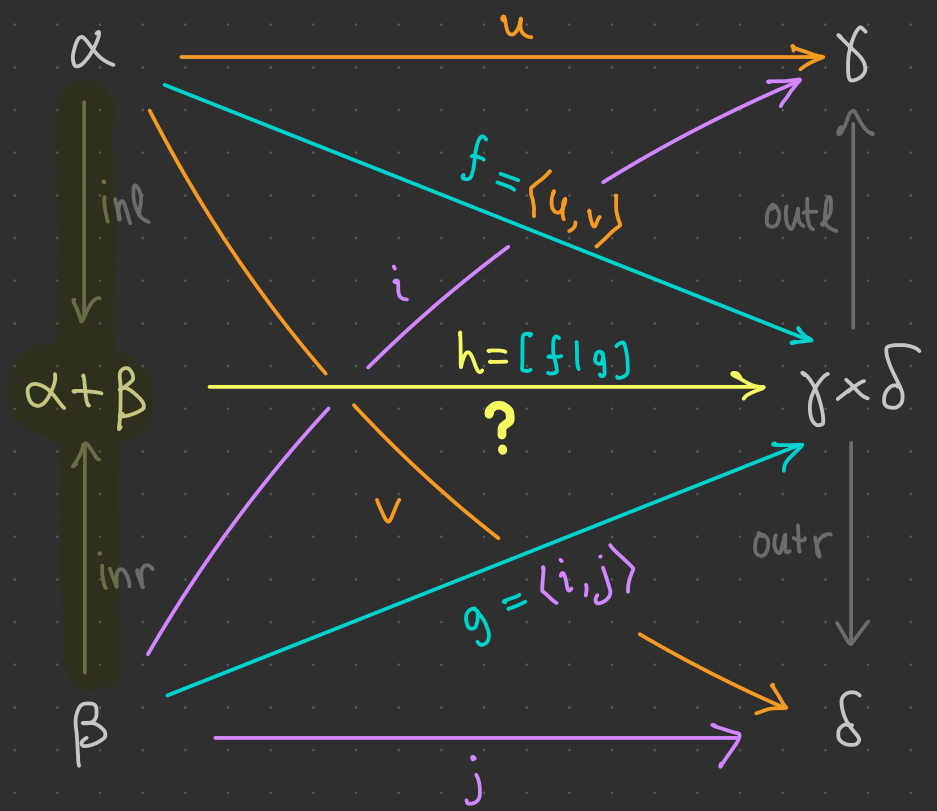
$\rightsquigarrow$ l.c.m

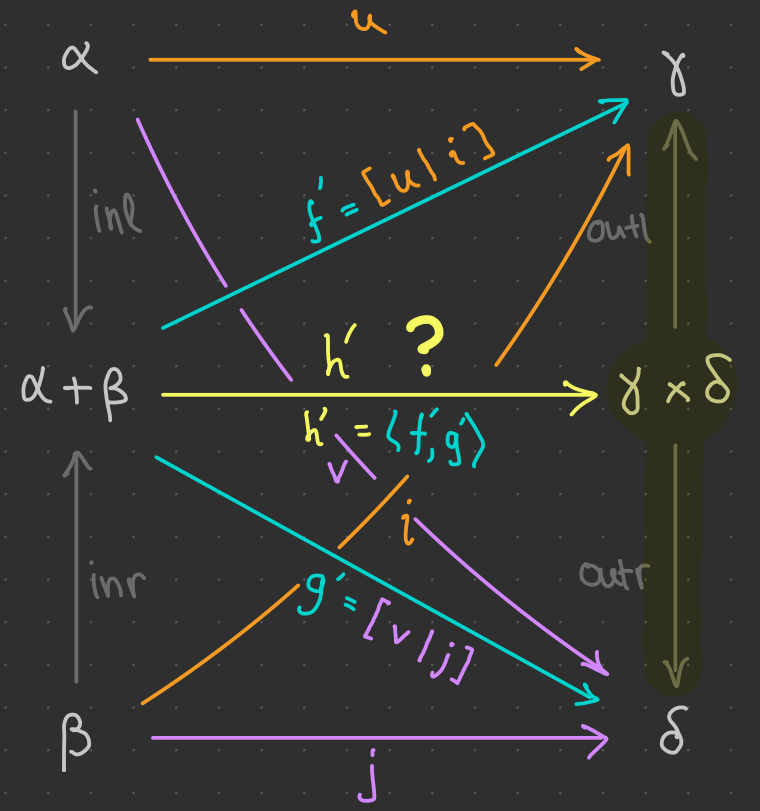
$\rightsquigarrow \cup$

$\rightsquigarrow \uplus$

Mapping out vs mapping into

$[\langle u,v\rangle \mid \langle i,j\rangle] \stackrel{?}{=} \langle [u \mid i]\ [v \mid j]\rangle$





Point-freeing injectivity

$f : A \to B$

f inj $\iff (\forall a \in A)(\forall a' \in A)\,[\, f\,a = f\,a' \Rightarrow a = a' \,]$

f surj $\iff (\forall b \in B)(\exists a \in A)\,[\, f\,a = b \,]$

$a \overset{f}{\longmapsto} b$

Looks like L-cancelation of f.
But for which op?

these two do not look dual/symmetric/etc.

↳ what is a good reason for this asymmetry?

# of vs of

We tend to "write" composition ($\circ$) and function application ($\_\,\_$) invisibly.

Let us write $@$ for function application so that

$$f \mathbin{@} x \equiv f\ x \quad (f \text{ applied to } x)$$

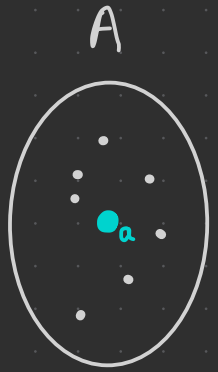
Remember the definition of $(\circ)$:

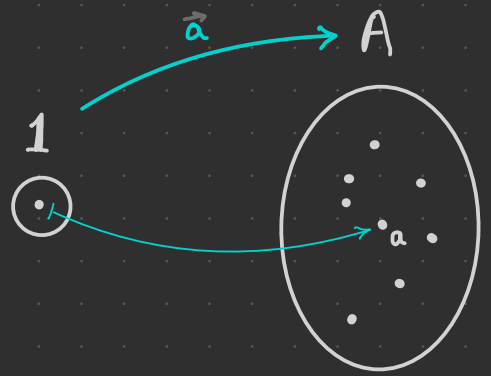
$$(f \circ g)\ x \overset{\text{def}}{\equiv} f\ (g\ x)$$

$$(f \circ g) \mathbin{\mathring{@}} \vec{x} \overset{\text{def}}{\equiv} f \mathbin{\mathring{@}} (g \mathbin{\mathring{@}} \vec{x})$$

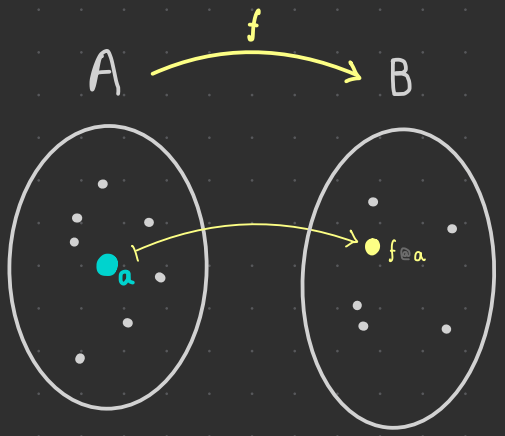
(associativity of «of»

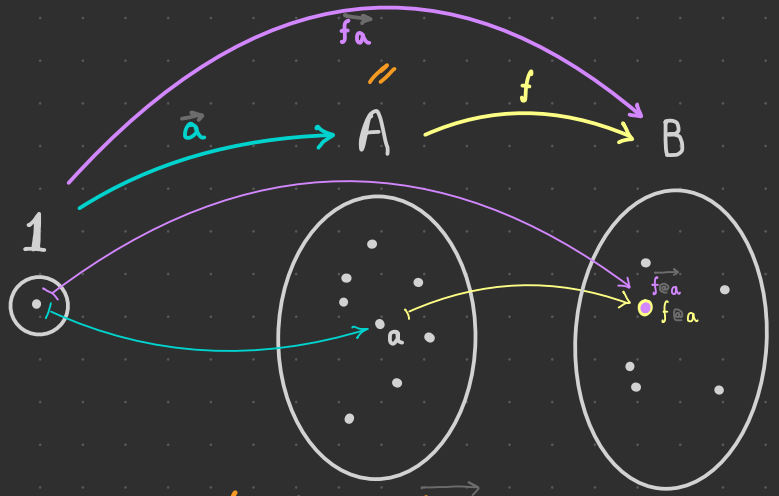
How do we say $a \in A$ and $(\forall a \in A)[\cdots]$ without points?





$(\forall a \in A)[\cdots]$ $\quad\rightsquigarrow\quad$ $(\forall 1 \xrightarrow{a} A)[\cdots]$

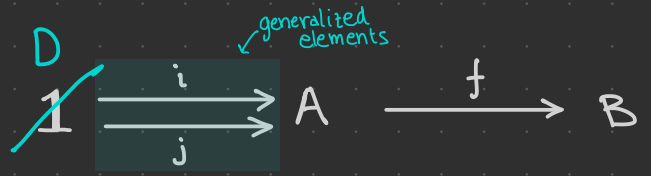




$f \circ \vec{a} = \overrightarrow{f@a}$

(...just fa)

Back to injectivity and surjectivity...



$$f \circ i = f \circ j \Rightarrow i = j$$

f mono $\overset{\text{def}}{\iff}$ f (∘)-can-L

f split mono $\overset{\text{def}}{\iff}$ f (∘)-inv-L

f epi $\overset{\text{def}}{\iff}$ f (∘)-can-R

f split epi $\overset{\text{def}}{\iff}$ f (∘)-inv-R

0. f split mono $\vdash$ f mono

Let f_L a (0)-invL of f.

Let $D \underset{j}{\overset{i}{\rightrightarrows}} A \xrightarrow{f} B$ s.t $fi = fj$

Thence $f_L(fi) = f_L(fj)$.

Thence $(f_L f)i = (f_L f)j$.

Thence $1_A i = 1_A j$.

Thence $i = j$

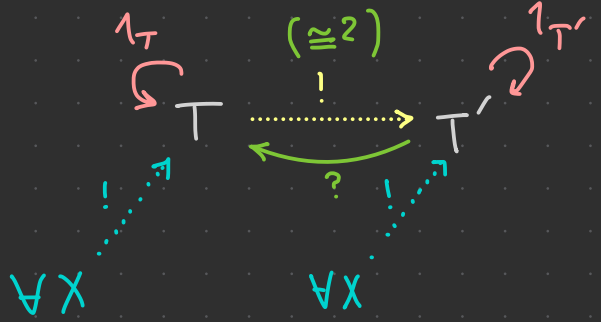
□

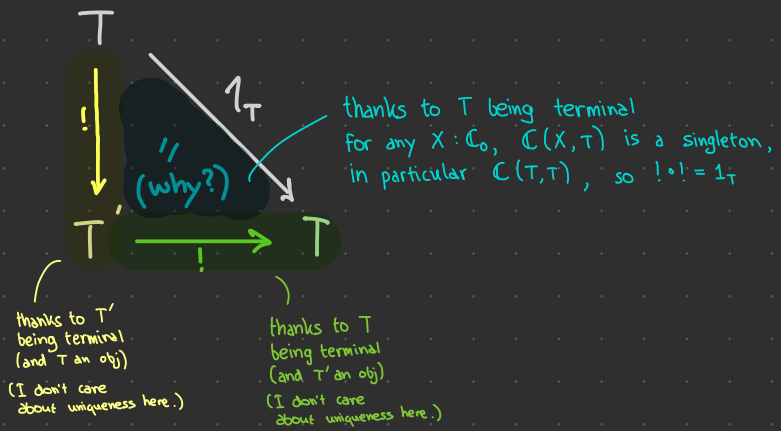
Uniqueness (very!)

lec13
2026-04-24

Uniqueness of terminals T, T' terminal $\vdash T \cong T'$

$\mathbb{C}$

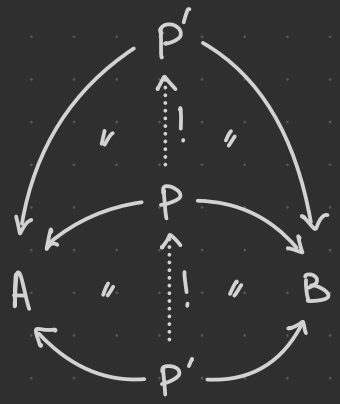




Uniqueness of initials

for free: initial means coterminal (terminal in the opposite (dual) cat).

Uniqueness of products



Finish this!

P, P' products of $A, B \vdash P \cong P'$

we can even get this for free!

Welcome spans!

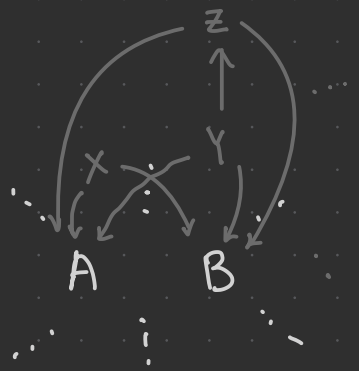
what is wrong with this?

Uniqueness of sums

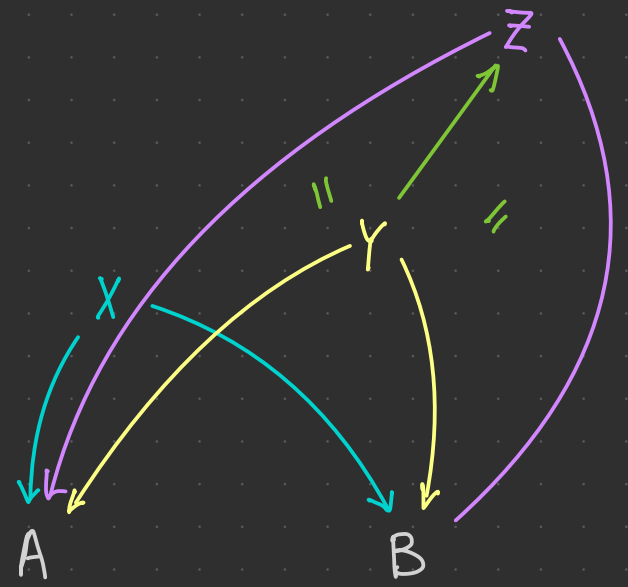
for free: sum means coproduct.

Spans

$\mathbb{C}$:



$\mathbb{C}\left[\begin{array}{ccc} & \cdot & \\ \swarrow & & \searrow \\ A & & B \end{array}\right]$:



in green : an arrow from the object $\overset{Y}{\cdot}$ to the object $\overset{Z}{\cdot}$

Remember proofs?

Objective: prove: $\Theta, \ldots, A \vee B \vdash C \,\&\, D$

Sep by cases on $A \vee B$.

Case L: -- $\ldots, A \vdash C \,\&\, D$

- Split. $[\vdash C \,\&\, D]$
  - Part L: -- $\ldots, A \vdash C$
    - ⋮
  - Part R: -- $\ldots, A \vdash D$
    - ⋮

Case R: -- $\ldots, B \vdash C \,\&\, D$

- Split. $[\vdash C \,\&\, D]$
  - Part L: -- $\ldots, B \vdash C$
    - ⋮
  - Part R: -- $\ldots, B \vdash D$
    - ⋮

Split. $[\vdash C \,\&\, D]$

Part L: -- $\ldots, A \vee B \vdash C$

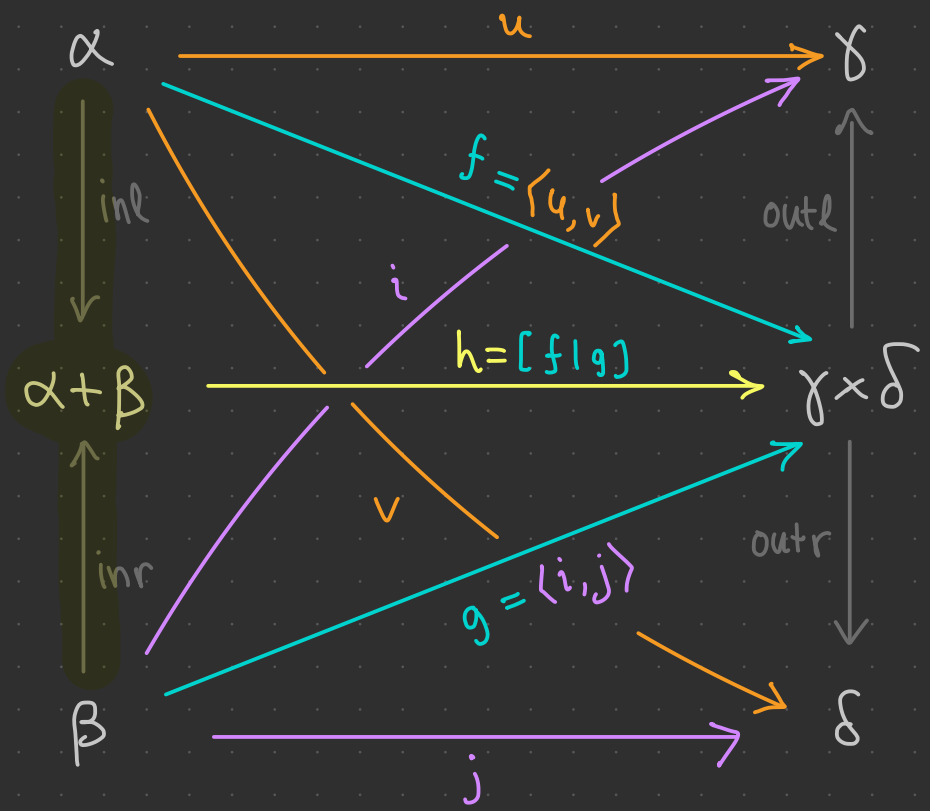
- Sep by cases on $A \vee B$.
  - Case L: -- $\ldots, A \vdash C$
    - ⋮
  - Case R: -- $\ldots, B \vdash C$
    - ⋮

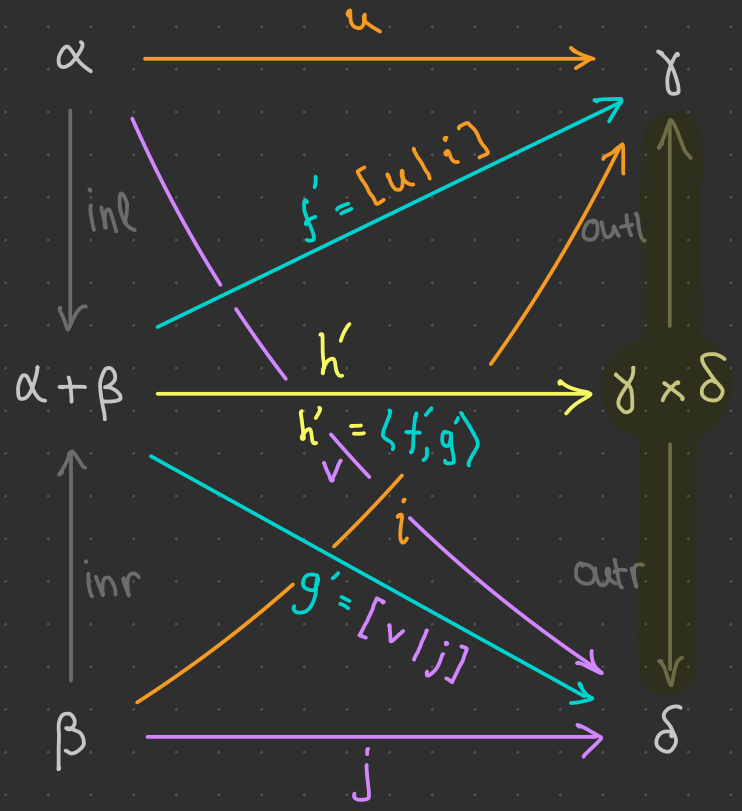
Part R: -- $\ldots, A \vee B \vdash D$

- Sep by cases on $A \vee B$.
  - Case L: -- $\ldots, A \vdash D$
    - ⋮
  - Case R: -- $\ldots, B \vdash D$
    - ⋮

Remember cats?

Objective: find an arrow from a (×) to a (+)





Remember school?

Objective: Pass the first exam on exponentiation

$$(\gamma\delta)^{\alpha+\beta}$$

$$= (\gamma\delta)^{\alpha} \cdot (\gamma\delta)^{\beta}$$

$$= \gamma^{\alpha}\delta^{\alpha}\gamma^{\beta}\delta^{\beta}$$

$$(\gamma\delta)^{\alpha+\beta}$$

$$= \gamma^{\alpha+\beta}\delta^{\alpha+\beta}$$

$$= \gamma^{\alpha}\gamma^{\beta}\delta^{\alpha}\delta^{\beta}$$

Remember Gentzen?

Objective: construct a proof-tree of $A \vee B \vdash C \,\&\, D$

$$\dfrac{\dfrac{A \vdash C \quad A \vdash D}{A \vdash C \,\&\, D}\,? \qquad \dfrac{B \vdash C \quad B \vdash D}{B \vdash C \,\&\, D}\,?}{A \vee B \vdash C \,\&\, D}\,?$$

$$\dfrac{\dfrac{A \vdash C \quad B \vdash C}{A \vee B \vdash C}\,? \qquad \dfrac{A \vdash D \quad B \vdash D}{A \vee B \vdash D}\,?}{A \vee B \vdash C \,\&\, D}\,?$$

An evergrowing hw (table)

split mono ⇄ mono

cats \ concepts	Initial	Terminal	(+)	(×)	. .	mono	epi	. .	monos split	epis split	mono & epi ⇒ iso	. . . .
𝟘	—	—	?	?		?	?		?	?	?	. . .
𝟙	•	•	?	?	. . .	?	?		?	?	?	. . .
⋮												
n	?	?	?	?		?	?		?	?	?	
(ℤ; 1)	±1	0	lcm	gcd		?	?					
(ℤ; ≤)	—	—	max	min	. .	?	?	.				
(pA; ⊆)	∅	A	∪	∩		?			.			
𝕃[P]	⊥	⊤	join	meet						.		
𝕃[M]	?	?	?	?								
Set Fun	∅	singletons	⊎	×				.				
⋮												
Semigroup	•											
Monoid	•											
Group			?								?	
Abel	•											
Ring									?			
⋮												
Field	?											
Proset				?								
⋮												
Lattice												
⋮												

An evergrowing hw (table)

We know: split mono ⇒ mono. ← so this holds in any cat.

This is the converse:

check to see if it happens that mono ⇒ split mono.

lec 14

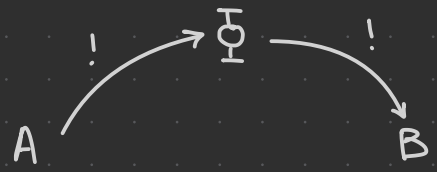
2026-04-28

cats \ concepts	Initial	Terminal	(+)	(×)	. .	mono	epi	. .	monos split	epis split	mono & epi ⇒ iso	. . .
0	–	–	?	?		?	?		?	?	?	. . .
1	•	•	?	?	. . .	?	?		?	?	?	. . .
n	?	?	?	?		?	?		?	?	?	
(ℤ; 1)	±1	0	lcm	gcd		?	?					
(ℤ; ≤)	–	–	max	min	. .	?	?	•				
(pA; ⊆)	∅	A	∪	∩		?			•			
ℂ[P]	⊥	⊤	join	meet						•		
ℂ[M]	?	?	?	?								
Set Fun	∅	singletons	⊎	×				•				
Semigroup	•				•							
Monoid	•											
Group	{e}	{e}		?	•							
Abel	•											
Ring	ℤ?	{0=1}	•	•		•			?			?
Field	?											
Proset	•				?							
Lattice	⋮											

Biterminators (aka: null, zero)

D. An object Z is a biterminator (also) iff Z is initial & Z is terminal.

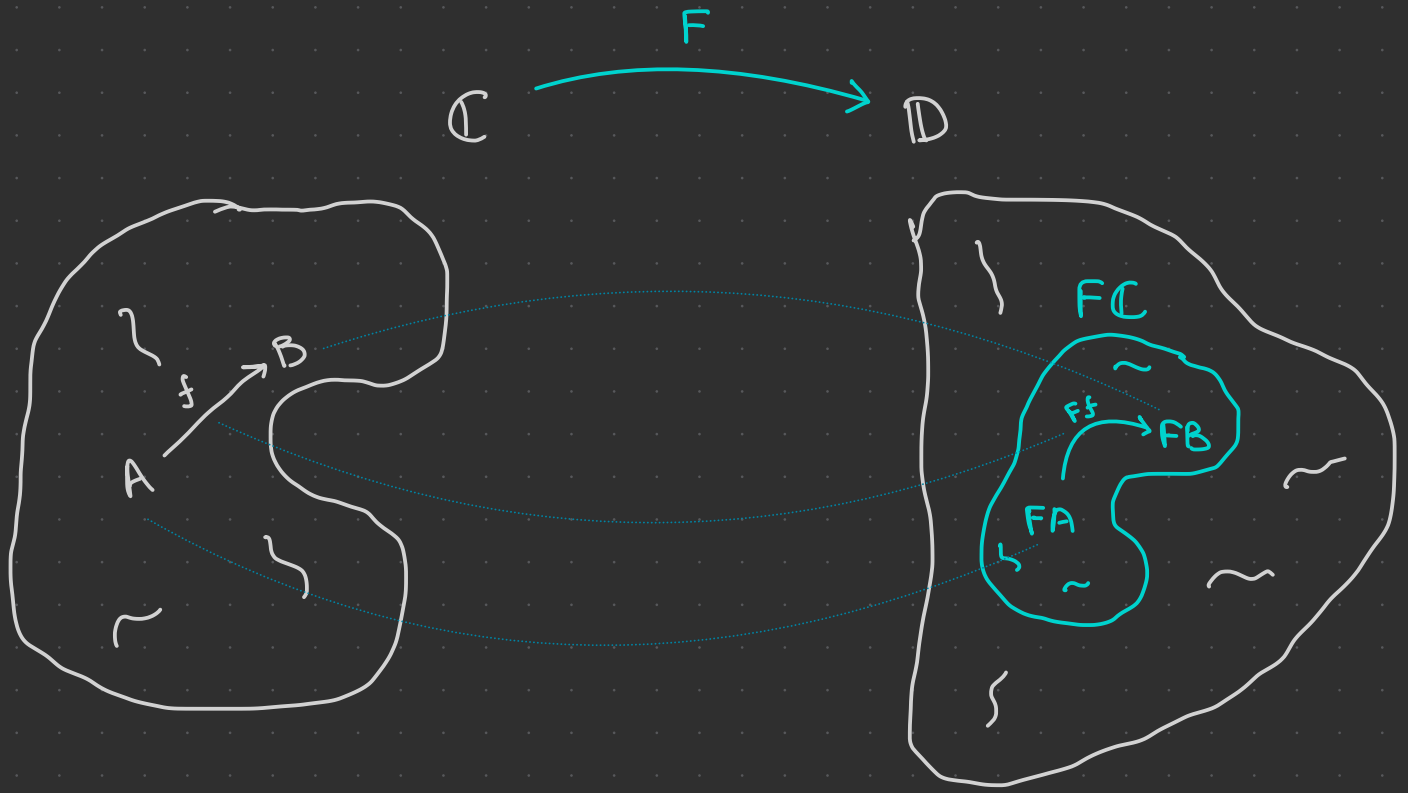
Consequence of the presence of a biterminator:



... All $\mathbb{C}(A,B)$'s are inhabited:

every obj can "look" at every obj

Functors



Functors (covariant & contravariant)

F functor

$F_0 : \mathbb{C}_0 \to \mathbb{D}_0$

$F_1 : \mathbb{C}_1 \to \mathbb{D}_1$

(covariant)

$$\frac{A \xrightarrow{f} B}{FA \xrightarrow{Ff} FB}$$

contravariant

$$\frac{A \xrightarrow{f} B}{FA \xleftarrow{Ff} FB}$$

$F_1 . (A : \mathbb{C}_0) \to (B : \mathbb{C}_0) \to \mathbb{C}(A,B) \to \mathbb{D}(A, B)$

$$\begin{cases} src \circ F_1 = F_0 \circ src \\ tgt \circ F_1 = F_0 \circ tgt \end{cases}$$

?

laws :

$$F_1 1_A =_{\mathbb{D}_1} 1_{F_0 A}$$
$$F_1 (f \circ g) =_{\mathbb{D}_1} F_1 f \circ F_1 g$$

?

Examples

powerset (image)

$$\frac{f : A \to B}{\vec{\wp} f : \vec{\wp} A \to \vec{\wp} B}$$

$\vec{\wp}_0$: $\quad \vec{\wp} X \equiv \wp X$

$\vec{\wp}_1$: $\quad \vec{\wp} f \equiv \lambda X . \{ f x \mid x \in X \}$

equiv: $(\vec{\wp} f)\, X \equiv \{ f x \mid x \in X \}$

HW: verify!

powerset (pre-image)

$$\frac{f : A \to B}{\overleftarrow{\wp} f : \overleftarrow{\wp} A \leftarrow \overleftarrow{\wp} B}$$

? ?

HW: verify!

HW: find another $\wp$-functor!

Forgetful

what do these functors forget?

examples:

Group $\xrightarrow{U}$ Monoid $\xrightarrow{U}$ Semigroup $\xrightarrow{U}$ Magma $\xrightarrow{U}$ Set **HW:** verify!

Hask (not really)

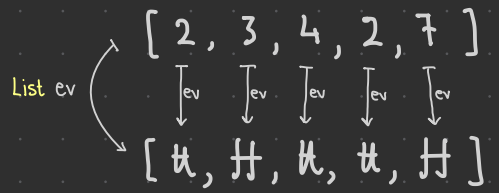
↱ for Haskellers: why not?

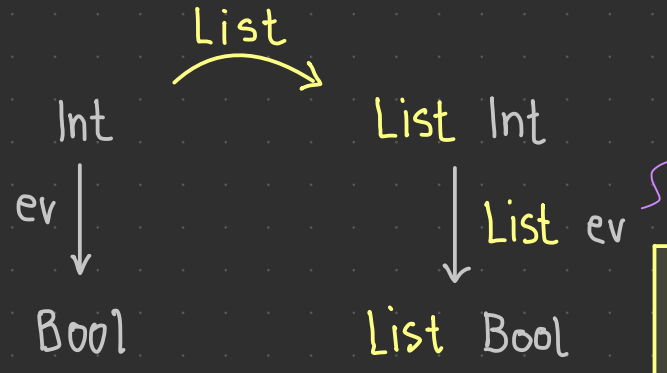
- Obj := types
- Arr := programs (functions) $\alpha \to \beta$

List/map

Hence an endofunctor (wow!)

List : Hask ⟶ Hask





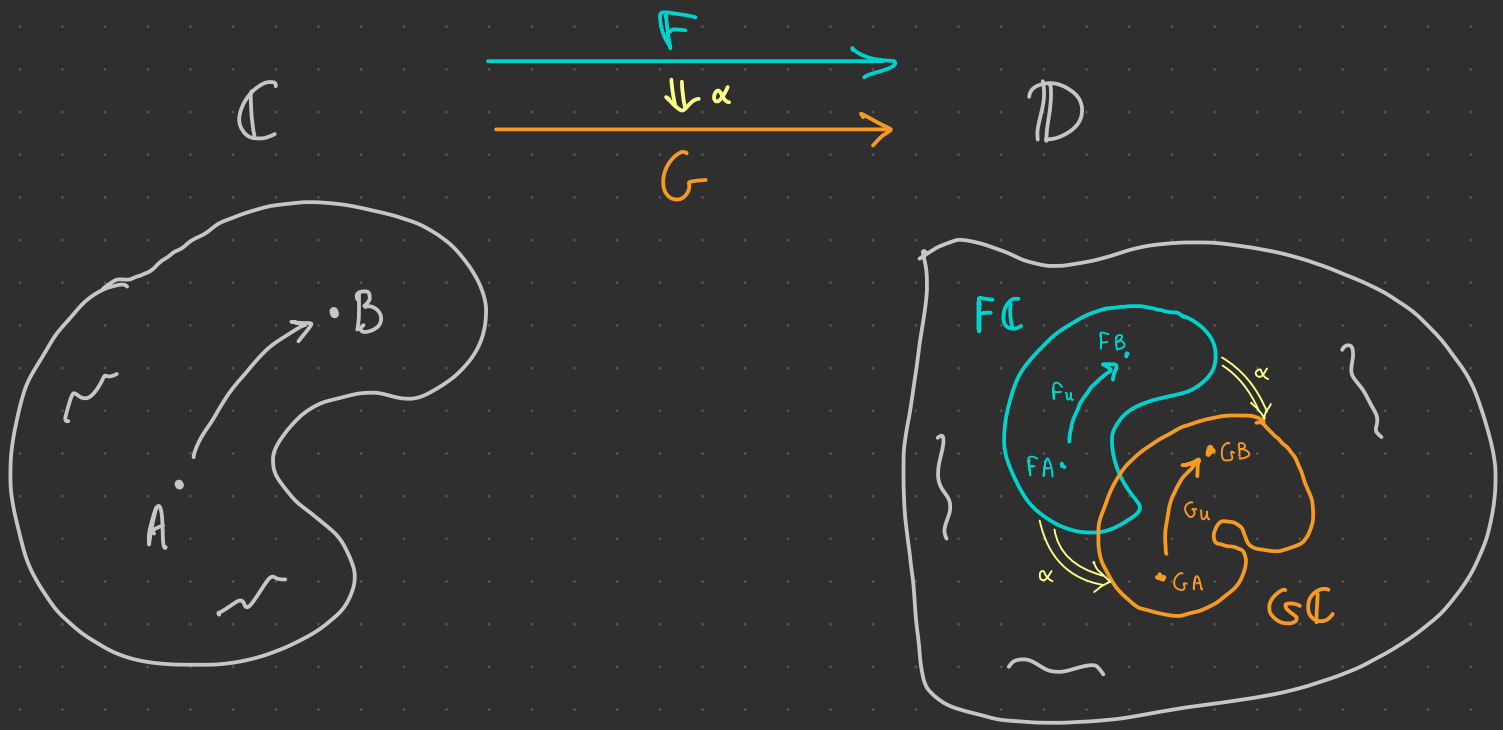
(Haskellers: fmap ev)

laws:

$$\text{map}\ \text{id}_{\text{Int}} = \text{id}_{\text{List Int}}$$

$$\text{map}\ (f \circ g) = \text{map}\ f \circ \text{map}\ g$$

Natural transformations

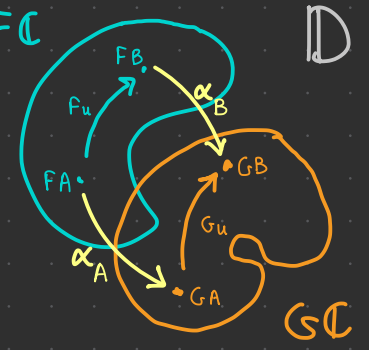


$F, G : \mathbb{C} \longrightarrow \mathbb{D}$

$\alpha : F \Longrightarrow G$

What should this α be?

A family of arrows in $\mathbb{D}$: ... indexed by ...?



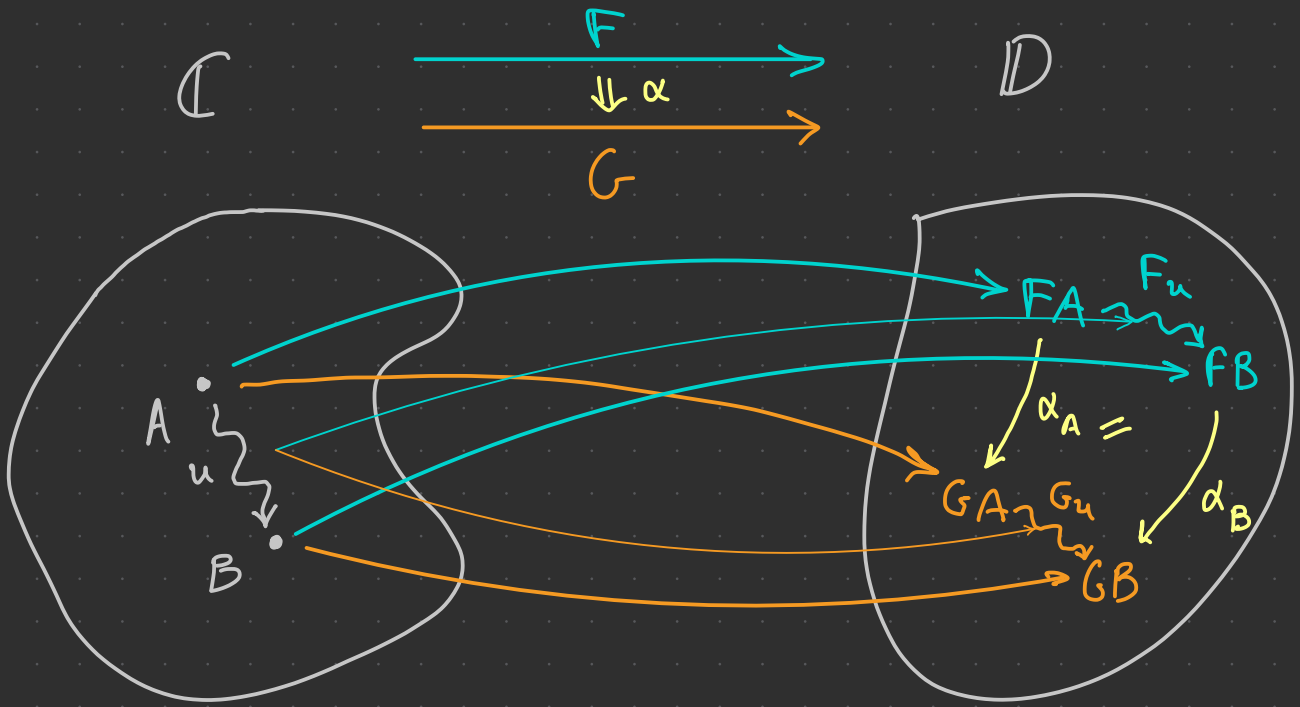
α must provide a way to transform blue A to orange A, blue B to orange B, etc.

For each object C of $\mathbb{C}$, we have a transformation from FC to GC.

$\alpha \equiv (\alpha_A : FA \to GA)_{A \in \mathbb{C}_0}$: $\mathbb{C}_0$-indexed family of arrows in $\mathbb{D}$

$\alpha : (A : \mathbb{C}_0) \to \mathbb{D}(FA, GA)$

law: $\left(\forall \begin{matrix} A \\ u \downarrow \\ B \end{matrix} \text{ in } \mathbb{C}\right)$ $\begin{matrix} FA & \xrightarrow{\alpha_A} & GA \\ Fu \downarrow & & \downarrow Gu \text{ in } \mathbb{D} \\ FB & \xrightarrow[\alpha_B]{} & GB \end{matrix}$


C
F
α
G
D
A
u
B
FA
Fu
FB
αA
GA
Gu
GB
αB

Back to logic: implication (⊃)

$$\frac{\Gamma, A \text{ true} \vdash B \text{ true}}{\Gamma \vdash A \supset B \text{ true}} \supset\text{-I}$$

$$\frac{\Gamma, x : \alpha \vdash b : \beta}{\Gamma \vdash \lambda x.b : \alpha \to \beta}$$

$$\frac{A \supset B \text{ true} \qquad A \text{ true}}{B \text{ true}} \supset\text{-E}$$

aka "Modus Ponens"

$$\frac{f : \alpha \to \beta \qquad a : \alpha}{f\ a : \beta}$$

curry

$$\frac{A \supset B \text{ true}}{\Gamma, A \text{ true} \vdash B \text{ true}}$$

Order-theoretic view on $(\vdash)$

$A \leq B$ means A true $\vdash$ B true

has bottom: $\bot$

$$\frac{}{\bot \leq X}$$

has top: $\top$

$$\frac{}{X \leq \top}$$

has joins: $\vee$

$$\frac{}{A \leq A \vee B} \qquad \frac{}{B \leq A \vee B}$$

upper bound

$$\frac{A \leq C \qquad B \leq C}{A \vee B \leq C}$$

best

has meets: $\wedge$

$$\frac{}{A \wedge B \leq A} \qquad \frac{}{A \wedge B \leq A}$$

lower bound

$$\frac{C \leq A \qquad C \leq B}{C \leq A \wedge B}$$

best

what about $\supset$?

Exponentials

$$\frac{}{B^A \wedge A \leq A}$$

combined (met) with A together (conjunction) — is below A proves

$$\frac{C \wedge A \leq A}{C \leq B^A}$$

best

other notations: $A \Rightarrow B$, $A \to B$, $A \supset B$

A lattice that guarantees exponentials is called a Heyting Algebra.

HA (Heyting Algebras)

Define $\neg A \overset{\text{def}}{=} A \Rightarrow \bot$

Θ. $\neg A$ is the best inconsistent with A. (?)

Θ. Any HA is distributive.

Lattices: how to prove...

lec15
2026-05-05

Example: $d \vee (a \wedge b) \leq (d \vee a) \wedge (d \vee b)$

this requires thinking/inspiration

$$\dfrac{\dfrac{d \leq d \vee a \quad ? \qquad a \wedge b \leq d \vee a \quad ?}{d \vee (a \wedge b) \leq d \vee a}\,(\vee)\text{-best} \qquad \dfrac{\dfrac{}{d \leq d \vee b}\,(\vee)\text{-ub}_L \qquad \dfrac{\dfrac{}{a \wedge b \leq b}\,(\wedge)\text{-lb}_R \qquad \dfrac{}{b \leq d \vee b}\,(\vee)\text{-ub}_R}{a \wedge b \leq d \vee b}\,(\leq)\text{-tr}}{d \vee (a \wedge b) \leq d \vee b}\,(\vee)\text{-best}}{d \vee (a \wedge b) \leq (d \vee a) \wedge (d \vee b)}\,(\wedge)\text{-best}$$

A familiar alternative:

... from...?

$$\dfrac{\vdots \quad \bullet \leq \bullet \qquad \vdots \quad \bullet \leq \bullet}{d \vee (a \wedge b) \leq (d \vee a) \wedge (d \vee b)}\,(\vee)\text{-best}$$

Distributivity & Modularity

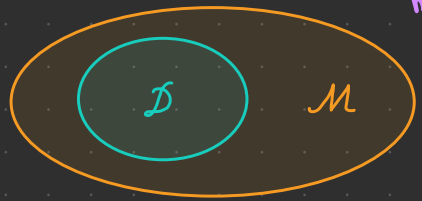
Every lattice is "half distributive" & "half modular"

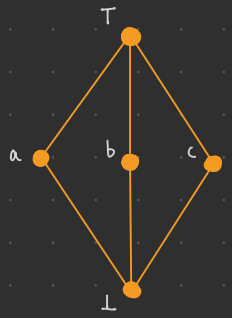
- θ. $d \wedge (a \vee b) \geqslant (d \wedge a) \vee (d \wedge b)$ [safe-distr$^{\perp}$ $(\wedge),(\vee)$]
- θ. $d \vee (a \wedge b) \leqslant (d \vee a) \wedge (d \vee b)$ [safe-distr$^{\perp}$ $(\vee),(\wedge)$]
- θ. $a \leq b \Rightarrow a \vee (x \wedge b) \leq (a \vee x) \wedge b$ [safe-modular]
- θ. (dual?)

Also:

- θ. $\underbrace{(a \wedge b) \vee (b \wedge c) \vee (c \wedge a)}_{\text{join of meets}} \leq \underbrace{(a \vee b) \wedge (b \vee c) \wedge (c \vee a)}_{\text{meet of joins}}$

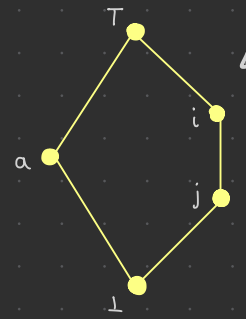
(dual?)



Θ 

M_3

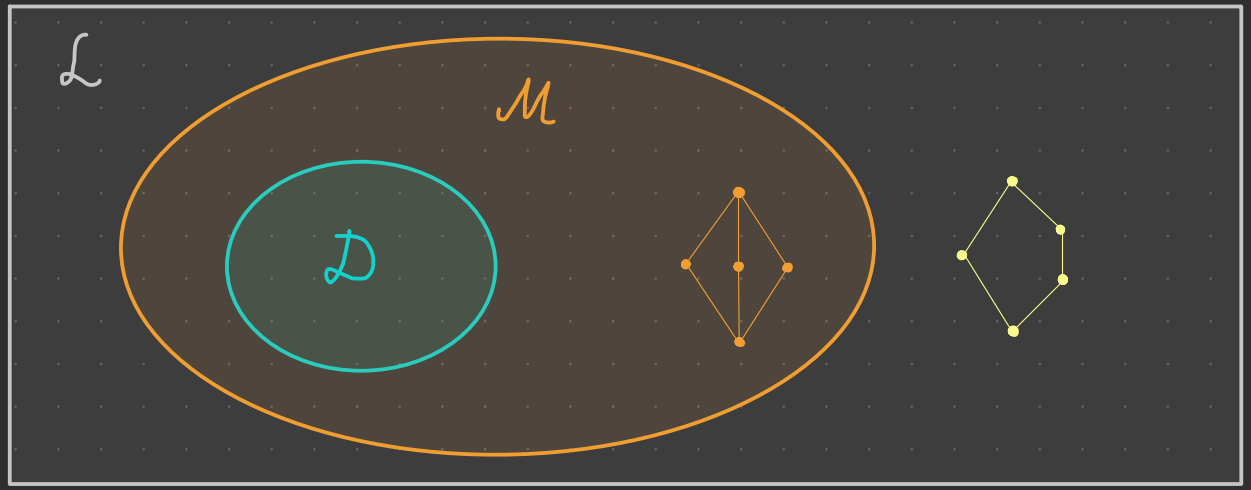
← modular
non-distributive

Θ. 

N_5

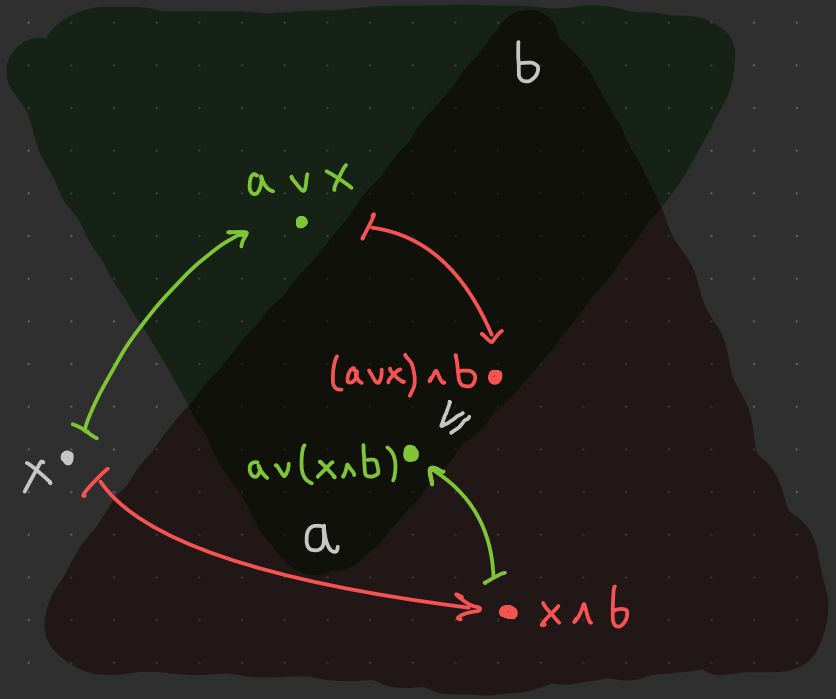
← non-modular
(& non-distributive)

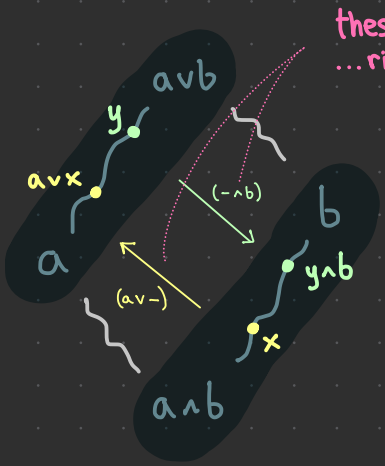
Θ. distr $\Rightarrow$ modular

Θ^2. 

Modularity

$$[a,b] \overset{\text{def}}{=} \{ x \mid a \leq x \leq b \}$$





$(a \vee -)$ $(- \wedge b)$

Σ-actions

Set / Semigroup / Monoid / Group

$$S \xrightarrow{\rho} \begin{matrix} X \longrightarrow X \\ \mathrm{End}(X) \\ \mathrm{Aut}(X) \end{matrix}$$

ρ Σ-homo

$$\begin{cases} \rho\,(r \cdot s) = \rho\, r \circ \rho\, s \\ \rho\, 1 = \mathrm{id}_X \\ \rho\,(r^{-1}) = (\rho\, r)^{-1} \end{cases}$$

X of $\mathbb{C}$

no further conditions imposed by $\mathrm{cod}(\rho)$ since X has no further known structure

$$S : \Sigma, \quad X : \mathbb{C}_0, \quad \rho : S \xrightarrow{\Sigma\text{-hom}} \mathrm{End}(X)$$

We call $(X ; \rho)$ a Σ S-action

We say that S Σ-acts on the object X via ρ.

What about Rings?

lec16
2026-05-08

Meet the best example of a ring:

Let $\mathcal{A}, \mathcal{B}$: Group, $\mathcal{B}$ abel, $\varphi, \psi : \mathcal{A} \longrightarrow \mathcal{B}$

Define on $\mathrm{Hom}(\mathcal{A}, \mathcal{B})$ the following binary operation $(+)$:

$$\varphi + \psi \overset{\text{def}}{\equiv} \lambda a.\ \varphi a +_{\mathcal{B}} \psi a$$

Prove: $(\mathrm{Hom}(\mathcal{A}, \mathcal{B}); +, ?, ?)$ is an abelian group.

$$(\varphi + \psi)(a + a')$$

$$\equiv (\varphi+\psi)\, a + (\varphi+\psi)\, a'$$

⋮

$$R \xrightarrow{\rho} \mathrm{End}(X) \quad (X \to X) \qquad \xrightarrow{\text{uncurry}} \qquad R \times X \xrightarrow{(\text{juxt})} X$$

$$ra \overset{\text{def}}{\equiv} (\rho\, r)\, a$$

(juxtaposition)

ρ $\mathbb{Z}$-homo:

$$\begin{aligned}
\rho\,(r \cdot s) &= \rho\, r \circ \rho\, s \\
\rho\, 1 &= \mathrm{id}_X \\
\rho\,(r+s) &= \rho\, r + \rho\, s \\
\rho\, 0 &= 0 \\
\rho\,(-r) &= -(\rho\, r)
\end{aligned}$$

$$\begin{aligned}
(\rho\,(r\cdot s))\,a &= (\rho\, r \circ \rho\, s)\,a \\
(r\cdot s)a &= (\rho\, r)((\rho\, s)\,a) \\
&= (\rho\, r)(sa) \\
&= r(sa)
\end{aligned}$$

$$(r\ s)a = r(sa)$$

$$\begin{aligned}
1a &= a \\
(r+s)a &= ra + sa \\
0a &= 0 \\
(-r)a &= -(ra)
\end{aligned}$$

must be a Group-homo

For each $r \in R$, $\rho\, r$ must be an arrow in $\mathbb{C}$

$\mathbb{C}$:

$$\begin{aligned}
(\rho\, r)(a+b) &= (\rho\, r)\,a + (\rho\, r)\,b \\
(\rho\, r)\,0 &= 0 \\
(\rho\, r)(-a) &= -((\rho\, r)\,a)
\end{aligned}$$

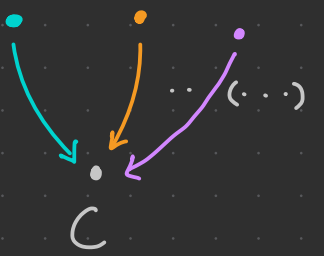
$$\begin{aligned}
r(a+b) &= ra + rb \\
r0 &= 0 \\
r(-a) &= -(ra)
\end{aligned}$$

More constructions

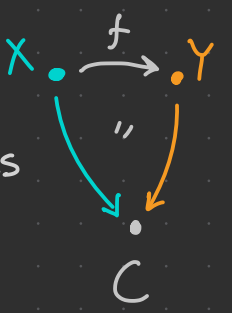
Over categories ← dualize to obtain under

$\mathbb{C} : \mathrm{Cat}$, $C : \mathbb{C}_0 \longmapsto \mathbb{C}/C$ or $\mathbb{C}\downarrow C$ or $\mathbb{C}\downarrow C : \mathrm{Cat}$

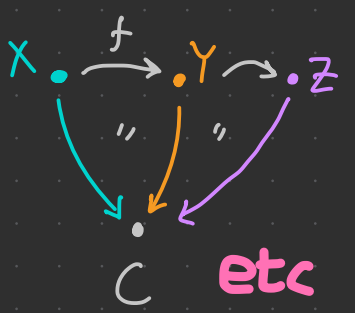
$\mathrm{Obj} := \bigcup_{X \in \mathbb{C}_0} \mathrm{Arr}(X, C)$



$\mathrm{Arr}\left(\begin{smallmatrix} X \\ \downarrow \\ C \end{smallmatrix}, \begin{smallmatrix} Y \\ \downarrow \\ C \end{smallmatrix}\right)$:= the commutative triangles



verify:

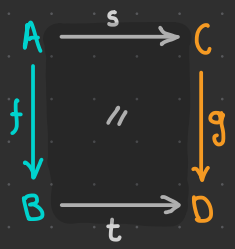


etc

Arrow categories

. Obj : $\mathbb{C}_1$

. Arr $\left(\begin{smallmatrix} A \\ \downarrow f \\ B \end{smallmatrix}, \begin{smallmatrix} C \\ \downarrow g \\ D \end{smallmatrix} \right)$:= the commutative squares



(verify!)

Maybe

abbrevs. M, J, N

Maybe : Type → Type

Just$_\alpha$: α → Maybe α

Nothing$_\alpha$: Maybe α

$$\frac{\alpha : \text{Type}}{M\,\alpha : \text{Type}}\ ? \qquad \frac{a : \alpha}{J\,a : M\,\alpha}\ ? \qquad \frac{}{N : M\,\alpha}\ ?$$

Example inhabitants of Maybe Nat

J 3, J 0, N, J 42, ..

We have defined Maybe as Type.Obj → Type.Obj

Can we also define it as Type.Arr → Type.Arr ?

$$\frac{f : \alpha \to \beta}{M\,f : M\,\alpha \to M\,\beta}$$

By pattern-matching:

verify the functor laws!

$(M\,f)\ (J\,a) \equiv J\,(f\,a)$

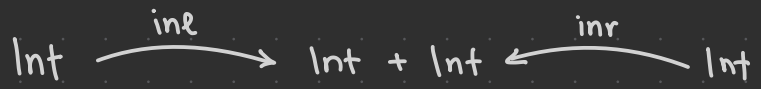
$(M\,f)\ N \equiv N$

Implementing Maybe

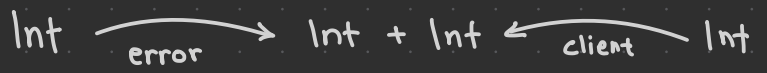
One could use $M\ \alpha \overset{\text{sug}}{\equiv} \alpha + 1$.

But names matter!

Example



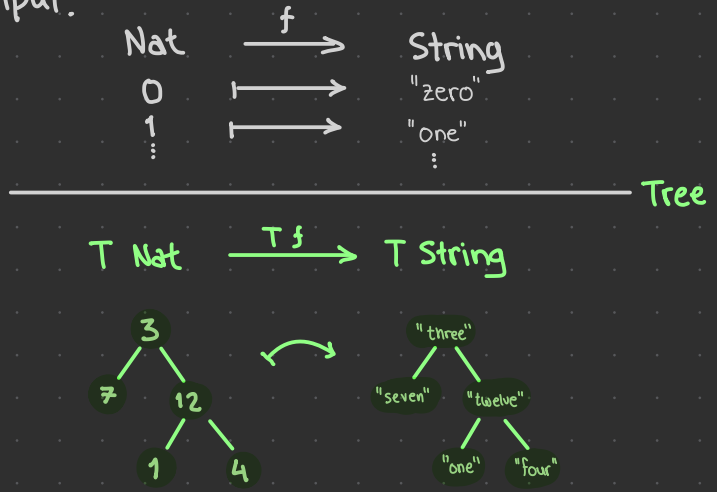
vs



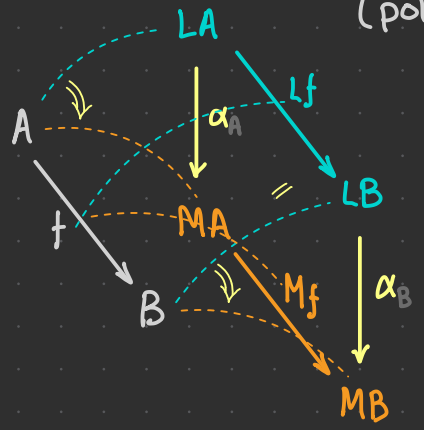
Functors vs Natural transformations

lec17
2026-05-15

Functoriality guarantees that "lifted" functions will not "change" (figure of speech) the form / shape / package of their input.



Naturality guarantees that α will be defined in a "natural" way, uniformly for all types A. (polymorphic function)



Propositions - Types

F/I/E/β/η

$(\times), (+), 0, 1, (\to), \Sigma, \Pi, \mathrm{Id}$

$(\wedge), (\vee), \bot, \top, (\supset), \exists, \forall, =$

$i : I \vdash D(i) : \mathrm{Type}$ $\quad (D : I \to \mathrm{Type})$

$a : A,\ b : B(a),\ c : C(a,b) \vdash \ldots$ ← dependent context

$a : \mathrm{Int} \vdash \text{is-even}\ a : \mathrm{Type}$

$a : A,\ b : B \vdash \mathrm{Id}\ a\ b : \mathrm{Type}$ (B ~~struck~~, A ?)

$A : \mathrm{Type},\ a : A,\ b : A \vdash \mathrm{Id}_A\ a\ b : \mathrm{Type}$

also: $a =_A b$

(the type of identifications between the elements $a, b : A$)

Examples

is-even n ≡ 2 ∣ n

divides : Int → Int → ~~Prop~~ Type

divides a b ≡ (∃k · Int)[a·k = b]

a ∣ b

$$\sum_{k:\mathrm{Int}} \left(\mathrm{Id}_{\mathrm{Int}}\ (a\ k)\ b\right)$$

$$\sum_{k:\mathrm{Int}} \left(a \cdot k =_{\mathrm{Int}} b\right)$$

Int × oops! I need to refer to the (value of) the previous component!

$$(k : \mathrm{Int}) \times \left(a \cdot k =_{\mathrm{Int}} b\right)$$

Example: what is an element of divides 3 12 ?

a pair ⟨w, p⟩ of an integer w (witness) and a proof that the witness is valid: 3 · w = 12 .

$$\text{trans-}| \equiv \prod_{a:\text{Int}} \prod_{b:\text{Int}} \prod_{c:\text{Int}} \left((a \mid b \times b \mid c) \to a \mid c \right)$$

what is wrong with this?

$$? : (a : \text{Int}) \to (b : \text{Int}) \to (c : \text{Int}) \to (a \mid b \times b \mid c) \to (a \mid c)$$

$$f\ a\ b\ c\ h \equiv \langle h.l.l \cdot h.r.l\ ,\ ?? \rangle$$

$$\begin{aligned} a &: \text{Int} \\ b &: \text{Int} \\ c &: \text{Int} \\ h &: a \mid b \times b \mid c \end{aligned}$$

$$f\ a\ b\ c\ \langle hab, hbc \rangle \equiv$$

$$\begin{aligned} &\vdots \\ hab &: a \mid b \\ hbc &: b \mid c \end{aligned}$$

witness: $wab : \text{Int}$

equation: $eab : a \cdot wab = b$

$wbc : \text{Int}$

$ebc : b \cdot wbc = c$

$$f\ a\ b\ c\ \langle \langle wab, eab \rangle, \langle wbc, ebc \rangle \rangle \equiv \langle wab \cdot wac$$

Names matter!

(even when we have unique up to unique isomorphism)

compare: `h.l.l` and `h.l.r` with `hab.witness` and `hab.eq`.

When you need a (co)product, think of which one you prefer (including names for the (co)projections).

Defining groups

$$G \equiv (G\ ;\ op,\ e,\ i)$$

$$\text{Group} \equiv (G : \text{Type}) \times (G \times G \to G) \times G \times (G \to G)$$

But what about the group laws?

$$(G : \text{Type}) \times (op : G \times G \to G) \times (e : G) \times (i : G \to G)$$
$$\times\ \text{assoc}\ op$$
$$\times\ \text{unit}\ op\ e$$
$$\times\ \text{inv}\ op\ e\ i$$

(... or something like that)

choose nice names for the projections!

G.carrier, G.op, G.id, G.inv, G.laws.assoc
G.assoc
G.assoc-op
⋮ etc.

$$\text{assoc}\ A\ op \equiv \prod_{a:A}\ \prod_{b:A}\ \prod_{c:A}\ op\ (op\ a\ b)\ c = op\ a\ (op\ b\ c)$$

lec18

2026-05-19

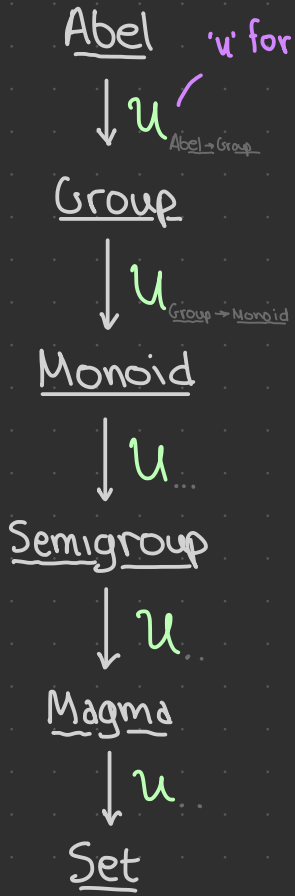
(Lean)

lec19

2026-05-22

(Lean)

Forgetful functor



$U : \underline{\text{Group}} \to \underline{\text{Set}}$

U_G forgets the group structure

$(U_G \equiv |G|)$

U_φ forgets that φ is respectful

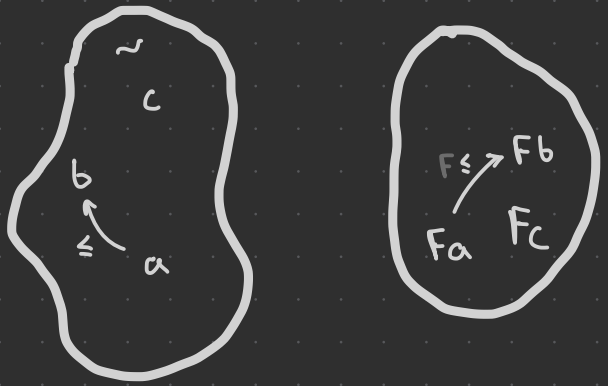
$(U_\varphi \equiv |\varphi|$

Can we find some-kind of dual/inverse/whatever functors going the other way?

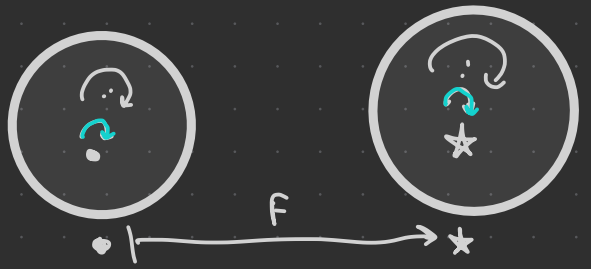
"just"

lec20
2026-05-26

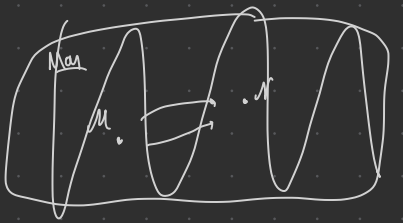
$\mathbb{C}[P] \xrightarrow{F} \mathbb{C}[Q]$



$\mathbb{C}[\mathcal{M}] \xrightarrow{F} \mathbb{C}[\mathcal{N}]$



$$F 1_\bullet = 1_{F\bullet} \equiv 1_\star$$
$$F(a \circ b) = Fa \circ Fb$$
$$\| \qquad\qquad \|$$
$$F(a \cdot_{\mathcal{M}} b) = Fa \cdot_{\mathcal{N}} Fb$$



Groupoid

↑ a cat where all arrows are iso

Group is "just" a groupoid with only one obj.

Group ⇝ Groupoid

Monoid ⇝ Monoidoid
"
Cat

Group Object

$\mathbb{C}$

Set
Grp
Top

$\mathrm{Groups}(\mathbb{C})$

$\mathrm{Groups}(\mathrm{Set}) \simeq \mathrm{Grp}$

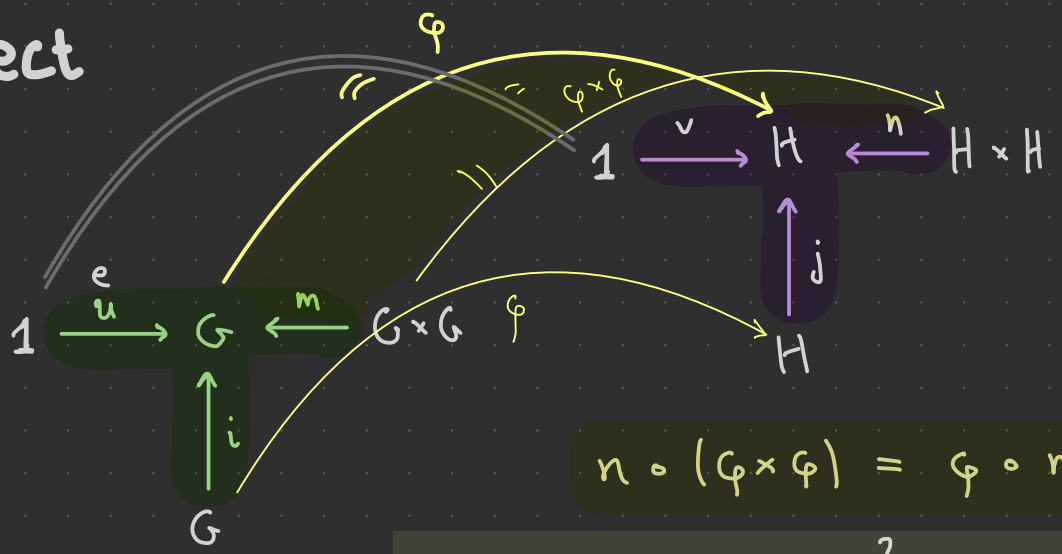
Grp Abel

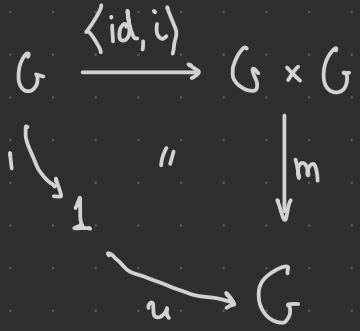
$$m \circ (\varphi \times \varphi) = \varphi \circ m$$

$$(m \circ (\varphi \times \varphi))(a,b) \overset{?}{=} (\varphi \circ m)(a,b)$$

$$\varphi a \cdot_H \varphi b = \varphi(a \cdot_G b)$$

Group Object





$$n \circ (\varphi \times \varphi) = \varphi \circ m$$

$$\begin{array}{ccc} (n \circ (\varphi \times \varphi))\,(a,b) & \overset{?}{=} & (\varphi \circ m)\,(a,b) \\ \| & & \| \\ n\,(\,(\varphi \times \varphi)\,(a,b)\,) & & \varphi\,(\,m\,(a,b)\,) \\ \| & & \| \\ n\,(\varphi a, \varphi b) & & \varphi\,(a \cdot_G b) \\ \| & & \\ \varphi a \cdot_H \varphi b & & \end{array}$$

. . . . ?

Groups($\mathbb{C}$)

Groups(Set) $\rightsquigarrow$ Groups

Groups(Group) $\rightsquigarrow$ Abelian groups

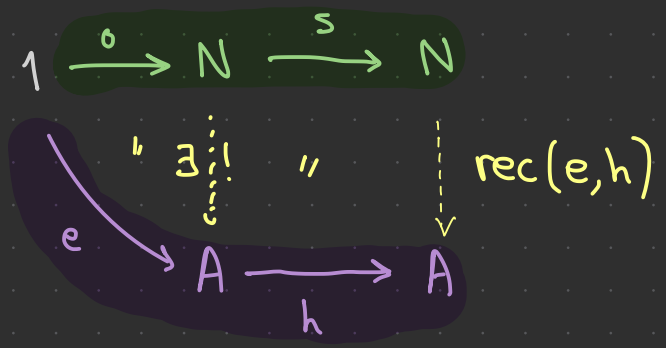
Groups(Abel) $\rightsquigarrow$ Abelian groups

Groups(Top) $\rightsquigarrow$ Topological groups

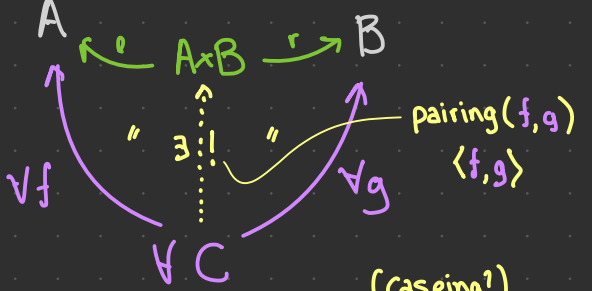
$\vdots$

NNO

should we demand more?



$$\frac{e : 1 \to A \qquad h : A \to A}{\mathrm{rec}(e,h) : N \to A}$$



(caseing?)

+ : copairing(f,g)

[f | g]

...etc

$f : D \to A \times B$

$f\ d = \langle \ldots, \ldots \rangle$

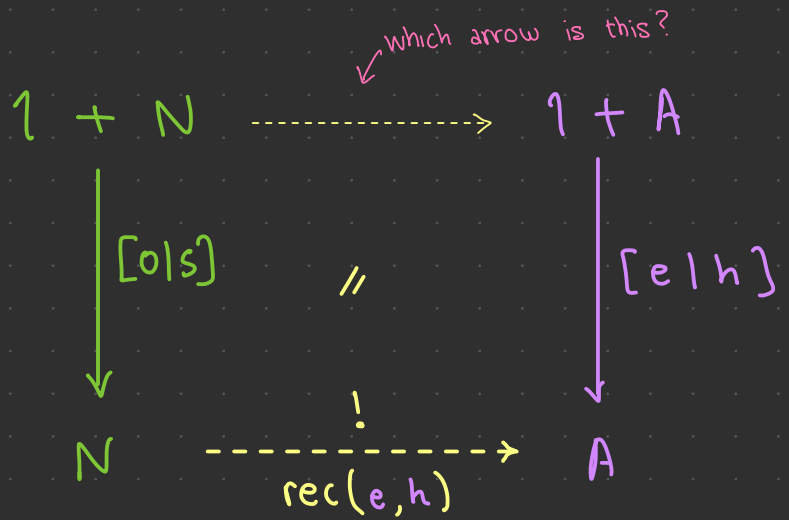
$f : A + B \to C$

$f\ (\mathrm{inl} \cdot a) \equiv \ldots$

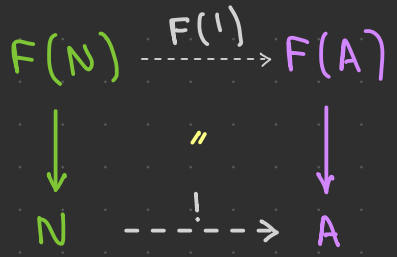
$f\ (\mathrm{inr} \cdot b) \equiv \ldots$

$\mathrm{Hom}(D, A \times B) \cong \mathrm{Hom}(D,A) \times \mathrm{Hom}(D,B)$

$\mathrm{Hom}(A+B, C) \cong \mathrm{Hom}(A,C) \times \mathrm{Hom}(B,C)$



$F(X) := 1 + X$



for groups:

$$F(X) = 1 + X + X^2$$

for lattices:

$$F(X) = 1 + 1 + X^2 + X^2$$

Say hello to initial f-algebras!

The type Nat

Formation:

$$\frac{}{\mathbb{N}\ \text{type}}\ \mathbb{N}\text{-F}$$

Introduction:

$$\frac{}{0 : \mathbb{N}}\ \mathbb{N}\text{-}I_0 \qquad \frac{n : \mathbb{N}}{\text{succ}(n) : \mathbb{N}}\ \mathbb{N}\text{-}I_s$$

Elimination:

$\vdash P : \mathbb{N} \to \mathsf{Type}$ ⇝ $n : \mathbb{N} \vdash P\ n\ \mathsf{type}$

$\vdash \mathsf{base} : P\ 0$

$k : \mathbb{N},\ \mathsf{ih} : P\ k \vdash \mathsf{step} : P\ (\mathsf{succ}\ k)$

$\vdash a : \mathbb{N}$

curry

$\mathbb{N}\text{-E}$

$\mathsf{ind}_{\mathbb{N}}(P, \mathsf{base}, \mathsf{step}, a) : P\ a$

$\mathsf{ind}_{\mathbb{N}}(P, \mathsf{base}, \mathsf{step}) : \prod_{a : \mathbb{N}} P\ a$

$(\forall a : \mathbb{N})[P(n)]$

Computation (β)

$$\mathrm{ind}_{\mathbb{N}}(P, \mathrm{base}, \mathrm{step}, 0) \equiv \mathrm{base}$$

$$\mathrm{ind}_{\mathbb{N}}(P, \mathrm{base}, \mathrm{step}, \mathrm{succ}\ k) \equiv \mathrm{step}(k, \mathrm{ind}_{\mathbb{N}}(P, \mathrm{base}, \mathrm{step}, k))$$

What about (η)?

Exponentials

internalizing $(\vdash)$, $(\supset)$, function spaces $(\to)$, ...

lec21

2026-05-29

$$\frac{}{b^a \wedge a \leq b}\ \text{exp.does}$$

$$\frac{c \wedge a \leq b}{c \leq b^a}\ \text{exp.best}$$

exp.? :

$$\frac{\begin{array}{c} c \leq b^a \\ \vdots \end{array}}{c \wedge a \leq b}$$

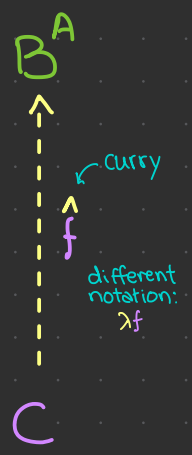
⊖. All lattices with exponentiation $(\Rightarrow)$ are distributive:

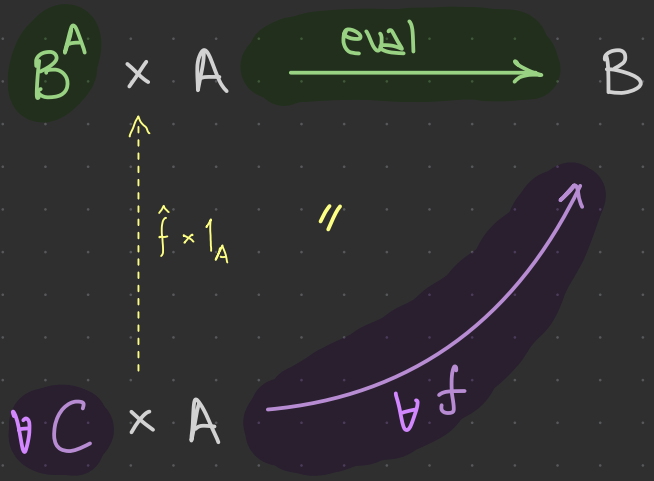
$$\dfrac{\dfrac{\dfrac{}{d \wedge a \leq (d \wedge a) \vee (d \wedge b)}\ \vee.\text{does.L}}{a \leq d \Rightarrow (d \wedge a) \vee (d \wedge b)}\ \text{exp.best} \qquad \dfrac{\dfrac{}{d \wedge b \leq (d \wedge a) \vee (d \wedge b)}\ \vee.\text{does.R}}{b \leq d \Rightarrow (d \wedge a) \vee (d \wedge b)}\ \text{exp.best}}{a \vee b \leq d \Rightarrow (d \wedge a) \vee (d \wedge b)}\ \vee.\text{best}$$

$$\frac{a \vee b \leq d \Rightarrow (d \wedge a) \vee (d \wedge b)}{d \wedge (a \vee b) \leq (d \wedge a) \vee (d \wedge b)}\ \text{exp ?}$$

What about the other "unsafe" half?

Exponential





$$\mathrm{Hom}(C \times A, B) \overset{?}{\cong} \mathrm{Hom}(C, B^A)$$

Free(ly generated)

set (of "generators")

A

WANTED: $\mathcal{M}[A]$ — monoid ("freely generated" by A)

$A = \{a, b, c\}$

insertion of generators

$$A \xrightarrow{\text{ins}} \mathcal{M}[A]$$

to define

$$\mathcal{M}[A] \xrightarrow{\varphi} \mathcal{N}$$

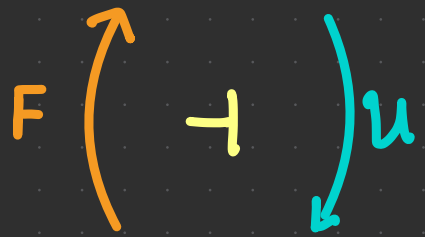
it suffices to:

$$a \longmapsto n$$

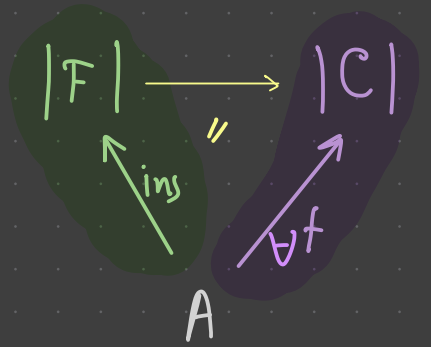
$$b \longmapsto m$$

$$c \longmapsto k$$

Mon

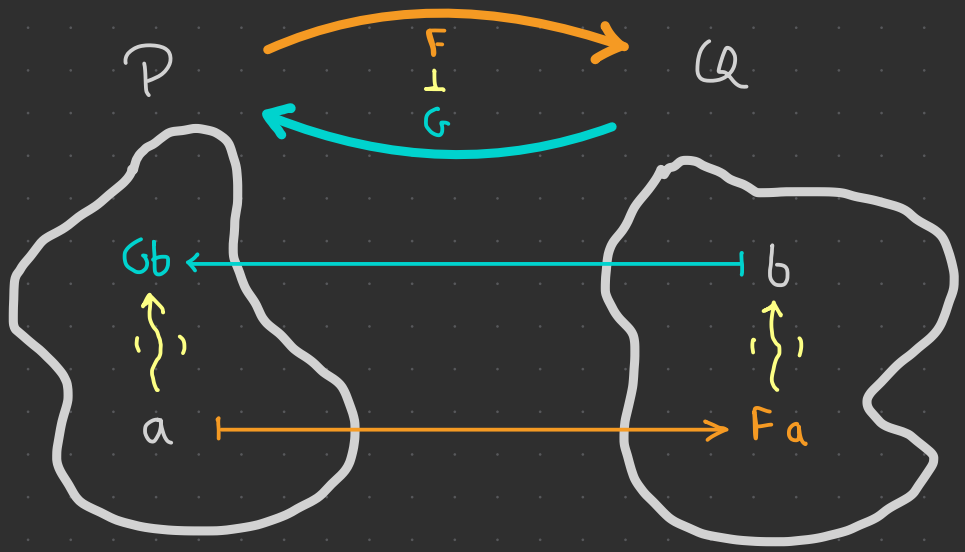


Set



Galois connections (teaser)

How can we compare a with b?



$F \dashv G \overset{\text{def}}{\iff} a \leq_P Gb \iff Fa \leq_Q b$

↳ F is a Lower (also Left) adjoint to G.

What is a congruence?

An equivalence relation that is compatible with the (relevant) operations:

$(\approx)$ is compatible with f $\overset{\text{def}}{\iff}$ $a \approx a' \Rightarrow f\,a \approx f\,a'$

$(=)$ is (supposed to be) a congruence w.r.t. everything.

How do we say this in type theory?