Category Theory Notes

Hasith Vattikuti

Chapter 1: Generative effects: Orders and adjunctions

1.1 More than the sum of their parts

- We begin by defining some relation $\leq$ on a system S as well as a function $\Phi : S \to \mathbb{B}$
- A basic ordered structure is a **preorder**, where some elements are related to each other. In a sense, the boolean set is also a preorder since false $\leq$ false, false $\leq$ true, and false $\leq$ true if we $a \leq b$ is defined as $a \Rightarrow b$.
- We have a possibility of a **generative effect** when $\phi(A) \vee \phi(B) \leq \phi(A \vee B)$. In the case where the inequality is strict, we do indeed have a generative effect

1.2 What is order?

Definition 1.12. A *relation* between X and Y is a subset $R \subseteq X \times Y$. A *binary relation* on X is a relation between X and X (so a relation between X and X).

- I think this definition is trying to say that if we have a binary relation such as $\leq$ on $\mathbb{N}$, the R will look like $\{(1,2),(1,3),\ldots,(2,3),(2,4),\ldots\}$. This is further confirmed by Example 1.13 since it says that $(5,6) \in R$.

Definition 1.14. A *partition* of a set A is a collection of disjoint subsets of A whose union is equal to A.

Definition 1.18. An *equivalence relation* on a set A is a binary relation $\sim$ such that $\forall\, a, b, c \in A$,

- $a \sim a$
- $a \sim b$ iff $b \sim a$
- if $a \sim b$ and $b \sim c$, then $a \sim c$

Each are called reflexivity, symmety, and transitivity.

- Note that each partition of a set gives rise to an equivalence relation and each equivalence relation gives rise to a partition. Therefore, there is a one-to-one correspondence between the ways to partition a set and the equivalence relations on it.
- I don't fully understand definition 1.21, however, it doesn't seem too important

Definition 1.30. A *preorder relation* on a set X is a binary relation on X, here denoted with infix $\leq$ such that

(a) $x \leq x$

(b) if $x \leq y$ and $y \leq z$, then $x \leq z$

These are called *reflexivity* and *transitivity*. Also, if $x \leq y$ and $y \leq x$, then we say $x \cong y$ and that x and y are equivalent. A pair $(X, \leq)$ is called a preorder

- *Discrete preorders* are preorders of the form $(X, =)$, so every item is only comparable with itself and nothing else. A **Codiscrete preorder** is the opposite where every two items are related by the binary relation, both ways. (A preorder does not have to be symmetric)
- A *Partial order* is a preorder with the additional condition
  (c) $x \cong y \Rightarrow x = y$
- A *Total order* is a partial order where any two elements are comparable.
- Haase diagrams are essentially graphs where the nodes are the elements, and the edges are the relations between them. We have an edge $a \in A$ with source and target functions defined as $s(a) = v$ and $t(a) = w$ given that a is an edge from v to w. Then, we have that $v \leq w$ if the binary relation is denoted by $\leq$.
- Partitions have a notion of "fineness". We say that a partition P is finer than another Q if for every $p \in P$, $\exists\, q \in Q$ such that $A_p \subseteq A_q$.
- In Exercise 1.53, we apply that paritions can be expressed as surjective functions. Given some surjective function, the preimage of each element in the range is a partition of the domain.
- A product preorder is when you have a product of two sets and you can say that $(x, y) \leq (x', y')$ iff $x \leq x'$ and $y \leq y'$.
- An opposite preorder is denoted as $\leq^{op}$ and is simply the reverse of the original preorder.

Definition 1.59. A *monotone map* between preorders $(A, \leq_A)$ and $(B, \leq_B)$ is a function $f : A \to B$ such that $\forall\, x, y \in A, x \leq_A y \Rightarrow f(x) \leq_B f(y)$

- Visually, monotone maps can be thought of as mappings between Hasse diagrams where the lines do not "criss cross".
- For finite sets, the power set has a natural monotone map to the natural numbers. So, the number of each elements in a subset is mapped to its cardinality.
- I am a little stuck on Example 1.64. It stems from a misunderstanding of upper sets as presented in Example 1.54. Reading the definition of an upper set again, I realize my confusion
- So an **upper set** is a set where, if an element exists in the set, so does any element greater than it. $U(P)$ denotes all upper sets that can be formed from P. I thought that $U(P)$ was eqivalent to an upper set, but they are two different things.
- Now, Example 1.64 is obvious since $U(P)$ has the same binary relation as the power set.
- I got exposed to the Yoneda lemma in Exercise 1.66, but I honestly do not see the importance of it.

1.3 Meets and joins

Definition 1.81. For a preorder $(P, \leq)$, we say $p \in P$ is a *meet* of A if

(a) $\forall\, a \in A, p \leq a$

(b) $\forall\, q \leq a, \forall\, a \in A$, we have that $q \leq p$

Alternatively, we can write $p = \bigwedge A$. Because it is a meet of all elements of A. This is essentially a redefinition of infimum.

Similarly, p is a *join* of A if

(a) $\forall\, a \in A, a \leq p$

(b) $\forall\, q$ such that $a \leq q, \forall\, a \in A$, we have that $p \leq q$

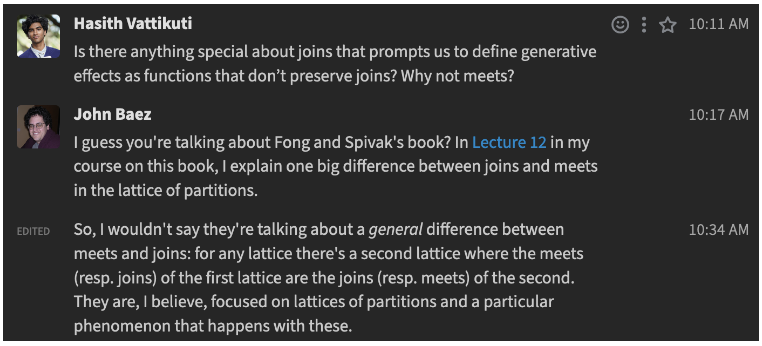
Again, we can rewrite this as $p = \bigvee A$ since it is the join of all elements in A, and this corresponds to the supremum of a set.

- Remark 1.82 is an interesting thought. I can't think of any examples as of now because I'm still thinking of meets and joins as infimums and supremums. Actually, if I think about it as a Haase diagram, if there are multiple "upper" and "lower" level elements, then there can be non-unique meets and joins.
- My above idea would not work in all cases. For example a discrete ordering has no meet and join.
- In Example 1.84, there is an good sample scenario of when two non-equal elements c and d are congruent. Also, it shows the correct case of when there can be multiple meets, and multiple joins are easily imagined.
- Multiple meets or joins are always congruent to each other
- Congruent but not equal elements are giving me a little insight into the "non-equal elements are treated as equals statement"
- join, $\bigvee \to \sup$
- meet, $\bigwedge \to \inf$

Definition 1.92. A monotone map $f : P \to Q$ *preserves meets/joins* if $f(a \wedge b) \cong f(a) \wedge f(b) \;\; \forall\, a, b \in P$ (the meet gets replaced by join for join preservation).

Definition 1.93. A monotone map has a *generative effect* if it does not preserve joins.

- A monotone map $\Phi : P \to Q$ is a phenomenon of P observed by Q.
- In the case of a generative effect, we are observing something 'unexpected' that we can't understand just by combining our observations of $f(a)$ and $f(b)$–we need information about the connection between sets P, Q.
- It seemed strange that generative effects aren't also defined when meets aren't preserved, but the last few paragraphs of 1.3 makes me think that it has essentially the same effect. Here is what John Baez had to say about it:



- For any monotone map $f : P \to Q$, if $a, b \in P$ have a join and so do $f(a), f(b)$, then assuming WLOG $a \leq b$:

$$\begin{aligned} a \leq b \leq a \vee b &\Rightarrow f(a) \leq f(b) \leq f(a) \vee f(b) \\ &\Rightarrow f(a) \vee f(b) \leq f(a \vee b) \end{aligned}$$

1.4 Galois connections

Definition A. *Galois connection* between preorders P and Q is a pair of monotone maps $(f : P \to Q, g : Q \to P)$ such that $\forall\, p \in P, q \in Q$,

$$f(p) \leq q \Leftrightarrow p \leq g(q)$$

We say that f is the *left adjoint* of g and g is the *right adjoint* of f.

- Essentially, a Galois connection is a pair of Hasse diagrams with mappings between them such that the lines do not criss-cross between the two diagrams.
- According to Example 1.97, taking the map $3 \times -$, if we treat is as a right adjoint, then the corresponding left adjoint is is $\lceil -/3 \rceil$
- I don't really understand the difference between a left adjoint and a right adjoint. For example, why would the right adjoint of the map in Example 1.97 not be the same as the left adjoint?
- To find a right adjoint g, the following must be true:

$$3x \leq y \Leftrightarrow x \leq g(y)$$

  for $x \in \mathbb{Z}$ and $y \in \mathbb{R}$. If your draw out the Haase diagrams and the left adjoint, it becomes clear that the right adjoint is $\lfloor y/3 \rfloor$. Still, I do not fully intuitively see this without the Hasse diagram. I should do more exercises.
- An explanation of what happened: We had a Galois connection and we were asked to find the right adjoint of a right adjoint. Visually, we had a mapping between two Hasse diagrams that did not criss cross, and then we removed all the left adjoint lines, moved the left Hasse diagram to the right hand side of the right diagram, and then reflected the right diagram–including its mapping lines–over the y-axis, and tried to find a suitable right adjoint for that. It is clear that the original left adjoint would not work because we would need to somehow reverse the cieling function, which is not possible.

- To be more specific, Hasse diagrams with maps that don't criss cross is not the only requirment. The left adjoint can't map multiple elements "up" and the right adjoint can't map multiple elements "down"

- The previous observation is better summed up in remark 1.100

- To show a function $g : S \to T$ induces a Galois connection $g_! : \text{Prt}(S) \leftrightarrow \text{Prt}(T) : g^*$ between preorders of the partitions of S and T, we show the following

- Start with a partition $\sim_S$ of S and to obtain $\sim_T$ on T, then two elements $t_1, t_2 \in T$ are in the same part $t_1 \sim_T t_2$ if there exists $s_1, s_t \in S$ such that $s_1 \sim_S s_2$ and $g(s_1) = t$ and $g(s_2) = t_2$

- Still, it is possible that we get $t_1 \sim t_2$ and $t_2 \sim t_3$ without $t_1 \overset{?}{\sim}_T t_3$ even though partitions must be transitive. So, we need to take the transitive closure which results in $g_!$.

- The reason we take the transitive closure is because $\sim_S$ and sim_T should be equivalence relations, which means they must be transitive. So, it is only natural to "force" g to be transitive from $S \to T$ by taking the transitive closure.

Proposition 1.107. If $f : P \to Q$ and $g : Q \to P$ are monotone maps, then the following are equivalent:

1. f and g form a galois connection and f is a left adjoint of g

2. For all $p \in P, q \in Q$ we have that $p \leq g(f(p))$ and $f(g(q)) \leq q$

- The above is an easy exercise. Note that replacing $\leq$ with $\cong$ gives us the definition of an isomorphism, so we can think of a Galois connection as a "weaker" isomorphism.

Definition 1.75. Let $(P, \leq_P)$ and $(Q, \leq_Q)$ be preorders. A monotone function $f : P \to Q$ is an *isomorphism* if there exists a monotone function $g : Q \to P$ such that $f \circ g = \text{id}_Q$ and $g \circ f = \text{id}_P$.

- According to exercise 1.110, the left and right adjoints to a monotone function are unique up to an isomorphism.

Proposition 1.111. Right adjoints preserve meets and left adjoints preserve joins: if $f : P \to Q$ and $g : Q \to P$ form a Galois connection, then

$$g\left(\bigwedge A\right) \cong \bigwedge g(A)$$

$$f\left(\bigvee A\right) \cong \bigvee f(A)$$

where A is any subset of P or Q.

Proof

Define $m = \bigwedge A$, then since $m \leq a$ for all $a \in A$ by definition, it follows that $g(m) \leq g(a) \ \ \forall a \in A$ because g is monotone.
Then, it suffices to show that $g(m)$ is the greatest lower bound for $g(A)$. For some b such that $b \leq g(a) \ \ \forall a \in A$, by the Galois connection then $f(b) \leq a$. So, $f(b)$ is a lower bound for a, but since m is the greatest lower bound, it follows that $f(b) \leq m$. Then, $b \leq g(m)$, therefore $g(m)$ is the greatest lower bound for $g(A)$. The joins are preserved similarly.

- Recall that we have a generative effect when a map does not preserve joins. Since left adjoints preserve joins, they do not have generative effects
- We will find that a monotone map does not have generative effects–it preserves joins–iff it is a left adjoint to some other monotone.

Theorem 1.115. (Adjoint functor theorem for preorders) Suppose P and Q are preorders and Q has all meets. A monotone map $g : Q \to P$ preserves meets iff it has a right adjoint. If instead P has all joins and Q is any preorder, a monotone map $f : P \to Q$ preserves joins iff it is a left adjoint.

Proof

We already showed that right adjoints preserve meets, so to prove the iff we only need to show that if g preserves meets, then it has a right adjoint:
Suppose g is a monotone map that preserves meets. Then, define $f : P \to Q$ by

$$f(p) = \bigwedge\{q \in Q : g(q) \leq p\}$$

We need to show that f is a right adjoint to g. First, it is seen that f is well defined since Q contains all meets. Second, we see that

$$f(p) \leq q \Leftrightarrow p \leq g(q)$$

by the definition of a meet. So now we need to show that f is monotone. Suppose $p \leq p'$, then $\{q' \in Q \mid p' \leq g(q')\} \subseteq \{q \in Q \mid p \leq g(q)\}$. So, this leads us to say that $f(p) \leq f(p')$, thus f is monotone.
The second statement is proved in a similar manner.

- The proof of the monotonicity of f is accurate, but I feel like I might have made some logical leaps in proving that it is also a left adjoint. The proof listed in the textbook is more thorough
- In example 1.117, I am a little confused by what is meant by 'the' left adjoint and 'the' right adjoint. We saw that the adjoints were unique wrt to each other upto an isomorphism, but that surely can't mean that we are guaranteed that $f_!$ and f_* are a galois connection, right?

Exercise 1.119. Part 1 is trivial. Part 2 goes as follows:
Let $p' = (f \fatsemi g)(p)$, and by proposition 1.107,

$$(f \fatsemi g)(p) \leq (f \fatsemi g \fatsemi f \fatsemi g)(p)$$

I am a little stuck on the other inequality, so I will reference the solution.
Next, letting $q = f(p)$, the same proposition gives us

$$f(g(f(p))) \leq f(p) \Rightarrow g(f(g(f(p)))) \leq g(f(p))$$

Which is

$$(f \fatsemi g \fatsemi f \fatsemi g)(p) \leq (f \fatsemi g)(p)$$

Definition 1.120. A *closure operator* $j : P \to P$ on a preorder P is a monotnoe map such that for all $p \in P$ we have

(a) $p \leq j(p)$

(b) $j(j(p)) \cong j(p)$

- We can define preorders on the set of relations on a set as well as the set of preorders relations on a set.
- A preorder of preorders in a level shift.