CPP Classes

- First week submissions were great: Keep the hard work up!
  - We will focus only on the difficult bits.
  - Please try! Noticing confusion is encouraged.
- Aidan and I will be marking: you will get two files each week.
- Do feel free to include topic requests for the class at the end of your submission.
  - "Class request: Could you explain X in more detail?
  - "Exercise n was unclear to me."
  - Time is very limited, but we will try our best.
- Please, submit on time. Late submissions may not be graded.
  - Current deadline: 5PM Tuesday.
  - Monday — and earlier — submissions are appreciated.
  - Marking is <u>so much</u> work.
  - Please write your name on the exercises (unless reason not to do so).

PROBLEM SHEET 1

- There are at least two definitions of category.

1. DISPLAYED. A category $\mathbb{C}$ is given by a set of objects, $\mathrm{Obj}(\mathbb{C})$, and a set of arrows, $\mathrm{Arr}(\mathbb{C})$, together with functions

$$\mathrm{src} : \mathrm{Arr}(\mathbb{C}) \longrightarrow \mathrm{Obj}(\mathbb{C}), \quad \text{and} \quad \mathrm{trg} : \mathrm{Arr}(\mathbb{C}) \longrightarrow \mathrm{Obj}(\mathbb{C}), \ \ldots$$

2. INDEXED. A category $\mathbb{C}$ is given by a set of objects, $\mathrm{Obj}(\mathbb{C})$, and, for each two objects $X, Y \in \mathrm{Obj}(\mathbb{C})$, a set of arrows, $\mathbb{C}(X;Y)$, ...

   - Indexed may be less difficult to define ("what is the source of the empty relation?").
   - You may want to know both.
   - Setting these naively in set theory may cause problems: are these disjoint?
   - But I dislike foundational discussion here: just do beautiful maths.
   - Care when defining comma categories: arrows are 'commuting triangles'.

Problem Sheet 1

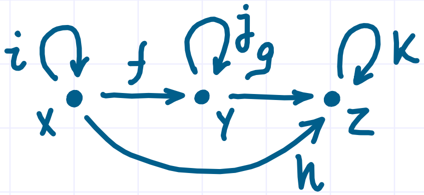
- Proving that something forms a category requires proving the axioms.
  - Your compositions & identities must be associative & unital.
  - Exercise. What goes wrong with the following composition?

$$a(S\circ R)c = \forall b.\ aRb \wedge bSc.$$

- Partial functions: not there unless specified. No need to worry about them.
  - There exists a category, Par, of sets and partial functions.
  - But they do not appear inside Set or Mon.

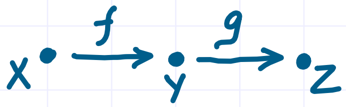
- Personally, I compose in diagrammatic order, $f ; g$, instead of classical, $g \circ f$.
  - Sorry if, going quickly, I mix both.
  - I believe it is more natural in many places.

- Counterexamples should be written explicitly. You only need one.
  - To prove that the union of subcategories is not a subcategory.



$$f \circ i = f = j \circ f; \quad i \circ i = i;$$
$$g \circ j = g = k \circ g; \quad j \circ j = j; \quad g \circ f = h;$$
$$h \circ i = h = k \circ h; \quad k \circ k = k;$$

  - This is tedious: better to use the free category on a graph. (↳ not a commutative diagram.)



Every graph induces a 'free category' whose objects are vertices and whose morphisms are paths on the graph. Identies are empty paths, and composition is concatenation.
Associativity and unitality are proven by induction on path length.

Problem Sheet 2

To myself: take attendance!

- Second week submissions were also great: thaks for keeping it up.
  - Little to say in Ex 1 and Ex 4, unless anything you may want to discuss.
  - Grading a bit more strictly so that it becomes more useful feedback.

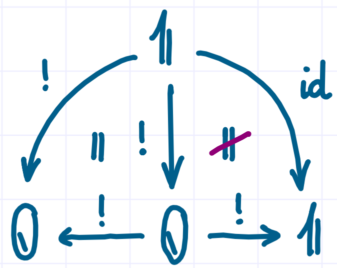
- Counterexamples must be explicit.
  - Showing that $A \times B$ is not the product of A and B in general requires us to pick A, B, some C, and morphisms $f_1 : C \to A$ and $f_2 : C \to B$. If you use a class of counterexamples, just show it is not empty.

- Universal arrows are not 'unique' but 'unique such that'.
  - Except for terminal/initial, there is important information in that 'such that'.
  - "Unique such that A" is not implied by "Unique such that A and B".
  - But, if the "unique such that A" satisfies B, it is the "Unique such that A and B".

EXERCISE 5.

- The cartesian product is not a product in PAR. How would you prove it?

① Look for an easy counterexample.



② Counting; counterexample in finite sets,

$$\#\mathrm{Par}(X;A\times B) = (\#A \times \#B + 1)^{\#X}$$
$$\#\mathrm{Par}(X;A)\times\#\mathrm{Par}(X;B) = (\#A+1)^{\#X}\times(\#B+1)^{\#X}$$

and pick, say, $X = A = B = 1$.

- The category PAR of partial functions does have products.

$$A \circledast B = A\times B + A + B$$

or also

$$A \circledast B = ((A+\{*\})\times(B+\{*\})) - \{(*,*)\}.$$

EXERCISE 5.

- But how would we find them?

① Counting.

$$
\begin{aligned}
&\mathrm{PAR}(X;Y)\times\mathrm{PAR}(X;Z)\\
\cong\ &\mathrm{SET}(X;Y+1)\times\mathrm{SET}(X;Z+1)\\
\cong\ &\mathrm{SET}(X;(Y+1)\times(Z+1))\\
\cong\ &\mathrm{SET}(X;Y\times Z+Y+Z+1)\\
\cong\ &\mathrm{PAR}(X;Y\times Z+Y+Z).
\end{aligned}
$$

$$A\times B+A+B$$

⓪ Thinking about the partial functions we need.

② Via pointed sets.

Pointed sets form a category — $f:(A,a)\to(B,b)$ means $f:A\to B$ with $f(a)=b$ — and an easy product.

$$(A,a)\times(B,b)=(A\times B,(a,b)).$$

Pointed sets are equivalent to partial functions.

$$X\overset{F}{\longmapsto}(X\uplus\{*\},*)$$

$$\mathrm{PAR}\ \underset{G}{\overset{F}{\rightleftarrows}}\ \mathrm{POINTEDSET}\quad(\cong)$$

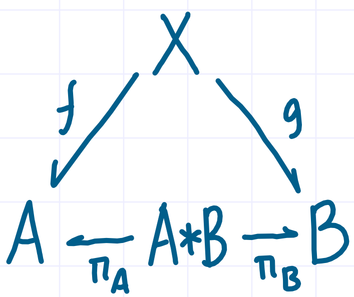
$$A-\{a\}\overset{G}{\longleftarrow\!\!\!\shortmid}(A,a)$$

We could transport the product: $A\circledast B=G^{-1}(F(A)\times F(B))$.

$$(A\uplus\{*\})\times(B\uplus\{*\})\,/\,\{(*,*)\}$$

EXERCISE 5.

How to prove this is a product?



The elements of $A*B = A + A\times B + B$ can be written as $L(a)$ for $a \in A$, $R(b)$ for $b \in B$ and $B(a,b)$ for $a \in A$ and $b \in B$. We write $\uparrow$ for 'non-defined'.

$\pi_A(La) = a$; $\quad \pi_B(La)\uparrow$
$\pi_A(Rb)\uparrow$ $\quad \pi_B(Rb) = b$;
$\pi_A(B(a,b)) = a$; $\quad \pi_B(B(a,b)) = b$;

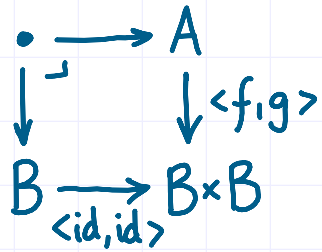
Let us define $h : X \to A*B$ by cases.

1) If $f(x)\uparrow$ and $g(x)\uparrow$, then we need $\pi_A(h(x))\uparrow$ and $\pi_B(h(x))\uparrow$, so $h(x)\uparrow$.
2) If $f(x)\uparrow$, then we need $\pi_A(h(x))\uparrow$ and $\pi_B(h(x)) = g(x)$, so $h(x) = R(g(x))$.
3) If $g(x)\uparrow$, then we need $\pi_B(h(x))\uparrow$ and $\pi_A(h(x)) = f(x)$, so $h(x) = L(f(x))$.
4) Otherwise, we need $\pi_A(h(x)) = f(x)$ and $\pi_B(h(x)) = g(x)$, so $h(x) = B(f(x), g(x))$.

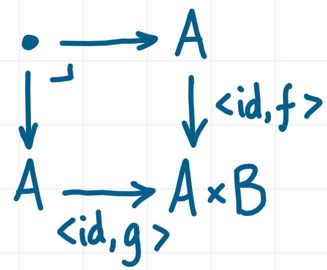
EXERCISE 3.

- How to come up with the solution: you may want to sketch on SETS.

Equalizer of f and g $\cong \{ a \mid f(a) = g(a) \}$
$\cong \{ a, b \mid f(a) = id(b),\ g(a) = id(b) \}$
$\cong \{ a, b \mid \langle f, g\rangle (a) = \langle id, id \rangle (b) \}$
$\cong$ pullback of $\langle f, g\rangle$ and $\langle id, id\rangle$

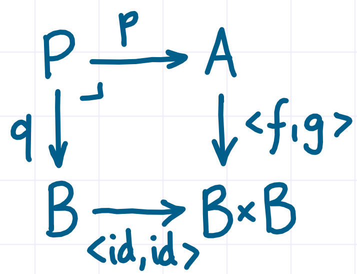


Equalizer of f and g $\cong \{ a \mid f(a) = g(a) \}$
$\cong \{ a_1, a_2 \mid id(a_1) = id(a_2),\ f(a_1) = g(a_2) \}$
$\cong \{ a_1, a_2 \mid \langle id, f\rangle (a_1) = \langle id, g \rangle (a_2) \}$
$\cong$ pullback of $\langle id, f\rangle$ and $\langle id, g\rangle$

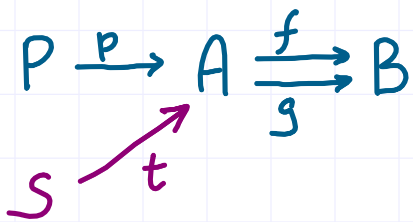


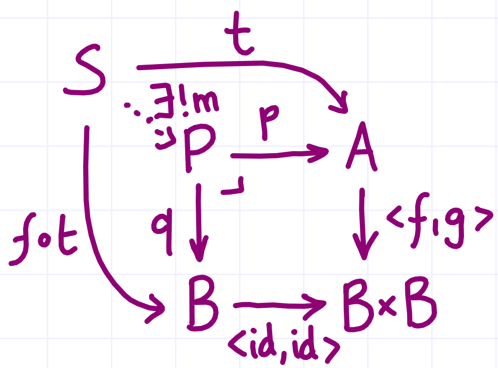
- Later on, you will be able to use these as formal proofs.
- For now, we will do diagram chasing.

EXERCISE 3.



We Know $f \circ p = q$ and $g \circ p = q$.
Then, we have a candidate equalizer





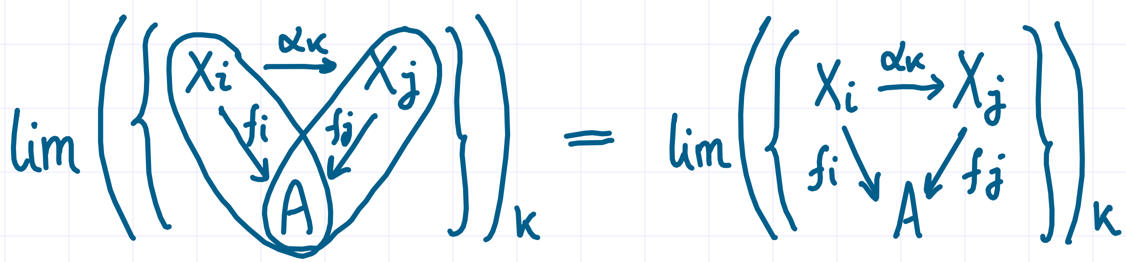
And for any other $t : S \to A$ such that $f \circ t = g \circ t$, we have a candidate pullback.

Thus, there exists a unique $m : S \to P$ SUCH THAT $p \circ m = t$ AND $q \circ m = f \circ t$.
But $q = f \circ t$, so the first condition implies the second. ⚠
Thus, there exists a unique $m : S \to P$ SUCH THAT $p \circ m = t$.

EXERCISE 6.

Strategy 1. Show terminals and pullbacks / products and equalisers.
Strategy 2. For any diagram in $\mathbb{C}/A$, we get a diagram in $\mathbb{C}$ sharing the limit.

If you have functors already, promote $F: I \to \mathbb{C}$ into $F^*: I^* \to \mathbb{C}$, where I^* has an extra object and unique morphisms from any other object to it.



CPP - 27th NOV

Take attendance.

Great submissions as always; thanks for the effort.

- Pullbacks and mediating arrows: let us try slowly.
- The rest was less problematic, but we can detail it more.

Grading.

- Your PS5 may not be marked yet: sorry! We will discuss it next week.
- If you have submitted, PS3 and PS4 should be marked.

Today, we will see PS3 and PS4; we will see PS5 and PS6 next week.

- Let us do the two hours with a 10 minute break in between.

CPP - Sheet 3

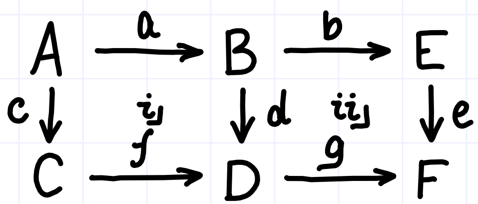
Exercise 1. We will only do part (b), very carefully.

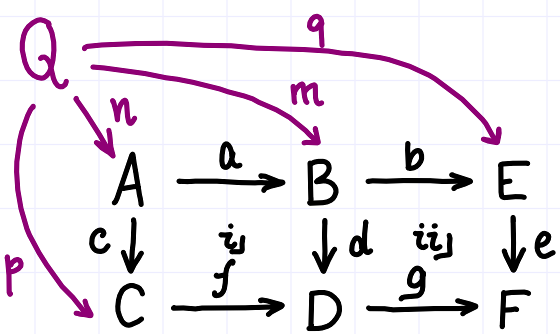
- Always in mind: are you proving uniqueness, existence, or both at the same time? If unsure, split!
- Pullback candidates are triples (Q, p, q) not objects (Q).
- We need to assume that the squares commute.
- This is not an exercise on pullbacks, as much as an exercise on patience.
- Let us write cone morphisms, $m : (Q, p, q) \to (A, c, a)$, so that uniqueness is clear.

Exercise 2.

- Notation: "F is faithful because $F(f) = f$" as with inclusions.

CPP - Sheet 3





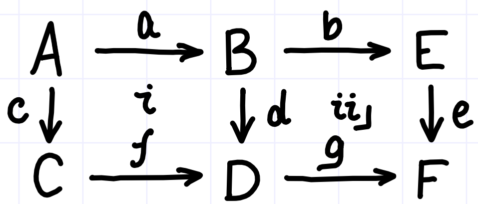
Exercise 1.a.

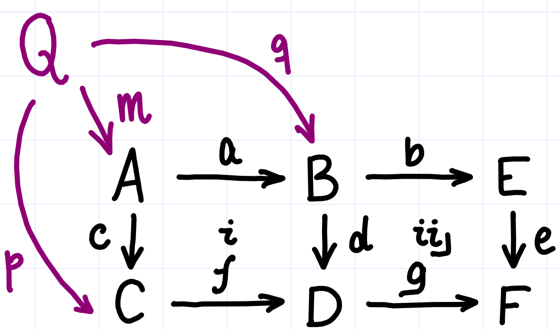
1. Let (Q,p,q) be a $(g\circ f, e)$-cone.
2. Then, $(Q, f\circ p, q)$ is a (g,e)-cone, because $g\circ f\circ p = e\circ q$ by 1.
3. There exists a unique $m: (Q, f\circ p, q) \to (B, d, b)$, by i.
4. Now, (Q,p,m) is a (f,d)-cone, because $f\circ p = d\circ m$ by 3.
5. There exists a unique $n: (Q,p,m) \to (A,c,a)$, by ii.
6. Now, $n: (Q,p,q) \to (A,c,b\circ a)$, because $c\circ n = p$ by 5 and $b\circ(a\circ n) = b\circ m = q$ by 5 and 3.
7. Let $n': (Q,p,q) \to (A,c,b\circ a)$.
8. Then $a\circ n': (Q, f\circ p, q) \to (B,d,b)$ and $a\circ n' = m$ because $d\circ a\circ n' = f\circ c\circ n' = f\circ p$ by i and 7, and $b\circ a\circ n' = q$ by 7.
9. Then $n': (Q,p,m) \to (A,c,a)$ and $n' = n$ because because $c\circ n' = p$ by 7 and $a\circ n' = m$ by 8.

Thus, there exists a unique $n: (Q,p,q) \to (A,c,b\circ a)$.

CPP - Sheet 3

Exercise 1. b.





① Let (Q,p,q) be a (f,d)-cone.

② Then, $(Q,p,b\circ q)$ is a $(g\circ f,e)$-cone, because $g\circ f\circ p = g\circ d\circ q = e\circ b\circ q$ by 1 and ii.

③ There exists a unique $m:(Q,p,b\circ q)\to(A,c,b\circ a)$, by iii.

④ Now, $q:(Q,f\circ p,b\circ q)\to(B,d,b)$ because $d\circ q = f\circ p$ by 1.

⑤ Also, $a\circ m:(Q,f\circ p,b\circ q)\to(B,d,b)$ because $b\circ a\circ m = b\circ q$ by 3, and $d\circ a\circ m = f\circ c\circ m = f\circ p$, by i and 3. Thus, $a\circ m = q$.

⑥ Then, $m:(Q,p,q)\to(A,c,a)$, because $c\circ m = p$ by 3 and $a\circ m = q$ by 5.

⑦ Let $m':(Q,p,q)\to(A,c,a)$.

⑧ Then, $m':(Q,p,b\circ q)\to(A,c,b\circ a)$, and $m=m'$ by 3, because $c\circ m' = p$ by 7 and $b\circ a\circ m' = b\circ q$ by 7.

Thus, there exists a unique $m:(Q,p,q)\to(A,c,a)$.

CPP - Sheet 3

$$\begin{array}{ccccc} A & \xrightarrow{a} & B & \xrightarrow{b} & E \\ {\scriptstyle c}\downarrow & i & \downarrow{\scriptstyle d} & ii\lrcorner & \downarrow{\scriptstyle e} \\ C & \xrightarrow{f} & D & \xrightarrow{g} & F \end{array}$$

Exercise 1. b.

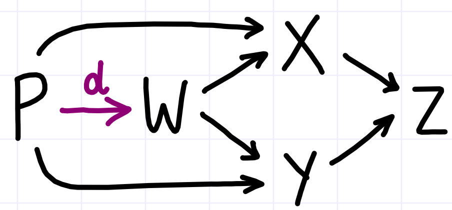
Can we do without assuming that (i) commutes?

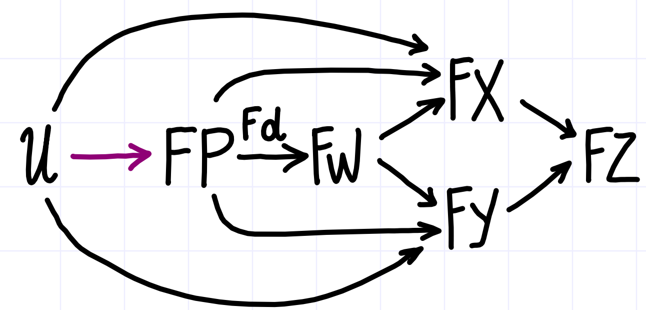
$$\begin{array}{ccccc} 2 & \xrightarrow{not} & 2 & \longrightarrow & 1 \\ | & & | & ii\lrcorner & \downarrow \\ 2 & \text{———} & 2 & \longrightarrow & 1 \end{array}$$

Here, ii and iii are pullbacks (products), but i is not.

CPP - SHEET 4

EXERCISE 1.i. Pullbacks are weak pullbacks, for the "only if". For the if,

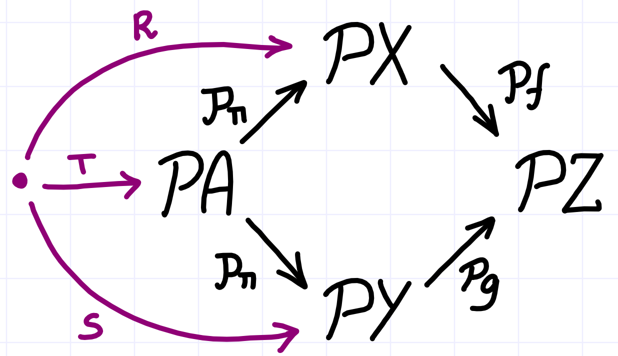




By W being a weak pullback. By FP being a weak pullback.

CPP - SHEET 4

EXERCISE 1.ii.



Note that $A = \{(x,y) \mid fx = gy\}$.

Let $R \in PX$ and $S \in PY$ such that $f(R) = g(S)$. Because of this, for each $x \in R$, there exists some $y_x \in S$ such that $f(x) = g(y_x)$; for each $y \in S$, there exists some $x_y \in R$ such that $f(x_y) = g(y)$. Pick

$$T = \{(x, y_x) \mid x \in R\} \cup \{(x_y, y) \mid y \in S\} \subseteq PA.$$

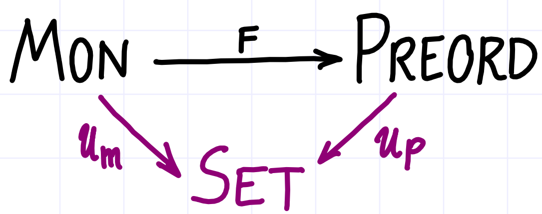
Alternatively, $T' = \{(x,y) \in A \mid x \in R, y \in S\}$, and then we see that the projections are precisely R and S.

CPP - SHEET 3

EXERCISE 2. We need to check the following.

- Each monoid induces a preorder.
- Each monoid homomorphism induces a monotone function.

- Faithfulness is 'trivial', but why?



$$Ff = Fg$$
$$U_p Ff = U_p Fg$$
$$U_m f = U_m g$$
$$f = g$$

- Not full; the function $(+1) : \mathbb{N} \to \mathbb{N}$ is monotone but not homomorphism.

CPP - SHEET 3

EXERCISE 5.a. We can find an example with two objects.

$$X \xrightarrow{f} Y \qquad\qquad A \underset{k}{\overset{h}{\rightleftarrows}} B \qquad \begin{array}{l} h \circ k = id_B \\ k \circ h = id_A \end{array} \qquad\qquad F(f) = h.$$

But even with a simpler idea, we get a faithful functor: $X \xrightarrow{f} Y$ to $\bullet^{1}$; but not full!

EXERCISE 5.b. Let $f: X \to Y$ and $Ff: FX \to FY$ an iso. By fullness, there exists $g: B \to A$ such that $Fg = (Ff)^{-1}$. Then, $F(g \circ f) = id$ and $F(f \circ g) = id$, by functoriality. By faithfulness, $g = f^{-1}$.

EXERCISE 5.c. The only point left is $F(id) = id$.

EXERCISE 5.d. By inclusion, $X \xrightarrow{f} Y$ into $Z \underset{b}{\overset{a}{\rightrightarrows}} X \xrightarrow{f} Y$, with $f \circ a = f \circ b$.

CPP - SHEET 4

EXERCISE 1.iii.

$$\begin{array}{ccc} X\times X & \xrightarrow{\pi_1} & X \\ {\scriptstyle \pi_2}\downarrow & /\!/ & \downarrow{\scriptstyle !} \\ X & \xrightarrow[!]{} & Z \end{array} \qquad \begin{array}{ccc} \mathcal{P}(X\times X) & \xrightarrow{\mathcal{P}\pi} & \mathcal{P}X \\ {\scriptstyle \mathcal{P}\pi}\downarrow & & \downarrow{\scriptstyle \mathcal{P}!} \\ \mathcal{P}X & \xrightarrow[\mathcal{P}!]{} & \mathcal{P}Z \end{array}$$

$\mathcal{P}X = \{\emptyset, \{a\}, \{b\}, \{a,b\}\}$.

$\mathcal{P}!(\emptyset) = \emptyset$

$\mathcal{P}!(x) = \{*\}$ for $x \neq \emptyset$.

Thus $\mathcal{P}(X\times X)$ has 16 elements, but the pullback must have $3\times 3+1 = 10$.

CPP - SHEET 4

EXERCISE 2.

Let $f : A \to B$. Let $\hat{f}_X : C(B,X) \to C(A,X)$ be defined by $f_X(g) = g \circ f$.

For any $h : X \to Y$, we use associativity,

$$\begin{array}{ccc}
C(B,X) & \xrightarrow{\hat{f}_X} & C(A,X) \\
\downarrow{\scriptstyle C(B,h)} & & \downarrow{\scriptstyle C(A,h)} \\
C(B,X) & \xrightarrow{\hat{f}_Y} & C(A,X)
\end{array}$$

$$C(A,h)(\hat{f}_X(g)) = h \circ g \circ f = \hat{f}_Y(C(B,h)(g)).$$

CPP - SHEET 4

EXERCISE 3. Let us define $\delta_X : X \to X \times X$ by $\delta_X(x) = (x,x)$. This is natural because $(f \times f)(\delta(x)) = \delta(f(x))$ by definition.

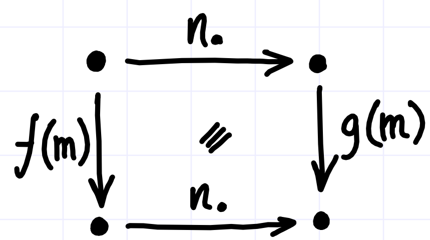
Assume any natural $\alpha_X : X \to X \times X$, by naturality, $\alpha_X(x) = (x,x)$.

$$\begin{array}{ccc} 1 & \xrightarrow{\;!\;} & 1 \times 1 \\ x \big\downarrow & & \big\downarrow (x,x) \\ X & \xrightarrow{\;\alpha_X\;} & X \times X \end{array}$$

"By Yoreda lemma, $\mathrm{Nat}(\mathrm{Set}(1,\cdot), \Delta) \cong \Delta 1 \cong 1$, using that $\mathrm{Id} \cong \mathrm{Set}(1,\cdot)$."

CPP - SHEET 4

EXERCISE 4. Functors between monoids are monoid homomorphisms, $f : M \to N$.



Natural transformations $n : f \to g$ are elements such that $n \cdot f(m) = g(m) \cdot n$ for each $m \in M$.

"Elements that conjugate f and g: these are conjugate homomorphisms."

CPP - 4th NOV

- Attendance!

- Adjunctions are fun, let us try to make them easier.
- Dedekind cuts are confusing, let us try to run away from that exercise.

(!) Project supervision in Category Theory: either Elena Di Lavore or me, remotly, or Ruben van Belle and Paolo Perrone (Sam Staton's group). Bartek may have more proposals.

(!) PhD positions in Tallinn, Estonia; in the Compositional Systems and Methods Group.
  - 4-year research positions in applications of category theory, categorical logic, & program semantics.

CPP-6 EXERCISE 1.

(!) We do not need to prove that $\mathcal{P}$ is a functor. It follows from freeness.

Let us prove that $(\mathcal{P}X, \bigcup)$ is the free complete semilattice over a set X. First, we should have checked that it is a semilattice: the least upper bound of $\{S_i\}_{i\in I} \in \mathcal{P}X$ is $\bigvee_{i\in I} S_i = \bigcup_{i\in I} S_i$.
Let $\eta_X : X \longrightarrow U\mathcal{P}X$ be defined by $\eta_X(x) = \{x\}$.

Given a complete semilattice, $(Q,\bigvee)$, and a function $f : X \to U(Q,\bigvee)$, let us assume a morphism $f^\# : (\mathcal{P}X, \bigcup) \to (Q,\bigvee)$ such that $Uf^\# \circ \eta_X = f$. It must be that

$$f^\#(S) = f^\#\left(\bigcup_{s\in S}\{s\}\right) = \bigvee_{s\in S} f^\#\{s\} = \bigvee_{s\in S} f^\#(\eta_X(s)) = \acute{\bigvee}_{s\in S} f(s).$$

⚠ This is the only possible homomorphism; but we need to check it is.

Let us prove $f^\#$ is a homomorphism. We need to check the formula for the least upper bound of a union.

$$f^\#\left(\bigcup_{i\in I} S_i\right) \overset{\text{def}}{=} \bigvee_{x\in \bigcup_{i\in I} S_i} f(x) \overset{\text{l.u.}}{=} \bigvee_{i\in I}\bigvee_{x\in S_i} f(x) \overset{\text{def}}{=} \bigvee_{i\in I} f^\#(S_i).$$

□

CPP-6

EXERCISE 3.

(!) For posets, finding a left adjoint to $h : \mathcal{D} \to \mathcal{C}$ means finding, for each $x \in C$, some $\ell x \in D$ such that $x \leq h\ell x$ and $x \leq hy$ implies $\ell x \leq y$; finding a right adjoint means finding, for each $x \in C$, some $rx \in D$ such that $hrx \leq x$ and $hy \leq x$ implies $y \leq rx$.

$$\frac{A \subseteq p^{-1}(B)}{pA \subseteq B} \qquad\qquad \frac{p^{-1}(B) \subseteq A}{B \subseteq \{x \mid p^{-1}(x) \subseteq A\}} = p^*(A)$$

$$A \subseteq p^{-1}p(A) \qquad\qquad p^{-1}(\{x \mid p^{-1}(x) \subseteq A\}) \subseteq A$$

Now, consider the projection $\pi : X \times Y \to Y$.

$$\pi(S) = \{y \mid \exists x.\ (x,y) \in S\}$$
$$\pi^*(S) = \{y \mid \forall x.\ (x,y) \in S\}.$$

CPP-5

$$\begin{array}{ccc} P & \xrightarrow{y} & \mathrm{Set}^{P^{op}} \\ & \searrow^{y'} & i \uparrow \downarrow j \\ & & 2^{P^{op}} \end{array}$$

EXERCISE 2.a.

We note that $|P(x;y)| \leq 1$. We may define functors $i : 2 \to \mathrm{Set}$ and $j : \mathrm{Set} \to 2$ by inclusion and $jX = \begin{cases} 0 & \text{if } X = \bot, \\ 1 & \text{otherwise.} \end{cases}$

EXERCISE 2.b.

Given a downwards closed set S, we pick $\hat{S}(a) = \begin{cases} 1 & \text{if } a \in S \\ 0 & \text{if } a \notin S. \end{cases}$ Given a functor, $\hat{T}$, we pick $T = \{a \mid \hat{T}(a) = 1\} = \hat{T}^{-1}(1)$. These are inverses.
We are left to show T is downwards closed and $\hat{S}$ extends to a functor.

EXERCISE 2.c.

A natural transformation $\alpha : \hat{S} \to \hat{T}$ consists of a family $\alpha_a : \hat{S}(a) \longrightarrow \hat{T}(a)$, meaning $a \in S$ implies $a \in T$. That is, $S \subseteq T$. In this case, it exists and is natural.

EXERCISE 2.d.

Thus, $S \subseteq T$ and $T \subseteq S$ imply $S = T$.

CPP-5

EXERCISE 2.e.

Each extended real number $r \in \overline{\mathbb{R}}$ induces a Dedekind cut $\hat{S}_r \in 2^{\mathbb{Q}^{op}}$, which fully faithfully embeds into $Set^{\mathbb{Q}^{op}}$. Morphisms are inclusions, and Dedekind cut inclusions are the less-or-equal poset in REL.

EXERCISE 2.f.

Products correspond to intersections. In particular, every Dedekind cut is an intersection of representable rationals.

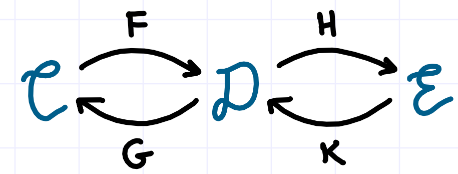
$$\prod_{r \leq q} \mathbb{Q}(\cdot, q) \text{ is the Dedekind cut of } r.$$

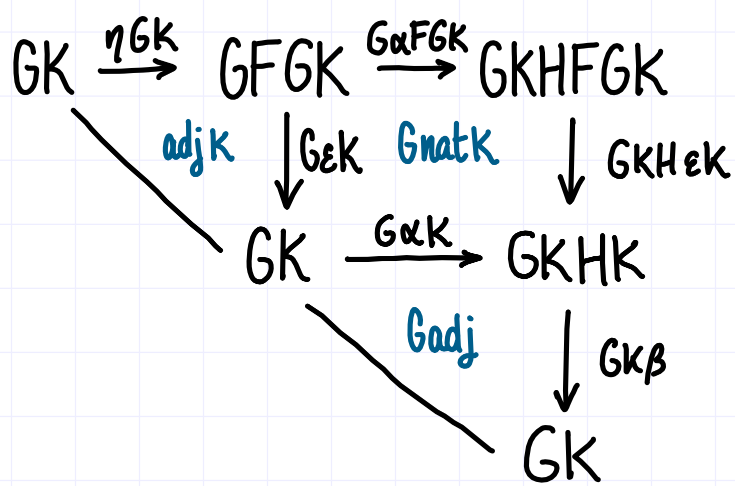
CPP-6

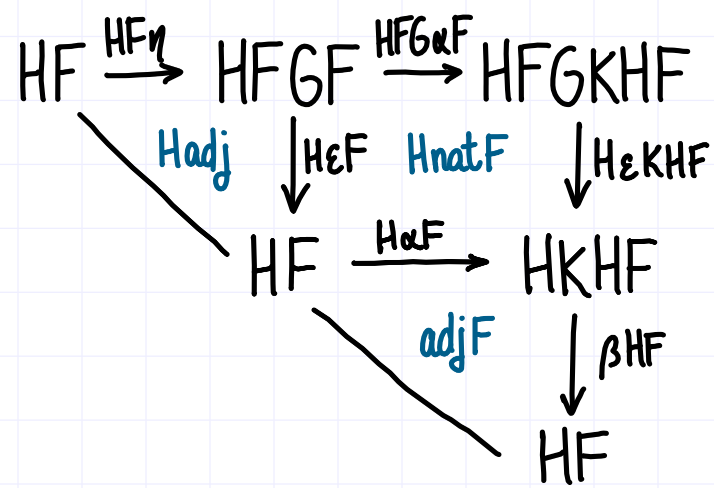
EXERCISE 4. $\eta : 1 \to GF$, $\varepsilon : FG \to 1$, $\alpha : 1 \to KH$, $\beta : HK \to 1$.

Unit: $1 \xrightarrow{\eta} GF \xrightarrow{G\alpha F} GKHF$

Counit: $HFGK \xrightarrow{H\varepsilon K} HK \xrightarrow{\beta} 1$







CPP - 4th DEC.

STRING DIAGRAMS.

Proving theorems in formal category theory quickly becomes tedious: chasing diagrams and applying naturality is prone to error.

$F : \mathbb{C} \to \mathbb{D} \quad H : \mathbb{C} \to \mathbb{D}$

$G : \mathbb{D} \to \mathbb{E} \quad K : \mathbb{D} \to \mathbb{E}$

$\alpha : GA \longrightarrow KA$

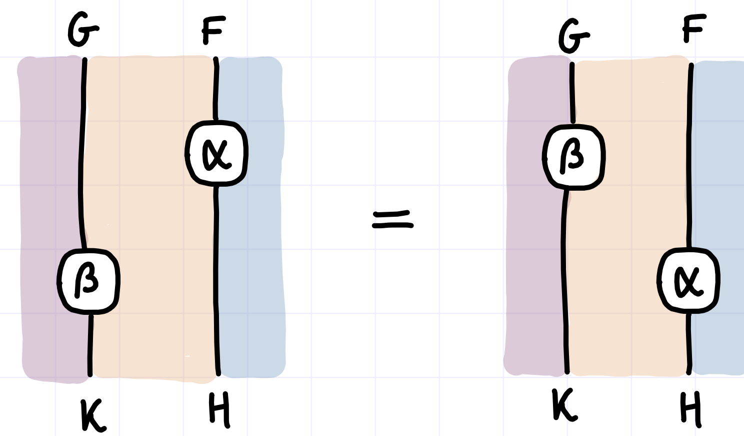
$\beta : FX \longrightarrow HX$

$$\begin{array}{ccc} GFX & \xrightarrow{G\alpha} & GHX \\ {\scriptstyle \beta F}\downarrow & \mathrm{nat}(\beta) & \downarrow{\scriptstyle \beta H} \\ KFX & \xrightarrow{K\alpha} & KHX \end{array}$$

(!) Categories are regions.

(!) Functors are wires.

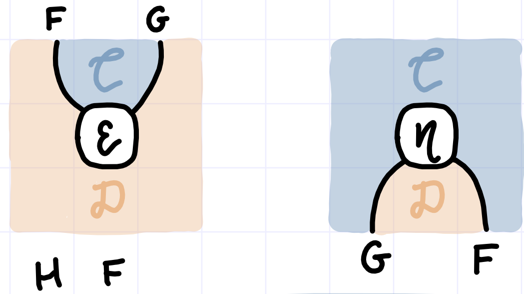
(!) Natural transformations are nodes.

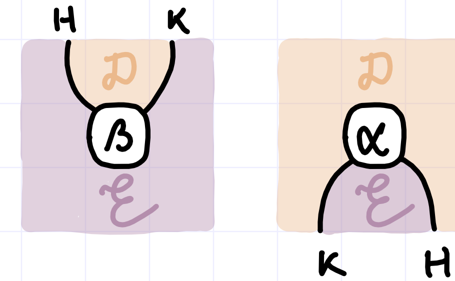


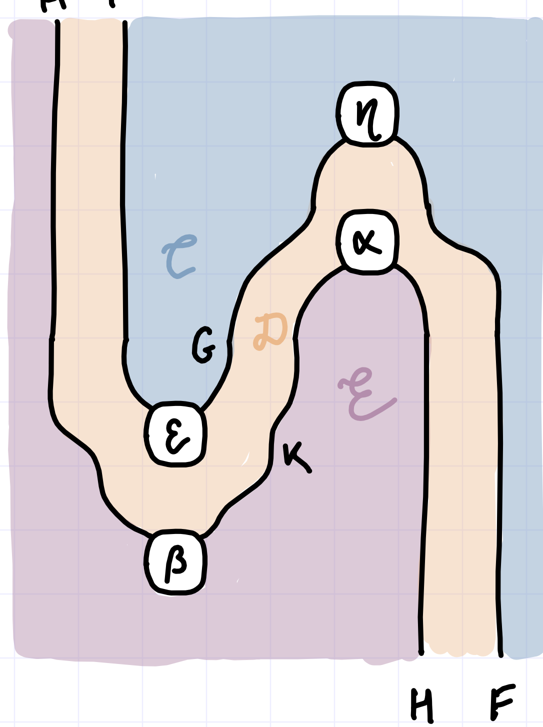
String diagrams provide a sound and complete calculus for bicategories: proving this would take an entire course, but they are intuitive anyway.

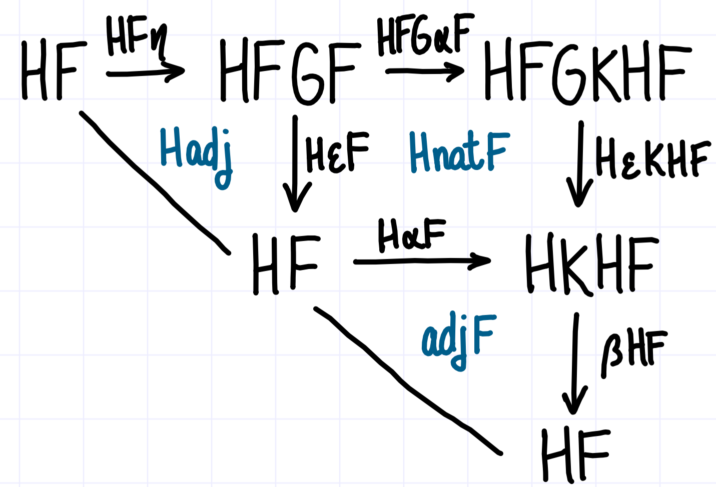
CPP- 4th DEC.

ADJUNCTIONS COMPOSE





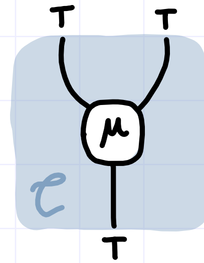




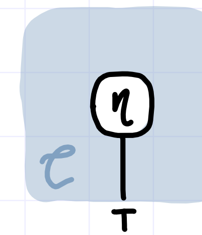
CPP - 4th DEC.

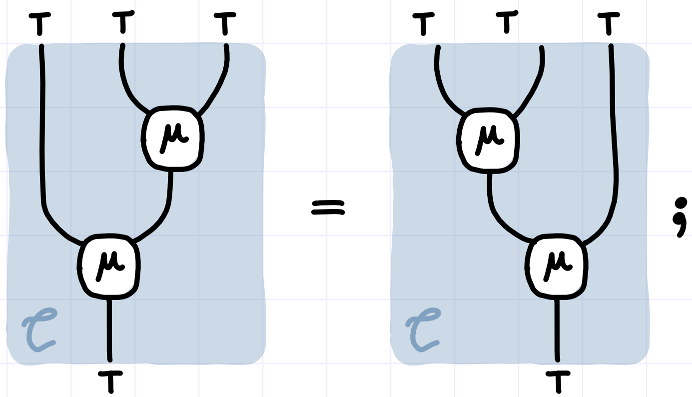
MONAD.

$\mu_X : TTX \to TX$

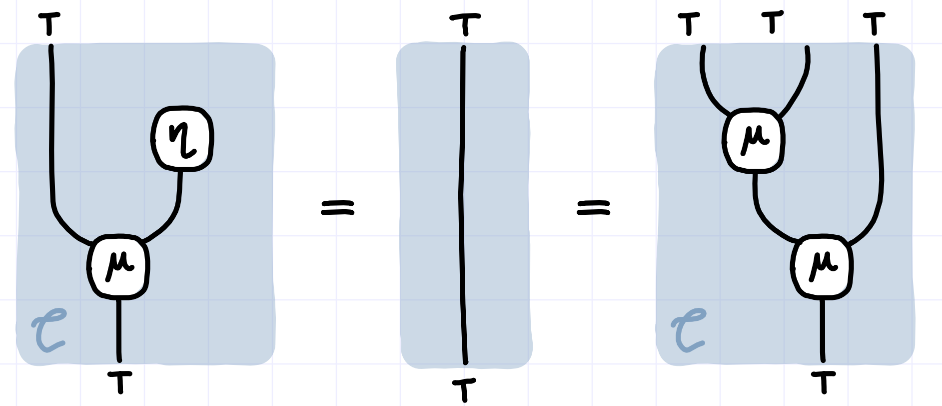


$\eta_X : X \to TX$



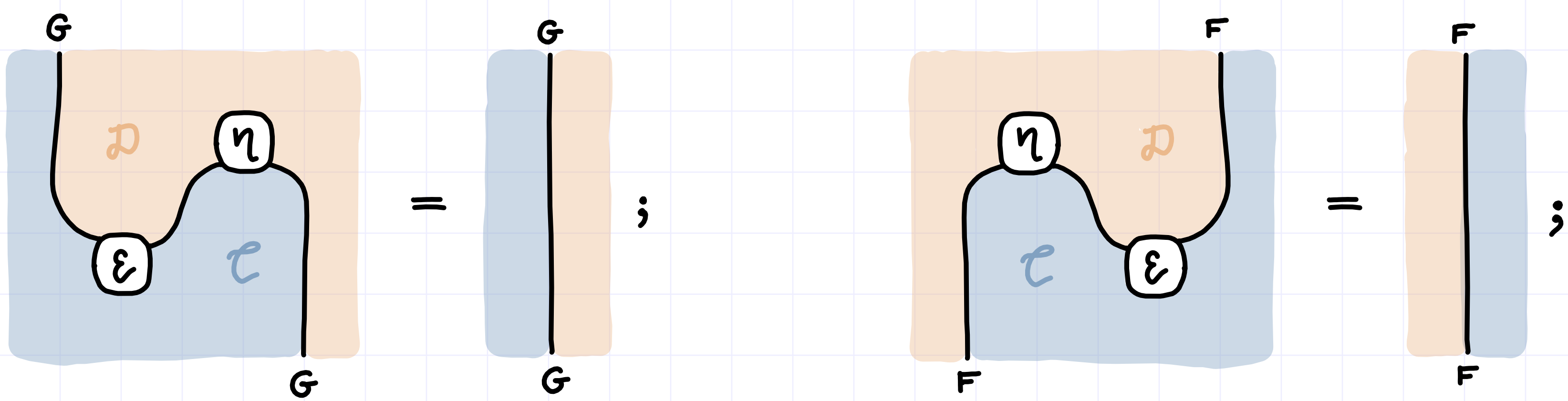


Associativity



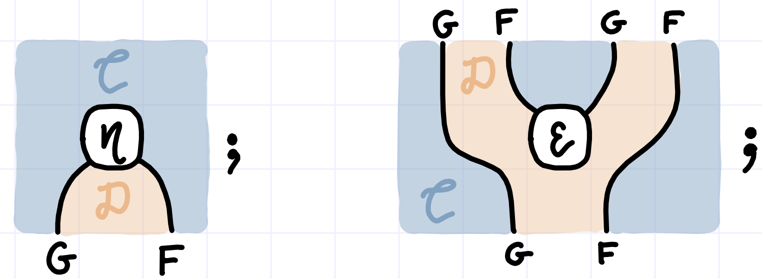
Unitality.

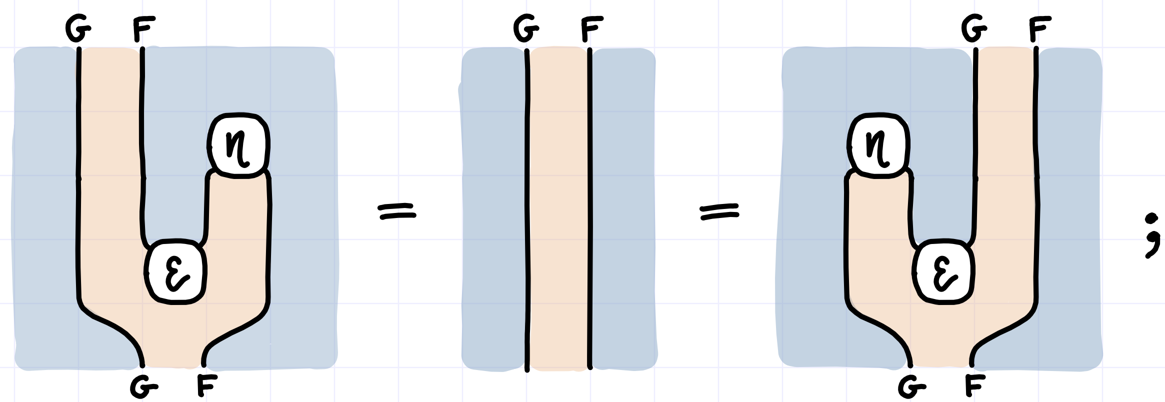
CPP - 4th DEC. ADJUNCTION.

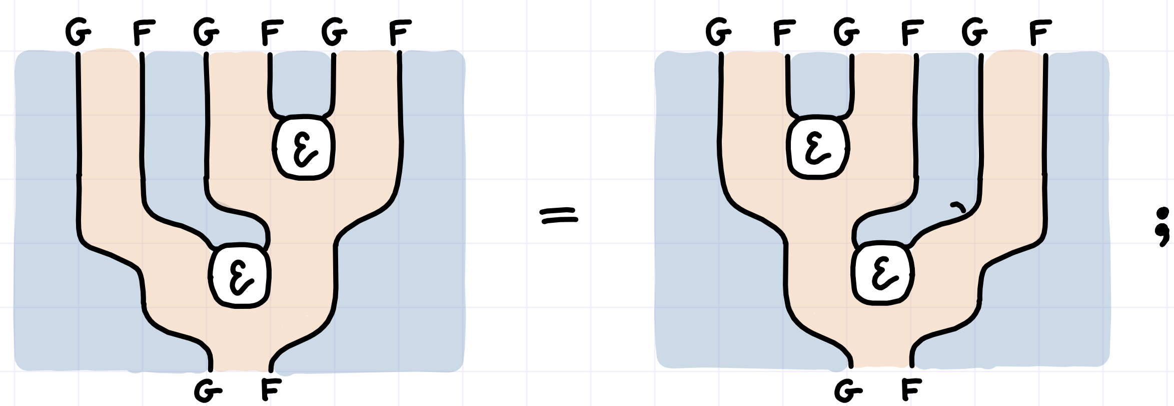


CPP - 4th DEC.

ADJUNCTIONS INDUCE MONADS.







A Preliminaries

Proposition A.1 (Reducing an adjunction)**.** *Let* $F\colon \mathbb{A} \to \mathbb{C}$ *and* $H \mathbin{⨾} U\colon \mathbb{C} \to \mathbb{A}$ *determine an adjunction* $(F, H \mathbin{⨾} U, \eta, \varepsilon)$ *and let* $P\colon \mathbb{B} \to \mathbb{C}$ *determine a second adjunction* (P, H, u, c) *such that the unit* $u\colon I \to P \mathbin{⨾} H$ *is a natural isomorphism (as in Figure 7). Then,* $F \mathbin{⨾} H$ *is left adjoint to* U*.*

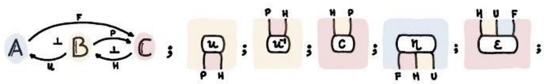


Figure 7: Setting for reducing an adjunction.

Proof. We employ the string diagrammatic calculus of bicategories to the bicategory of categories, functors and natural transformations. We define the morphisms in Figure 8 to be the unit and the counit of the adjunction. We then prove that they satisfy the snake equations in Figures 8 and 9.

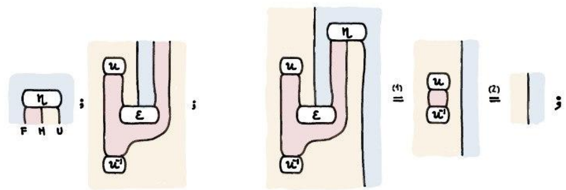


Figure 8: Unit and counit of the reduced adjunction (left). First snake equation (right).

In the first snake equation, in Figure 8, we use *(i)* that there is a duality (η, ε), and *(ii)* that u is invertible. In the second snake equation, in Figure 9, we use *(i)* that there is a duality (u, c), *(ii)* that u is invertible, *(iii)* that there is a duality (u, c), again; and *(iv)* that there is a duality (η, ε).

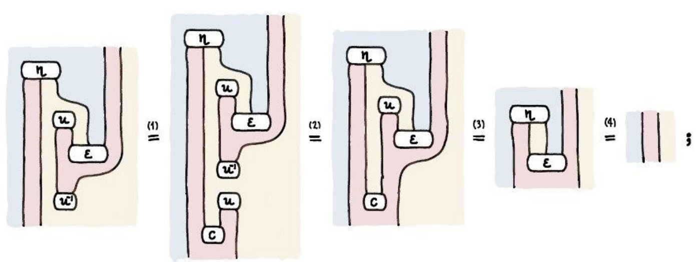


Figure 9: Second snake equation.

CPP - 4th DEC.

Addressing the question on adjoints.

Constructing adjoints: Mac Lane's book [Sec IV, Thm 2] shows an efficient recipe to construct adjoints. This used to be an exercise for CPP, but perhaps you did not have time to see it during classes, and my solutions for PS6 assumed you knew it (sorry!).

Theorem 2. *Each adjunction $\langle F, G, \varphi \rangle : X \rightharpoonup A$ is completely determined by the items in any one of the following lists:*

This is the definition on your lecture slides; note how it asks for a functor and a nat. transf.

(i) *Functors F, G, and a natural transformation $\eta : 1_X \dot{\to} GF$ such that each $\eta_x : x \to GFx$ is universal to G from x. Then φ is defined by (6).*

This one is what I use normally. It is efficient because we do not prove F a functor nor η natural.

(ii) *The functor $G : A \to X$ and for each $x \in X$ an object $F_0 x \in A$ and a universal arrow $\eta_x : x \to GF_0 x$ from x to G. Then the functor F has object function F_0 and is defined on arrows $h : x \to x'$ by $GFh \circ \eta_x = \eta_{x'} \circ h$.*

These are dual.

(iii) *Functors F, G, and a natural transformation $\varepsilon : FG \dot{\to} I_A$ such that each $\varepsilon_a : FGa \to a$ is universal from F to a. Here φ^{-1} is defined by (7).*

(iv) *The functor $F : X \to A$ and for each $a \in A$ an object $G_0 a \in X$ and an arrow $\varepsilon_a : FG_0 a \to a$ universal from F to a.*

And triangles, which you know by now.

(v) *Functors F, G and natural transformations $\eta : I_X \dot{\to} GF$ and $\varepsilon : FG \dot{\to} I_A$ such that both composites (8) are the identity transformations. Here φ is defined by (6) and φ^{-1} by (7).*

Thanks to all of you.