Contents

1 Generative Effects: Orders and Galois Connections

2 Resource Theories: Monoid Preorders and Enrichment

2.1 TODO $\mathcal{V}$ Enrichment

2.2 Symmetric Monoidal Preoder

Preorder $(X, \leq)$: Set X with a reflexive $x \leq x$ and transitive $x \leq y \wedge y \leq z \Rightarrow x \leq z$ preorder relation $\leq$.

Partially ordered: preorder with skeletality requirement ($x \cong y \Rightarrow x = y$; equivalence implies equality) where $x \leq y \wedge y \leq x \Rightarrow x \cong y$.
Also: poset - partially ordered set. Every preorder can be made into partial order by adding the skeletality requirement.

Monoid $(X, \otimes, I)$: Set X, monoidal product (i.e. multiplication) $\otimes : X \times X \to X$, monoidal unit element $I \in X$. Conditions:

- unitality: $\forall\, x \in X : I \otimes x = x \otimes I = x$
- associativity: $\forall\, x, y, z \in X : (x \otimes y) \otimes z = x \otimes (y \otimes z)$

Commutative Monoid $(X, \otimes, I)$ Note: matrix multiplication is not commutative.

- commutativity: $\forall\, x, y \in X : (x \otimes y) = y \otimes x$

Monoidal Preorder: Condition:

- monotonicity: $x_1 \leq y_1 \wedge x_2 \leq y_2 \Rightarrow x_1 \otimes x_2 \leq y_1 \otimes y_2$

Symmetric: condition:

- symmetry: $x \otimes y = y \otimes x$

Closed: $\mathcal{V} = (V, \leq, I, \otimes)$ is symmetric monoidal closed: $(a \otimes v) \leq w \Leftrightarrow a \leq (v \multimap w)$ where $(v \multimap w)$ is the *hom-element*.

2.3 Quantales

Unital Commutative Quantale: Symmetric Monoidal Closed Preorder $\mathcal{V} = (V, \leq, I, \otimes, \multimap)$

3 Databases: Categories, Functors and Universal Constructions

Free category on a graph $G = (V, A, s, t)$: $\mathbf{Free}(G)$:
Generated by a graph with sets of Vertices and Arrows, source and target functions.

- objects are vertices V
- morphisms are paths from c to d
- identity morphism on an object c is a trivial path at c
- morphism composition - concatenation of paths

Morphisms of $\mathbf{Free}(G)$ are exactly the paths in G and they form the *closure* of the set of Arrows A. $\mathbf{Free}(G)$ is a category that in a sense contains G and obeys no equations other than those that categories are forced to obey.

4 Collaborative Design: Profunctors, Categorification, and Monoidal Categories

4.1 Chapter 4, lecture 1 (Spivak)

Reference to Andrea Censi; CoDesign = Collaborative design; Functionalities - resources provided vs. resources required

Feed back loop - compact closure

Pareto optimal front

Preorder $(P, \leq)$ velocity v and weight w are preorders; $v \times w$ is also a preorder;

$v \times w$ is not a linear preorder anymore; certain thing are neither worse nor better than the other things

antichain: subset A of P: $A \subseteq P$ such that (s.th.) for all $a1, a2$ from A if $a1 \leq a2$ then $a1 = a2$. IOW no two different things are comparable.

Categorical idea: $\mathcal{V}$-*profunctors* = feasibility relationships especially if $\mathcal{V}$ is **Bool**.

$\mathcal{V}$-*category* is a diagram where by the elements of $\mathcal{V}$. $\mathcal{V}$ knows how to compose by what's called tensor.

$\mathcal{V}$-*profunctor* of **Bool**: "Can I get a motor that can provide this much torque and speed for this much weight, current and voltage?"

$\mathcal{V}$-*profunctor* of **Cost**: "How much would it cost to get a motor that can providing this much torque and speed for this much

$\mathcal{V}$-*profunctor* of **Set**. see 8:11 what are the ways to

Idea: (wire diagrams)
$\mathcal{V}$-*category*: wires - each wire is carrying a preorder
$\mathcal{V}$-*profunctor*: boxes
$\mathcal{V}$-*profunctor-composition*: whole design problem; composition = feed-forward co-design.
compact closed structure: add feedback

$\mathcal{V}$ is **Bool**:
$\mathcal{V}$-*category* is a preorder: Less than or equal to is a true/false question.

Opposite of a $\mathcal{V}$-*category* P:
A $\mathcal{V}$-*category* w/ the same objects, arrows are reversed. I.e. if $p' \leq p$ in P then $p \leq p'$ in P^{op}.

4.1.1 $\mathcal{V}$-*profunctor*: from one category to another

14:19 $\mathcal{V}$-*profunctor*: $P \nrightarrow Q$ is A $\mathcal{V}$-*functor* : $P^{\text{op}} \times Q \to \mathcal{V}$ between $\mathcal{V}$-*categories* P and Q.

In Hasse diagram P and Q are wires and $\nrightarrow$ is a box Φ (phi-easibility).

$\mathcal{V}$ is a Symmetric Monoidal Poset (i.e. a Symmetric Monoidal Category where the Category is a Poset) equipped with:

1. Notion of object: has a set of objects $Ob(P)$
2. Notion of element: for all $p1, p2 \in Ob(P)$ we have $P(p1, p2) \in \mathcal{V}$

Symmetric Monoidal Preorder (i.e. a Symmetric Monoidal Category where the Category is a Preorder; Poset is a Preorder with skeletality requirement) i.e. a Category where the morphism are "easy", i.e. between any two objects there either is one or isn't one morphism. I.e. only one or none morphism.

Conditions for:

1. monoidal unit $I \leq_{\mathcal{V}} P(p, p)$
2. monoidal product $P(p1, p2) \otimes P(p2, p3) \leq_{\mathcal{V}} P(p1, p3)$

$P = (\mathcal{V}, \otimes, I)$ is a $\mathcal{V}$-*category* - it means it is enriched in itself. That also means it's a quantale, and that means it has all joins. $\mathcal{V}$ is also a symmetric monoidal preorder with joins that distribute over tensor. i.e. a quantale. 43:40

$\mathcal{V}$-*profunctor*: $P^{\text{op}} \times Q \to \mathcal{V}$ where $\mathcal{V} = \{\mathit{true}, \mathit{false}\}$ is a boolean.

$\nrightarrow$ is a profunctor, $\to$ is a normal functor. IOW _ $\nrightarrow$ _ packages up _ $\to$ _

<u>Unpacking $\Phi(p, q)$</u>: is p feasible, given q?
p - resources provided
q - res/ources required

<u>Meaning of opposite $^{\text{op}}$</u>: is there a path?"
Can you give me a dinner for two p? - Yes that's feasible. Actually I need just a dinner for one p':

if $p' \leq p$ and $q' \leq q$ then $\Phi(p, q) \leq \Phi(p', q')$

Bool-*profunctor* drawn in a form of collage. Like a Hasse diagram for the whole profunctor.

Profunctor: a generalisation of functor where not everything from the domain has to be included and two things may be spread out. See page 7Sketches.pdf, page 122. Also: Every functor is a kind of profunctor.

Monotone map: order preserving function $f : x \leq y$ then $f(x) \leq f(y)$

A functor between **Bool** categories is a monotone map. So any monotone map is a profunctor. $\mathbb{N}$ are natural number with $\leq$ and $+$ relations / operations. $\mathbb{N} \times \mathbb{N} \xrightarrow{+} \mathbb{N}$.

Whenever some says a "functor", "category", "profunctor" w/o mentioning the $\mathcal{V}$ they always mean a **Set**-*category* or a **Set**-(*pro*)*functor* Note: **Set** is a monoidal category.

4.1.2 Profunctor composition

Composing Φ with Ψ and asking if it is feasible means that we can find some $q \in Q$, such that:

$$(\Phi;\ \Psi)(p, r) = \bigvee_{q \in Q} \Phi(p, q) \wedge \Psi(q, r)$$

where Φ, Ψ are boolean feasibilities and $\wedge$, $\bigvee$ are AND and OR in **Bool**.

Identity on P:

$$id_P : P^{\text{op}} \times P \to \mathcal{V}$$

where $\mathcal{V}$ is **Bool**

$$id_P(p, p') := P(p, p')$$

For any category that category is it's own profunctor.

Andrea Censi passes around the pareto optimal anti-chains

4.2 Chapter 4, lecture 2 (Fong)

Collaborative design problem asks for: Given a set of specifications of teams what can the team as a whole produce?

Hasse diagram is intuitive but also formal at the same time. It also provide a particular algorithm how do we compute the entire capability of the team. How this team can collaborate to design some product.

4.3 Symmetric Monoidal Categories SMC

- Preorder $(P, \leq)$; e.g. $1 \leq 2$; P is the wires, $\leq$ is the boxes/series
- Monoid $(M, *, e)$ e.g. string of processes $(1+2)+3$; M is the boxes, $*$ is series of composition.
- Monoidal Preorder $(P, \leq, *, e)$ where P is a set. We can put things in parallel (wires, boxes, parallel boxes)

- Category: generalization of Monoid and Preorder $(Ob(\mathscr{C}), Mo(\mathscr{C}), \fatsemi, id)$: (wires, boxes, series)
- Monoidal Preorder and Category are special types of Preoder and Monoid 4:30
- Monoidal Category: special type of Monoidal Preoder and Category $(\mathscr{C}, \otimes, I)$ - (-, parallel, -)

Axioms - ways to ensure that Hasse diagrams have unambiguous interpretation associativity.

Symmetric Monoidal Category SMC $(\mathscr{C}, \otimes, I)$ consists of a:

- Category $\mathscr{C}$
- Functor for monoidal product $\otimes : \mathscr{C} \times \mathscr{C} \to \mathscr{C}$
- Functor I: $\mathbf{1} \to \mathscr{C}$ i.e. an object $I \in Ob(\mathscr{C})$
- Natural Isomorphism:
  - $\lambda_X : I \otimes X \to X$ i.e. left unitor
  - $\rho_X : X \otimes I \to X$ i.e. right unitor
- Associativity: $\alpha_{X,Y,Z} : (X \otimes Y) \otimes Z \to X \otimes (Y \otimes Z)$

SMC: $\delta_{X,Y} : X \otimes Y \to Y \otimes X$ i.e. swap map (symmetricity)
SMC is this data such that the natural isomorphisms are well behaved

Tensor product $f \otimes g$ - parallel "execution" of f and g

SMC Examples:

1. $(\mathbf{Set}, \times, \mathbf{1})$: underlying **Set** category is the category of all sets: objects are sets, morphisms are functions; monoidal product $\times$ is a product of sets and product of functions. See 27:38
2. $(\mathbf{Set}, \sqcup, \varnothing)$: $\sqcup$ is the coproduct of disjoint unional sets.
3. $(\mathbf{Vect\_k}, \otimes, k)$: k is a field; objects are vector spaces; monoidal product $\otimes$ i.e. monoidal structure comes from the tensor product of linear maps and vector spaces
4. $(\mathbf{Prof}_{\mathcal{V}}, \times, \mathbf{1})$: category of profunctors; objects are $\mathcal{V}$-*categories* for some symmetric monoidal preorder; morphisms are the profunctors; monoidal product $\times$ is product of $\mathcal{V}$-*categories*.

4.4 Categorification

Take a known thing and add structure to it. So that properties become structures. See 7Sketches.pdf, page 133.

FinSet: Category of finite sets and functions.

Example - Categorify $\mathbb{N}$ using **FinSet** :

- replace every number with a set of that many elements.
- replace + with disjoint union of sets $\sqcup$.
- replace equality with the structure of an isomorphism.

5 Signal Flow Graphs: Props, Presentations and Proofs

5.1 Chapter 5, lecture 1 (Spivak)

Signal Flow Graphs - used in amplifiers filter, cyber-physical systems (tightly interacting physical and computational parts)

I.e. It makes sense over any **Rig** which is basically a **Ring** : $R[s, s^{-1}]$

Prop $(\mathscr{C}, \otimes, I)$: Special kind of a monoidal category where the objects are "easy" such that:

- $Ob(\mathscr{C}) := \mathbb{N}$
- $I := 0$
- $\forall\, m, n \in Ob(\mathscr{C}) := \mathbb{N} : m \otimes n := m + n$

I.e. *Prop* is a SMC where objects just have some finite cardinality. They're just numbers (i.e. lines)

Example:
$PropMat_{\mathbb{R}}$ of matrices over a **Rig** $\mathbb{R}$; in this case real numbers $\mathbb{R}$. A **Rig** is an algebraic object where you can add and multiple things. I.e.

- $Ob(Mat_{\mathbb{R}}) := \mathbb{N}$
- $Mat_{\mathbb{R}}(m, n) := Mat_{\mathbb{R}}(m, n)$ - can't distinguish between the notations.

Compose an tensor of two matrices:

Presented Prop

String Diagrams (Syntax and Semantics, Soundness and Completeness)

7:13 String diagrams are syntax for something, Semantics is the math formula with integrals

Soundness: if you can prove that one diagram equals to another using String diagram manipulations

Prop Functor

Port Graph

5.2 Chapter 5, lecture 2 (Fong)

Free structure: free from unnecessary constraints. See $\textbf{Free}(G)$

Transitive closure R^+ of a binary relation R:
Example: $R = \{(1,2),(2,3)\}$ then $R^+ = \{(1,2),(2,3),(1,3)\}$ i.e. extend the R by every possible composition.