# Boston Comp. Club Meeting #1
# Programming from Gallois Connections

Max von Hippel

## Table of Contents



### ① Motivation

Programming problems are often of the form

"solve this, under this constraint"

How can we formalize this notion of notions?
We will use Gallois connections to define an operator $\upharpoonright$, s.t. we can write

$$\text{solve this} \upharpoonright \text{under this condition}$$

### ② Gallois Connections

Def A "Gallois Connection" is a pair of functions $f : A \to B$ and $g : B \to A$, and preorders $\leq$ on $B$ and $\sqsubseteq$ on $A$, such that:

$$f(a) \leq b \iff a \sqsubseteq g(b)$$

Coming up next: EXAMPLES

$\boxed{\text{2-continued}}$ examples of Gallois connections

(a) division vs multiplication for $a, b, c > 0$

$$\underbrace{a \times c}_{f} \le b \iff a \le \underbrace{b \div c}_{g}$$

(b) addition vs subtraction

$$\underbrace{a + c}_{f} \le b \iff a \le \underbrace{b - c}_{g}$$

(c) length vs take

$$\text{length } z \le n \wedge z \sqsubseteq x \iff z \sqsubseteq \text{take}(n, x)$$

let $S$ be the set of finite-length strings and $\mathbb{N}$ the set of string-lengths.

let $\le$ be the normal order on $\mathbb{N}$.
let $\sqsubseteq$ be the prefix order on $S$.
let $\trianglelefteq$ be the preorder on $\mathbb{N} \times S$ defined by
$(n, x) \trianglelefteq (m, y) \iff n \le m$ and $x \sqsubseteq y$.
let $f : S \to \mathbb{N} \times S$ be the function $x \mapsto (\text{length } x, x)$
let $g : \mathbb{N} \times S \mapsto S$ be the function
$(n, x) \mapsto x[0 : n-1]$ returning the $n$-length prefix.

Then the Gallois Connection is $(f, g, \trianglelefteq, \sqsubseteq)$.

Cool, right? $\longrightarrow$

③ Consider the following Galois relation:

$$f(x) \leq y \iff x \sqsubseteq g(y)$$

express $\leq$ and $\sqsubseteq$ as relations:

$$(f(x), y) \in (\leq) \iff (x, g(y)) \in (\sqsubseteq)$$

express $f$, $g$ as relations

$$(x, y) \in f^{\circ} \cdot (\leq) \iff (x, y) \in (\sqsubseteq) \cdot g$$

$x, y$ are free variables, so express in point-free form

$$f^{\circ} \cdot (\leq) = (\sqsubseteq) \cdot g$$

Split relational equality into two subset statements

$$\underbrace{f^{\circ} \cdot (\leq) \subseteq (\sqsubseteq) \cdot g}_{(\mathrm{I})} \quad \text{and} \quad \underbrace{(\sqsubseteq) \cdot g \subseteq f^{\circ} \cdot (\leq)}_{(\mathrm{II})}$$

---

Assume $f$ and $g$ are monotonic. Then:

$(\mathrm{II}) \Rightarrow \{$ as $\sqsubseteq$ is a preorder $\}$

$$g \subseteq id \cdot g \subseteq (\sqsubseteq) \cdot g \quad \text{so} \quad g \subseteq f^{\circ} \cdot (\leq)$$

$\Rightarrow \{$ by monotonicity of $\circ \}$

$$(\sqsubseteq) \cdot g \subseteq (\sqsubseteq) \cdot f^{\circ} \cdot (\leq)$$

$\Rightarrow \{$ by monotonicity of $f \}$

$$(\sqsubseteq) \cdot g \subseteq f^{\circ} \cdot (\leq) \cdot (\leq)$$

$\Rightarrow \{$ by associativity of $\leq$, noting therefore $(\leq) \cdot (\leq) \subseteq (\leq)\}$

$$\boxed{(\sqsubseteq) \cdot g \subseteq f^{\circ} \cdot (\leq)}$$

---

Note also that $(\mathrm{I}) = f^{\circ} \cdot (\leq) \subseteq (\sqsubseteq) \cdot g$

$\Rightarrow \{$ taking converses of both sides $\}$

$$(f^{\circ} \cdot (\leq))^{\circ} \subseteq ((\sqsubseteq) \cdot g)^{\circ} = g^{\circ} \cdot (\sqsupseteq)$$

$\Rightarrow \{$ shunting $g^{\circ}$ to the left $\}$

$$\boxed{g \cdot (f^{\circ} \cdot (\leq))^{\circ} \subseteq (\sqsupseteq)}$$

Finally:

$$\boxed{f^\circ \cdot (\leq)} = (\sqsupseteq) \circ g$$

$$\Longleftrightarrow$$

(a) $g \subseteq \boxed{f^\circ \cdot (\leq)}$ The "easy" part

and

(b) $g \circ (\boxed{f^\circ \cdot (\leq)})^\circ \subseteq (\sqsupseteq)$ The "hard" part

Call this part → "S".

Then (a) means $f$ must return a result permitted by $S$.

(b) means that if $S$ maps $x$ to $y$, then $gx \sqsupseteq y$, meaning, $g$ must return a maximum result.

Definition: Given relations

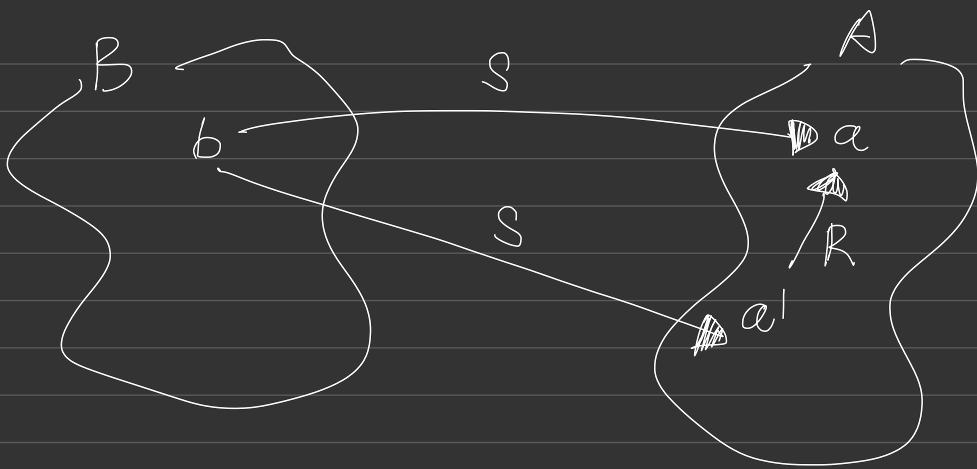
$$\begin{tikzcd} B \arrow[r, "S"] & A \arrow[loop right, "R"] \end{tikzcd}$$

Define "$S$ shrunk by $R$" to be

$$S \upharpoonright R = S \cap R / S^\circ$$

i.e, $g \subseteq (f^\circ \cdot (\leq)) \upharpoonright (\sqsupseteq)$

$S \wedge R$ takes $b \in B$ to $a \in A$ such that:




This is a very topological idea.

We are shrinking S to only perform maps that are unmolested by R.

If R is a maximizing function, then we are shrinking S to only accept maximal solutions.

④ What can we do with it?

TODO: Cave Scheduling

PROBLEM: Let $A$ be a set of tasks;

for each $a \in A$, let $g(a)$ be the set of tasks that must wait for $a$ to complete before they may begin;

let spans: $A \to \mathbb{N}$ give the timespan of each task;

and let schedule: $A \to \mathbb{N}$ give starting times to tasks.

---

Given $g$, the problem is to compute a function

$$f_g : \text{spans} \to \text{schedule}$$

that gets all the tasks done as quickly as possible.

Let $lazy_g$ : schedule $\to$ spans compute for each task in the schedule its max. execution time.

So, $\sum_{s \in lazy_g(\text{schedule})} s$ = max time the schedule could take to execute.

---

$$\Rightarrow \quad f_g = ((lazy\ g)^{\circ} \cdot (\geq)) \upharpoonright (\leq)$$

So what is the converse of $(lazy\ g)$?

(( this is where I'm lost ))

(c) length vs take

$$\text{length } z \le n \wedge z \sqsubseteq x \iff z \sqsubseteq \text{take}(n, x)$$

let $\lhd$ be $\le \times \sqsubseteq$.

$$(\text{length } z, z) \lhd (n, x) \iff z \sqsubseteq \text{take}(n, x)$$

let $f(z) = (\text{length } z, z)$

$$f(z) \lhd (n, x) \iff z \sqsubseteq \text{take}(n, x)$$

rewrite using $\upharpoonright$ operator

$$\text{take} \subseteq (f^{\circ} \circ (\lhd)) \upharpoonright (\sqsupseteq)$$

rewrite according to def of $f$

$$\text{take} \subseteq (\text{length}^{\circ} \circ (\le), \text{id}^{\circ} \circ (\sqsubseteq)) \upharpoonright (\sqsupseteq)$$

obviously $\text{id}^{\circ} = \text{id}$ so

$$\text{take} \subseteq (\text{length}^{\circ} \circ (\le), (\sqsubseteq)) \upharpoonright (\sqsupseteq)$$

so, take$(n, x)$ is some $z$ such that

(easy part): $\text{length}(z) \le n$ and $z \sqsubseteq x$

(hard part): $z$ is longest possible.

The calculational aspect is to synthesize $\text{length}^{\circ}$.