

“How much is in a square?” An appraisal of relational thinking

IFIP WG2.1 meeting #81

Kloster Neustadt, Germany, 4th April 2024

J.N. OLIVEIRA



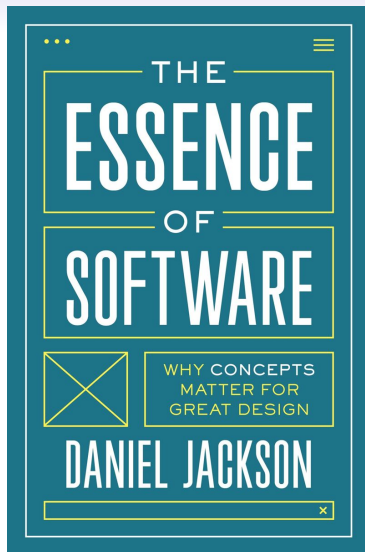
UNIVERSITY OF MINHO & INESC TEC

Ageing

Quote from *Du und die Musik. Eine Einführung für alle Musikfreunde* by Frederich Herzfeld, Berlin, 1951:

*"The young are to the **content** what the elderly are to the **continent**".*

More and more interested in simple **patterns** (**shapes**, "continents") easy to **use** and **communicate** to a **wider audience**.



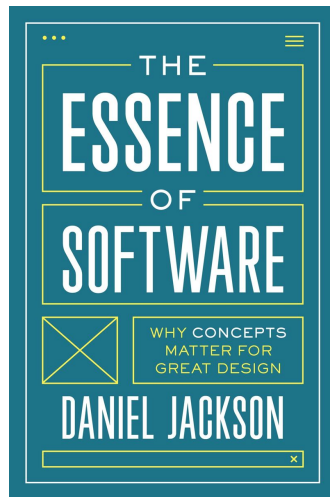
The Essence of Software

"(...) The best services revolve around **a small number of concepts** that are **well designed and easy** (...) **to understand and use**, and their innovations often involve simple but compelling new concepts."



The Essence of Software

- ... **small number**
- ... **well defined**
- ... **easy to understand**
- ... **easy to use**



Graduate Texts in Mathematics

Saunders Mac Lane

Categories for the Working Mathematician

Second Edition

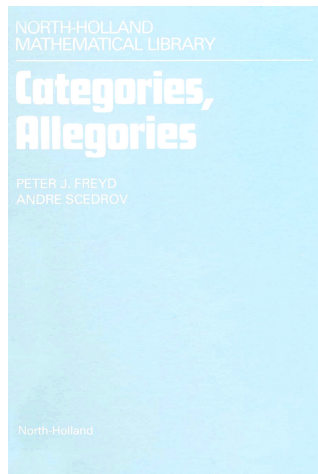


Springer

Freyd & Ščedrov, 1990

"(...) A special feature of our approach is a general **calculus of relations** presented in part two.

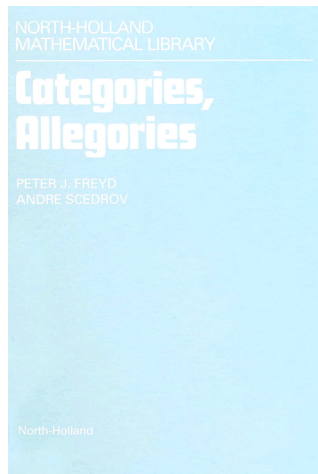
This calculus offers another, **often more amenable** framework for concepts and methods discussed in part one."



Freyd & Ščedrov, 1990

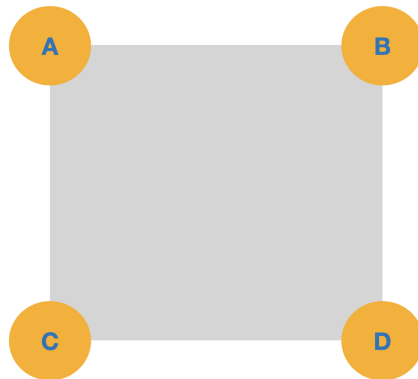
"(...) A special feature of our approach is a general **calculus of relations** presented in part two.

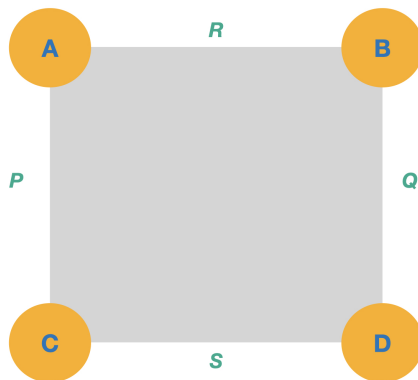
This calculus offers another, **often more amenable** framework for concepts and methods discussed in part one."

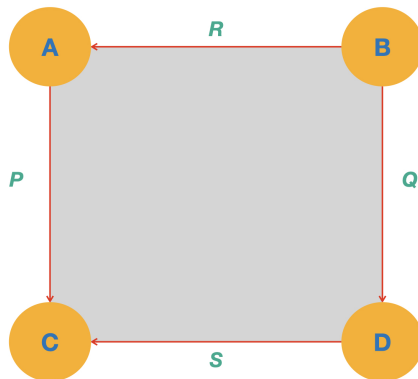


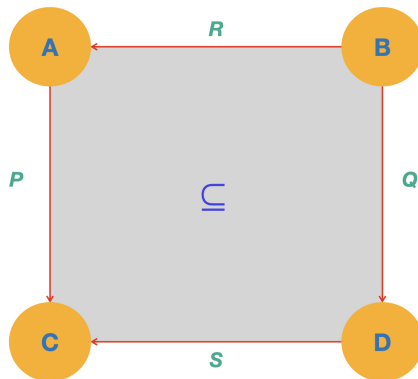
Squares



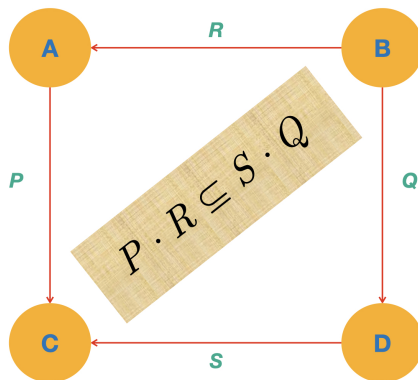






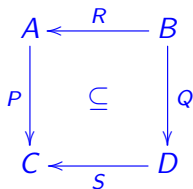


“Magic” square



“Magic” square

Four binary relations:



$$P \cdot R \subseteq S \cdot Q \quad (1)$$

Terminology:

R — *pre-producer*

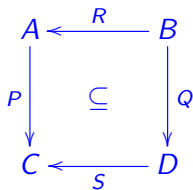
P — *pre-consumer*

Q — *post-producer*

S — *post-consumer*

“Magic” square

Pointfree:



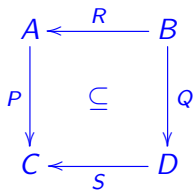
$$P \cdot R \subseteq S \cdot Q$$

Pointwise:

A pointwise diagram showing the equation $P \cdot R \subseteq S \cdot Q$. The variables a and d are green, b and c are red. The quantifiers \exists and \forall are red. The diagram shows the relationship between the variables and the quantifiers.

“Magic” square

Pointfree:



$$P \cdot R \subseteq S \cdot Q$$

Pointwise:

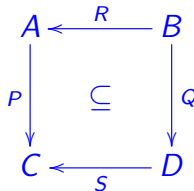
$$\begin{array}{c} \exists \\ \quad a \\ \quad \vdots \\ P \cdot R \\ \quad \vdots \\ \forall \quad c \quad b \end{array} \Rightarrow \begin{array}{c} \quad d \\ \quad \vdots \\ S \cdot Q \\ \quad \vdots \\ \quad c \quad b \end{array}$$

“Magic” square

1	14	14	4
11	7	6	9
8	10	10	5
13	2	3	15

Perhaps not what you were expecting...

... but we can do **a lot** with



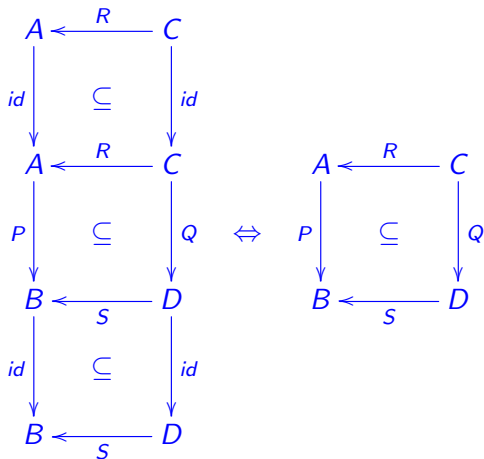
Vertical composition

$$\begin{array}{ccc}
 \begin{array}{ccc}
 A & \xleftarrow{R} & C \\
 P \downarrow & \subseteq & \downarrow Q \\
 B & \xleftarrow{S} & D \\
 P' \downarrow & \subseteq & \downarrow Q' \\
 B' & \xleftarrow{S'} & D'
 \end{array}
 & \Rightarrow &
 \begin{array}{ccc}
 A & \xleftarrow{R} & C \\
 P' \cdot P \downarrow & \subseteq & \downarrow Q' \cdot Q \\
 B' & \xleftarrow{S'} & D'
 \end{array}
 \end{array}
 \quad (2)$$

Horizontal composition

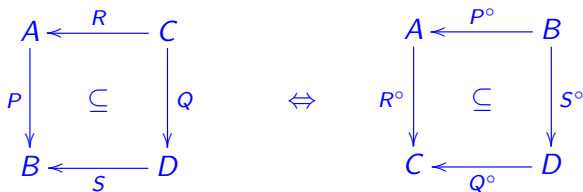
$$\begin{array}{ccccc}
 A & \xleftarrow{R} & C & \xleftarrow{R'} & C' \\
 \downarrow P & \subseteq & \downarrow Q & \subseteq & \downarrow Q' \\
 B & \xleftarrow{S} & D & \xleftarrow{S'} & D'
 \end{array}
 \Rightarrow
 \begin{array}{ccccc}
 A & \xleftarrow{R \cdot R'} & C & & \\
 \downarrow P & \subseteq & & & \downarrow Q' \\
 B' & \xleftarrow{S \cdot S'} & D' & &
 \end{array}
 \quad (3)$$

Identity



(Similarly for horizontal.)

Converse



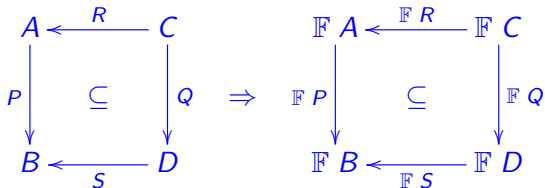
The converse of a square is its “*passive voice*” 😊

Square (direct) sums

$$\begin{array}{ccc}
 \begin{array}{ccc}
 A & \xleftarrow{R} & C \\
 \downarrow P & \subseteq & \downarrow Q \\
 B & \xleftarrow{S} & D
 \end{array} & + & \begin{array}{ccc}
 A' & \xleftarrow{R'} & C' \\
 \downarrow P' & \subseteq & \downarrow Q' \\
 B' & \xleftarrow{S} & D'
 \end{array} \\
 & = & \begin{array}{ccc}
 A + A' & \xleftarrow{R+R'} & C + C' \\
 \downarrow P+P' & \subseteq & \downarrow Q+Q' \\
 B + B' & \xleftarrow{S+S'} & D + D'
 \end{array}
 \end{array}$$

Functorial squares

Functor \mathbb{F} :



\mathbb{F} should be monotonic and preserve converses — a **relator** (Freyd and Scedrov, 1990).

Squares with functions

Some relations f fit into the following squares:

$$\begin{array}{ccccc}
 A & \xleftarrow{id} & A & \xleftarrow{f^\circ} & B \\
 id \downarrow & \subseteq & \downarrow f & \subseteq & \downarrow id \\
 A & \xleftarrow{f^\circ} & B & \xleftarrow{id} & B
 \end{array} \tag{4}$$

Left square: $\langle \forall a :: \langle \exists b :: b f a \rangle \rangle$ f is **total**.

Right square: $\langle \forall b, b' :: \langle \exists a :: b f a \wedge b' f a \rangle \Rightarrow (b = b') \rangle$

f is **univocal**.

Such relations f are called **functions**.

Squares with functions

Some relations f fit into the following squares:

$$\begin{array}{ccccc}
 A & \xleftarrow{id} & A & \xleftarrow{f^\circ} & B \\
 id \downarrow & \subseteq & \downarrow f & \subseteq & \downarrow id \\
 A & \xleftarrow{f^\circ} & B & \xleftarrow{id} & B
 \end{array} \tag{4}$$

Left square: $\langle \forall a :: \langle \exists b :: b f a \rangle \rangle$

f is **total**.

Right square: $\langle \forall b, b' :: \langle \exists a :: b f a \wedge b' f a \rangle \Rightarrow (b = b') \rangle$

f is **univocal**.

Such relations f are called **functions**.

Squares with functions

Some relations f fit into the following squares:

$$\begin{array}{ccccc}
 A & \xleftarrow{id} & A & \xleftarrow{f^\circ} & B \\
 id \downarrow & \subseteq & \downarrow f & \subseteq & \downarrow id \\
 A & \xleftarrow{f^\circ} & B & \xleftarrow{id} & B
 \end{array} \tag{4}$$

Left square: $\langle \forall a :: \langle \exists b :: b f a \rangle \rangle$

f is **total**.

Right square: $\langle \forall b, b' :: \langle \exists a :: b f a \wedge b' f a \rangle \Rightarrow (b = b') \rangle$

f is **univocal**.

Such relations f are called **functions**.

Squares with functions

Some relations f fit into the following squares:

$$\begin{array}{ccccc}
 A & \xleftarrow{id} & A & \xleftarrow{f^\circ} & B \\
 id \downarrow & \subseteq & \downarrow f & \subseteq & \downarrow id \\
 A & \xleftarrow{f^\circ} & B & \xleftarrow{id} & B
 \end{array} \tag{4}$$

Left square: $\langle \forall a :: \langle \exists b :: b f a \rangle \rangle$ f is **total**.

Right square: $\langle \forall b, b' :: \langle \exists a :: b f a \wedge b' f a \rangle \Rightarrow (b = b') \rangle$

f is **univocal**.

Such relations f are called **functions**.

Squares with functions

Let f be a **function**. Then:

$$\begin{array}{ccc}
 A & \xleftarrow{R} & C \\
 f \downarrow & \subseteq & \downarrow Q \\
 B & \xleftarrow{id} & B
 \end{array}
 \Leftrightarrow
 \begin{array}{ccc}
 A & \xleftarrow{R} & C \\
 id \downarrow & \subseteq & \downarrow Q \\
 A & \xleftarrow{f^\circ} & B
 \end{array}$$

This is the **shunting** rule

$$f \cdot R \subseteq Q \Leftrightarrow R \subseteq f^\circ \cdot Q \quad (5)$$

which, by taking converses, becomes:

$$R \cdot f^\circ \subseteq Q \Leftrightarrow R \subseteq Q \cdot f \quad (6)$$

“Nice” rules about functions

(Functional) **equality**:

$$f \subseteq g \Leftrightarrow f = g \Leftrightarrow g \subseteq f \quad (7)$$

Existential quantifiers go away:

$$b (f^\circ \cdot R \cdot g) a \Leftrightarrow (f b) R (g a) \quad (8)$$

$$\begin{array}{ccccc}
 B & \xrightarrow{f} & C & \xleftarrow{R} & D & \xleftarrow{g} & A \\
 & & \searrow & & \swarrow & & \\
 & & & f^\circ \cdot R \cdot g & & &
 \end{array}$$



Very useful in practice



Squares with functions

A very common square with 2 **functions**:

$$\begin{array}{ccc}
 A & \xleftarrow{R} & B \\
 f \downarrow & \subseteq & \downarrow g \\
 C & \xleftarrow{S} & D
 \end{array}
 \qquad f \cdot R \subseteq S \cdot g
 \qquad (9)$$

This square captures a **higher-order relation** on functions:

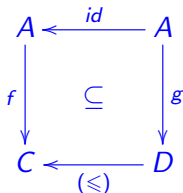
$$f \ S^R \ g \Leftrightarrow f \cdot R \subseteq S \cdot g \qquad (10)$$

In words:

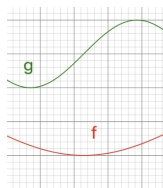
“ R -related inputs are mapped to S -related outputs”.

Squares with functions

Let $R := id$, $S := (\leq)$:



$$f \subseteq (\leq) \cdot g$$



This square captures the (\leq) -pointwise-ordering of functions:

$$f (\leq)^{id} g \Leftrightarrow \langle \forall a :: f a \leq g a \rangle$$

In words:

“The same input is mapped to (\leq) -related outputs”.

“Higher-order” squares

Because of their role in *free theorems*, these squares will be referred to as **Reynolds squares**:

$$\begin{array}{ccc}
 A & \xleftarrow{R} & B \\
 f \downarrow & \subseteq & \downarrow g \\
 C & \xleftarrow{S} & D
 \end{array}
 \quad \text{that is to say,} \quad
 \frac{A \xleftarrow{R} B \quad C \xleftarrow{S} D}{C^A \xleftarrow{S^R} D^B}$$

Thus one is lead to **relational exponentials** S^R such that e.g.

$$(S^R)^\circ = (S^\circ)^{(R^\circ)} \quad (11)$$

$$id^{id} = id \quad (12)$$

etc. **NB**: We often write $S \leftarrow R$ or $R \rightarrow S$ instead of S^R when exponents get too nested.

“Higher-order” squares

Functions-only Reynolds squares:

$$f (h \rightarrow k) g \Leftrightarrow f \cdot h = k \cdot g \quad (13)$$

In case of h° instead of h ,

$$f (h^\circ \rightarrow k) g \Leftrightarrow f \cdot h^\circ \subseteq k \cdot g \quad (14)$$

we get a **higher-order function** (via shunting + equality):

$$(h^\circ \rightarrow k) g = k \cdot g \cdot h \quad (15)$$

Then:

$$(id \rightarrow k) g = k \cdot g \quad (16)$$

$$(h^\circ \rightarrow id) g = g \cdot h \quad (17)$$

cf. **covariant** and **contravariant** exponentials.

“Higher-order” squares

In fully pointfree notation, the exponentials (16,17) become

$$k^{id} = (k \cdot)$$

$$id^{(h \circ)} = (\cdot h)$$

Then, by (11):

$$id^h = (\cdot h)^\circ \tag{18}$$

and so and so forth.

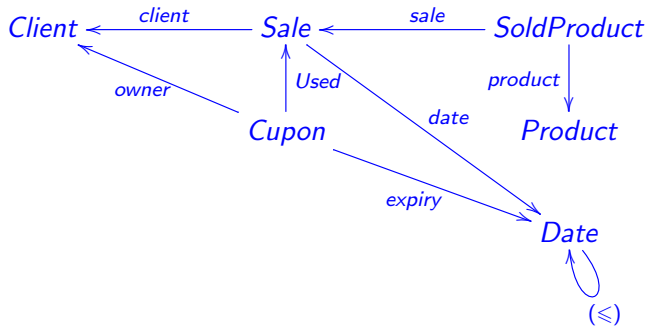


Rich construction!



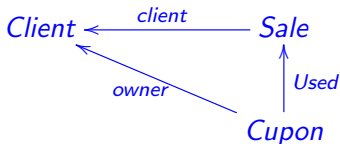
Prosaic applications

“Chase” the squares:

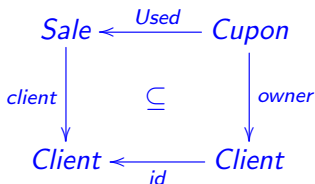


“Chase” the squares

Pick



and try orienting it:

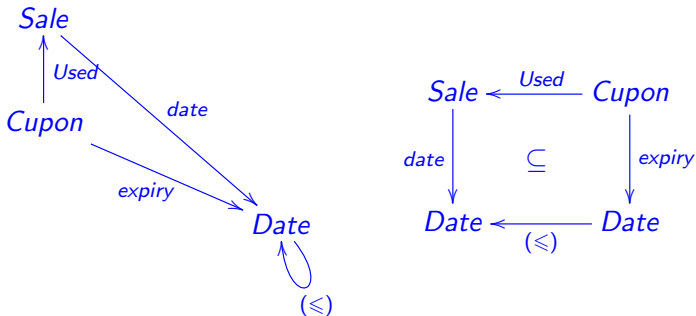


Indeed:

Coupons can only be used by clients who own them.

“Chase” the squares

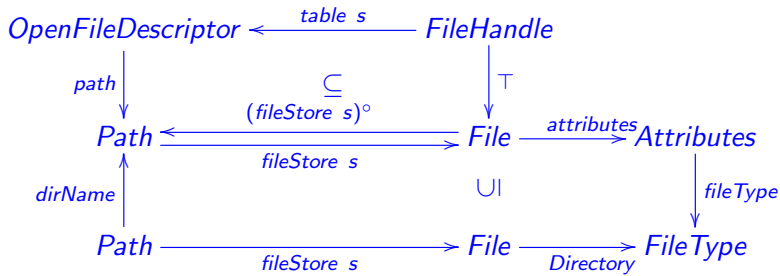
Now this one:



Coupons cannot be used beyond their expiry date.

Formal modelling

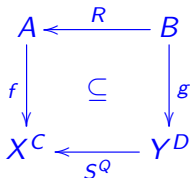
Model **invariants** are squares, cf. e.g.



(Verified File System (VFS) case study in (Oliveira and Ferreira, 2013).)

Higher-order Reynolds squares

Exponential relations S^R can involve other exponentials, for instance $(S^Q)^R$ i.e. $R \rightarrow S^Q$:



$$f (R \rightarrow S^Q) g$$

Let us unfold this, assuming all fresh variables universally quantified:

Higher-order Reynolds squares

$$f (R \rightarrow S^Q) g \tag{19}$$

$$\Leftrightarrow \quad \{ \text{Reynolds square (9)} \}$$

$$f \cdot R \subseteq S^Q \cdot g$$

$$\Leftrightarrow \quad \{ \text{shunting (5) followed by "nice rule" (8)} \}$$

$$a R b \Rightarrow (f a) S^Q (g b)$$

$$\Leftrightarrow \quad \{ \text{(9) again} \}$$

$$a R b \Rightarrow ((f a) \cdot Q \subseteq S \cdot (g b))$$

$$\Leftrightarrow \quad \{ \text{(5) followed by (8) again} \}$$

$$a R b \Rightarrow c Q d \Rightarrow (f a c) S (g b d) \tag{20}$$

Relational types

$S^R \cap id$ captures all Reynolds squares (9) in which $f = g$:

$$\begin{array}{ccc}
 A & \xleftarrow{R} & A \\
 f \downarrow & \subseteq & \downarrow f \\
 C & \xleftarrow{S} & C
 \end{array}
 \qquad
 f \cdot R \subseteq S \cdot f
 \qquad
 (21)$$

In this case we often abbreviate $f (R \rightarrow S) f$ to $f : R \rightarrow S$, meaning that f has **relational type** $R \rightarrow S$.

Note how type variables A and C in $f : A \rightarrow C$ are straightforwardly replaced by relations R and S in $f : R \rightarrow S$.



Types “are” relations (Voigtländer, 2019).

Category

Objects — binary relations R, S, \dots

Morphisms — $R \xrightarrow{f} S$ as above (21)

 This category is named Rel_2 in (Plotkin et al., 2000).

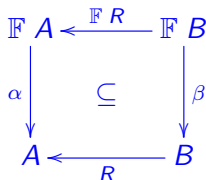
Relational type $R \rightarrow S$ corresponds to the homset $Rel_2(R, S)$.

Rel_2 is Cartesian closed, meaning that homset $R \rightarrow Q^S$ is, by uncurrying, isomorphic to $R \times S \rightarrow Q$.

NB: “Tensor” product: $(y, x) (R \times S) (b, a) \Leftrightarrow y R b \wedge x S a$.

Algebraic squares

Let $f, g := \alpha, \beta$ in a Reynolds square, where α and β are **\mathbb{F} -algebras**:



In a succinct way, the square tells that R is a **logical relation** from α to β .

Compare with:

Definition 2.2. Given a signature Σ and two models, M and N , of the language L generated by Σ , a (binary) logical relation from M to N consists of, for each type σ of L , a relation $R_\sigma \subseteq M_\sigma \times N_\sigma$ such that

- for all $f \in M_{\sigma \rightarrow \tau}$ and $g \in N_{\sigma \rightarrow \tau}$, we have $f R_{\sigma \rightarrow \tau} g$ if and only if for all $x \in M_\sigma$ and $y \in N_\sigma$, if $x R_\sigma y$ then $f(x) R_\tau g(y)$;
- for all $(x_0, x_1) \in M_{\sigma \times \tau}$ and $(y_0, y_1) \in N_{\sigma \times \tau}$, we have $(x_0, x_1) R_{\sigma \times \tau} (y_0, y_1)$ if and only if $x_0 R_\sigma y_0$ and $x_1 R_\tau y_1$;
- $* R_1 *$;
- $M(c) R_\sigma N(c)$ for every constant c in Σ of type σ .

(Plotkin et al. (2000) 'Lax Logical Relations', ICALP 2000: 85-102)

Algebraic squares

In case R is a **function** h ($R := h$),

$$\begin{array}{ccc}
 \mathbb{F} A & \xleftarrow{\mathbb{F} h} & \mathbb{F} B \\
 \alpha \downarrow & = & \downarrow \beta \\
 B & \xleftarrow{h} & A
 \end{array}$$

the square means

$$\alpha \cdot \mathbb{F} h = h \cdot \beta$$

by (7) and h is said to be a \mathbb{F} -**homomorphism**.

Coalgebraic squares

Let $f, g := \gamma, \phi$ in a Reynolds square, where γ and ϕ are \mathbb{F} -coalgebras:

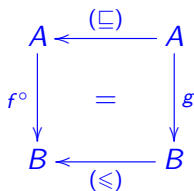
$$\begin{array}{ccc}
 A & \xleftarrow{R} & B \\
 \gamma \downarrow & \subseteq & \downarrow \phi \\
 \mathbb{F} A & \xleftarrow{\mathbb{F} R} & \mathbb{F} B
 \end{array}$$

R is said to be a **bisimulation** between the two coalgebras, meaning:

$$\langle \forall a, b : a R b : (\gamma a) (\mathbb{F} R) (\phi b) \rangle$$

Very special squares

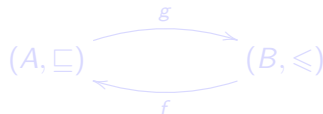
Assume two preorders (\sqsubseteq) and (\leq) in:



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

$$f \ b \sqsubseteq \ a \Leftrightarrow b \leq \ g \ a \quad (22)$$

In this very special situation,
 f and g in

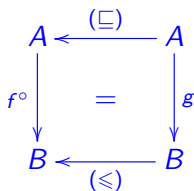


are said to be **Galois connected** (GC) and we write

$$f \vdash g \quad (23)$$

Very special squares

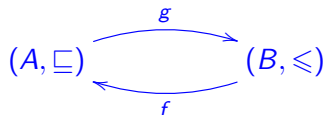
Assume two preorders (\sqsubseteq) and (\leq) in:



$$f \circ (\sqsubseteq) = (\leq) \cdot g$$

$$f \ b \sqsubseteq a \Leftrightarrow b \leq g \ a \quad (22)$$

In this very special situation,
 f and g in



are said to be **Galois connected** (GC) and we write

$$f \vdash g \quad (23)$$

Free theorem squares

Let a parametric function $f : \mathbb{F} X \rightarrow \mathbb{G} X$ be given.

Its **free theorem** states that f has **relational type**

$$f : \mathbb{F} R \rightarrow \mathbb{G} R \quad (24)$$

for any R relating its parameters, as shown in the corresponding square:

$$\begin{array}{ccc} \mathbb{F} A & \xleftarrow{\mathbb{F} R} & \mathbb{F} B \\ \downarrow f & \subseteq & \downarrow f \\ \mathbb{G} A & \xleftarrow{\mathbb{G} R} & \mathbb{G} B \end{array}$$

This extends to multi-parametric f , as shown next.

Free theorem squares

Example: Haskell constant function $\text{const} : a \rightarrow b \rightarrow a$, that is $\text{const} : A \rightarrow A^B$.

By (24), const has relational type $R \rightarrow R^S$, that is:

$$\begin{array}{ccc}
 A & \xleftarrow{R} & C \\
 \text{const} \downarrow & \subseteq & \downarrow \text{const} \\
 A^B & \xleftarrow{R^S} & C^D
 \end{array}
 \quad \text{const} \cdot R \subseteq R^S \cdot \text{const} \quad (25)$$

Pointwise equivalent, recall (19,20):

$$a R c \Rightarrow b S d \Rightarrow (\text{const } a \ b) R (\text{const } c \ d)$$

for all a, b, c, d .

Free theorem squares

Example: Haskell constant function $\text{const} : a \rightarrow b \rightarrow a$, that is $\text{const} : A \rightarrow A^B$.

By (24), const has relational type $R \rightarrow R^S$, that is:

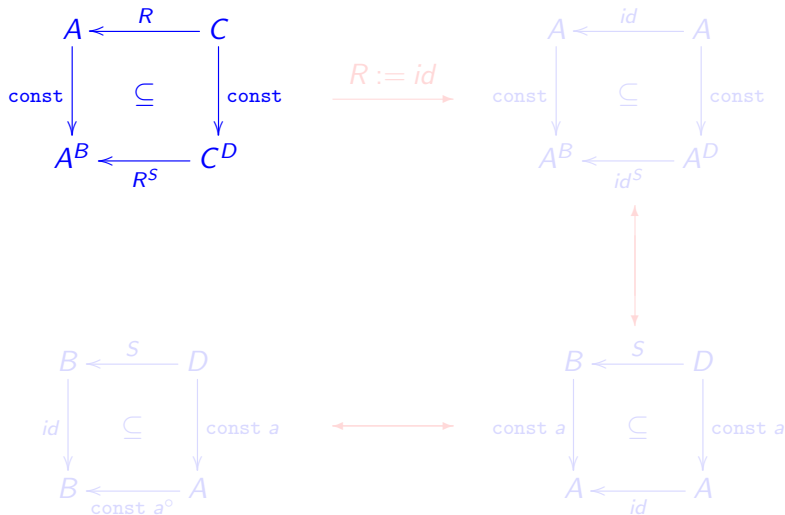
$$\begin{array}{ccc}
 A & \xleftarrow{R} & C \\
 \text{const} \downarrow & \subseteq & \downarrow \text{const} \\
 A^B & \xleftarrow{R^S} & C^D
 \end{array}
 \quad \text{const} \cdot R \subseteq R^S \cdot \text{const} \quad (25)$$

Pointwise equivalent, recall (19,20):

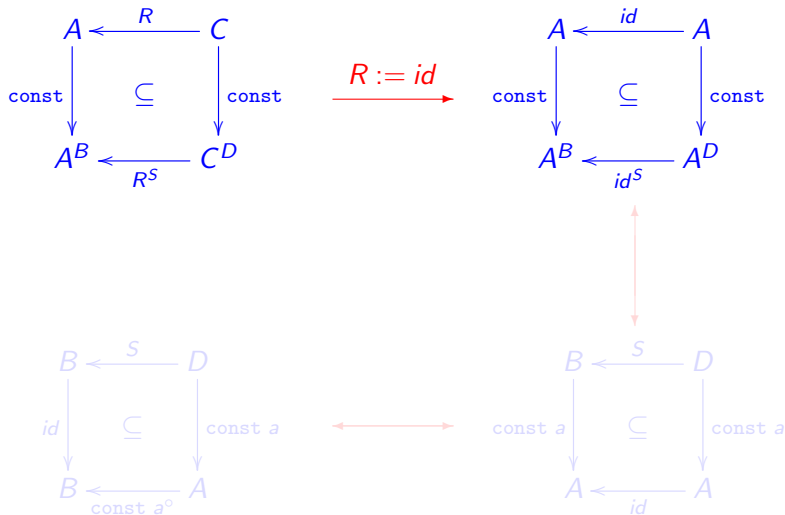
$$a R c \Rightarrow b S d \Rightarrow (\text{const } a \ b) R (\text{const } c \ d)$$

for all a, b, c, d .

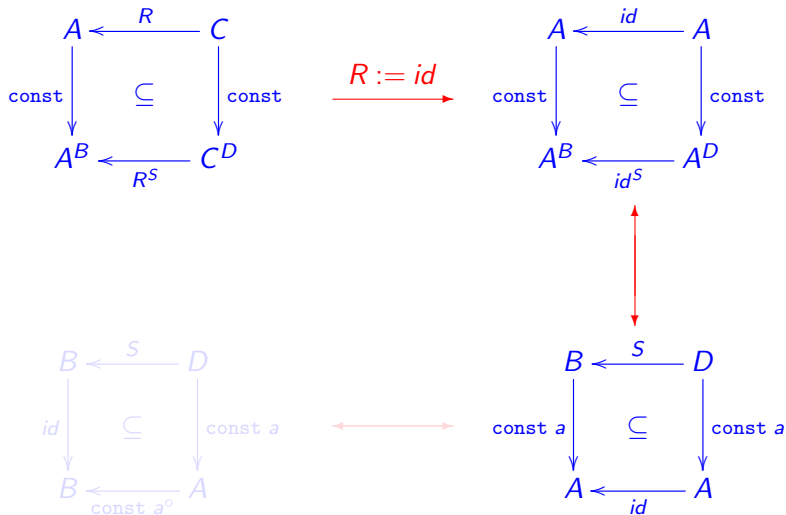
Free theorem squares



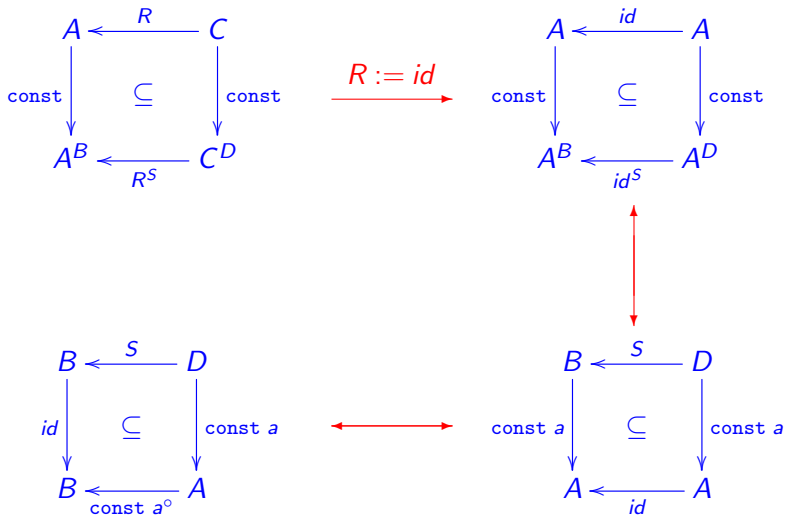
Free theorem squares



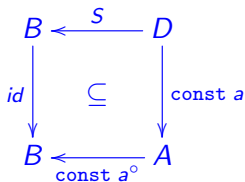
Free theorem squares



Free theorem squares



Free theorem squares



$$S \subseteq (\text{const } a)^\circ \cdot (\text{const } a)$$

So $(\text{const } a)^\circ \cdot \text{const } a$ is the largest possible S , i.e. the **top relation** \top :

$$(\text{const } a)^\circ \cdot (\text{const } a) = \top \quad (26)$$

Thus no other function can be **less injective** than $\text{const } a$.

On Injectivity

Injective functions fit in the following square:

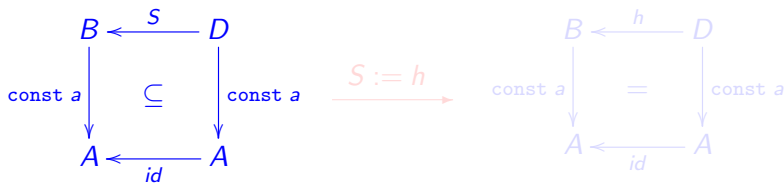
$$\begin{array}{ccc} B & \xleftarrow{f} & A \\ f^\circ \downarrow & \subseteq & \downarrow id \\ A & \xleftarrow{id} & A \end{array}$$

Path $f^\circ \cdot f$ is the **kernel** of f .

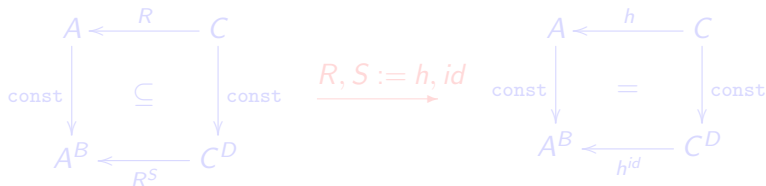
The kernel $f^\circ \cdot f$ of a function f tells how **injective** f is.

The larger the kernel the **least injective** the function is.

Free theorem squares

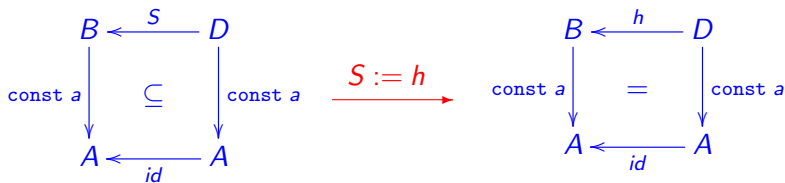


$$\text{const } a \cdot h = \text{const } a$$

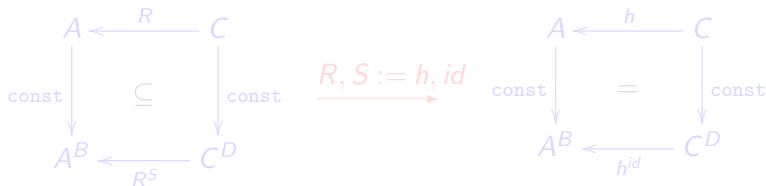


$$h \cdot (\text{const } c) = \text{const } (h \ c)$$

Free theorem squares

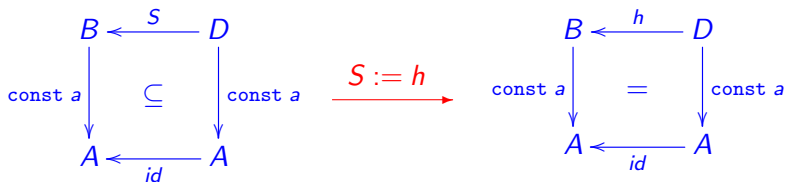


$$\text{const } a \cdot h = \text{const } a$$

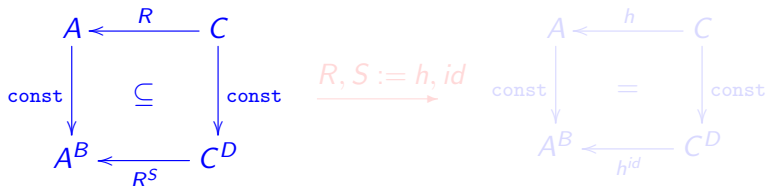


$$h \cdot (\text{const } c) = \text{const } (h c)$$

Free theorem squares

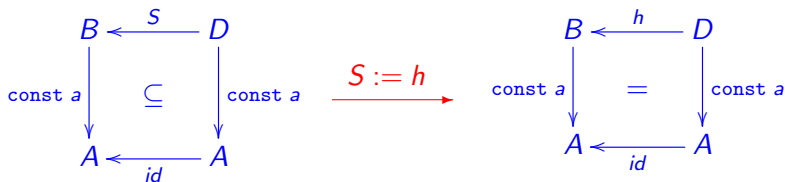


$$\text{const } a \cdot h = \text{const } a$$

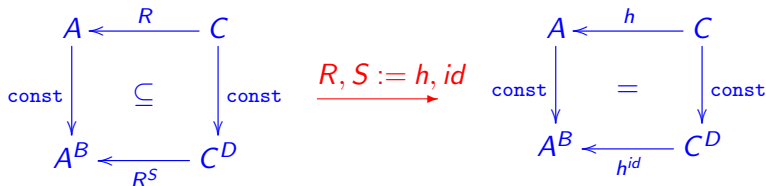


$$h \cdot (\text{const } c) = \text{const } (h \ c)$$

Free theorem squares



$$\text{const } a \cdot h = \text{const } a$$



$$h \cdot (\text{const } c) = \text{const } (h \ c)$$

Free theorem squares

Example:

$$\text{flip} :: (a \rightarrow b \rightarrow c) \rightarrow b \rightarrow a \rightarrow c \quad (27)$$

Free theorem: $\text{flip} : Q^{SR} \rightarrow Q^{RS}$, i.e.

$$g (R \rightarrow Q^S) f \Rightarrow (\text{flip } g) (S \rightarrow Q^R) (\text{flip } f)$$

that is:

$$\begin{array}{ccc}
 \begin{array}{ccc}
 A & \xleftarrow{R} & X \\
 g \downarrow & \subseteq & \downarrow f \\
 C^B & \xleftarrow{Q^S} & Z^Y
 \end{array}
 & \Rightarrow &
 \begin{array}{ccc}
 B & \xleftarrow{S} & Y \\
 \tilde{g} \downarrow & \subseteq & \downarrow \tilde{f} \\
 C^A & \xleftarrow{Q^R} & Z^X
 \end{array}
 \end{array} \quad (28)$$

Notation: \tilde{f} abbreviates $\text{flip } f$ where convenient.

Free theorem squares

For $Q := id$, $S := id$ and $R := r$ (a function):

$$\begin{aligned}
 & f (r \rightarrow id) g \Rightarrow \widetilde{f} (id \rightarrow id^r) \widetilde{g} \\
 \Leftrightarrow & \quad \{ (13) ; (16) \} \\
 & f \cdot r = g \Rightarrow \widetilde{f} \subseteq id^r \cdot \widetilde{g} \\
 \Leftrightarrow & \quad \{ id^r = (\cdot r)^\circ (18) ; \text{substitution of } g; \text{shunting (5)} \} \\
 & (\cdot r) \cdot \widetilde{f} = \widetilde{f \cdot r}
 \end{aligned}$$

This is the **fusion-law** of *flipping* — here obtained more directly than through an **adjunction** as in e.g. (Oliveira, 2020).

Free theorem squares

Since **types** are (higher-order) **squares**...

... *“how much is in a **type**”* ?

Quite a lot.

As we shall see by handling the types of the following functions:

$$\begin{aligned} \text{foldl} &:: \text{Foldable } t \Rightarrow (b \rightarrow a \rightarrow b) \rightarrow b \rightarrow t \rightarrow a \rightarrow b \\ \text{foldr} &:: \text{Foldable } t \Rightarrow (a \rightarrow b \rightarrow b) \rightarrow b \rightarrow t \rightarrow a \rightarrow b \end{aligned} \quad (29)$$

(Hackage's `Data.Foldable`)

foldl and foldr squares

Relational types (for \mathbb{T} in the **Foldable** class):

$$\mathbf{foldl} : (S \rightarrow S^R) \rightarrow (S \rightarrow S^{\mathbb{T} R}) \quad (30)$$

$$\mathbf{foldr} : (R \rightarrow S^S) \rightarrow (S \rightarrow S^{\mathbb{T} R}) \quad (31)$$

As seen above:

- Two squares in each type.
- The left one is a side-condition for the right one to hold.

foldl squares

The squares of

$$\mathbf{foldl} : (S \rightarrow S^R) \rightarrow (S \rightarrow S^{\mathbb{T} R})$$

are:

$$\begin{array}{ccc}
 \begin{array}{ccc} B & \xleftarrow{S} & Y \\ g \downarrow & \subseteq & \downarrow f \\ B^A & \xleftarrow{S^R} & Y^X \end{array} & \Rightarrow & \begin{array}{ccc} B & \xleftarrow{S} & Y \\ \mathbf{foldl} \, g \downarrow & \subseteq & \downarrow \mathbf{foldl} \, f \\ B^{\mathbb{T} A} & \xleftarrow{S^{\mathbb{T} R}} & Y^{\mathbb{T} X} \end{array}
 \end{array} \quad (32)$$

foldl squares

For $R, S := id, h$ (hence $X = A$), both $S^{\mathbb{T} R}$ and S^R reduce to $(h \cdot)$ by $\mathbb{T} id = id$ and (16).

So the squares become equalities:

$$\begin{array}{ccc}
 \begin{array}{ccc} B & \xleftarrow{h} & Y \\ g \downarrow & & \downarrow f \\ B^A & \xleftarrow{(h \cdot)} & Y^A \end{array} & \Rightarrow & \begin{array}{ccc} B & \xleftarrow{h} & Y \\ \text{foldl } g \downarrow & & \downarrow \text{foldl } f \\ B^{\mathbb{T} A} & \xleftarrow{(h \cdot)} & Y^{\mathbb{T} A} \end{array}
 \end{array}$$

Pointwise:

$$h(f y x) = g(h y) x \Rightarrow h(\text{foldl } f y xs) = \text{foldl } g(h y) xs$$

*Fusion law of **foldl** proved in (Bird and Gibbons, 2020) for finite lists.*

foldr

Repeating the above for **foldr** (31):

$$\begin{array}{ccc}
 \begin{array}{ccc}
 A & \xleftarrow{R} & X \\
 g \downarrow & \subseteq & \downarrow f \\
 B^B & \xleftarrow{S^S} & Y^Y
 \end{array}
 & \Rightarrow &
 \begin{array}{ccc}
 B & \xleftarrow{S} & Y \\
 \text{foldr } g \downarrow & \subseteq & \downarrow \text{foldr } f \\
 B^T A & \xleftarrow{S^T R} & Y^T X
 \end{array}
 \end{array}
 \quad (33)$$

Same right square as in (32), but the side-condition square is different:

$$g \cdot R \subseteq S^S \cdot f$$

foldr squares

For $R, S := id, h$ we get

$$\begin{array}{ccc}
 A & \xleftarrow{id} & A \\
 g \downarrow & & \downarrow f \\
 B^B & \xleftarrow{h^h} & Y^Y
 \end{array}
 \Rightarrow
 \begin{array}{ccc}
 B & \xleftarrow{h} & Y \\
 \text{foldr } g \downarrow & & \downarrow \text{foldr } f \\
 B^T A & \xleftarrow{(h \cdot)} & Y^T A
 \end{array}$$

where the side-condition square unfolds to:

$$\begin{aligned}
 & g (id \rightarrow h^h) f \\
 \Leftrightarrow & \quad \{ (11) \} \\
 & (g \ x) \ h^h \ (f \ x) \\
 \Leftrightarrow & \quad \{ (13) \} \\
 & (g \ x) \cdot h = h \cdot (f \ x)
 \end{aligned}$$

foldr squares

Altogether, one has

$$\begin{array}{ccc}
 \begin{array}{ccc} B & \xleftarrow{h} & Y \\ g \times \downarrow & & \downarrow f \times \\ B & \xleftarrow{h} & Y \end{array} & \Rightarrow & \begin{array}{ccc} B & \xleftarrow{h} & Y \\ \text{foldr } g \downarrow & & \downarrow \text{foldr } f \\ B^{\mathbb{T} A} & \xleftarrow{(h \cdot)} & Y^{\mathbb{T} A} \end{array}
 \end{array}$$

that is:

$$(g \times) \cdot h = h \cdot (f \times) \Rightarrow \text{foldr } g \cdot h = (h \cdot) \cdot \text{foldr } f \quad (34)$$

Going fully pointwise:

$$g \times (h y) = h (f \times y) \Rightarrow h (\text{foldr } f \ e \ xs) = \text{foldr } g \ (h e) \ xs \quad (35)$$

foldr-fusion law proved in (Bird and Gibbons, 2020) for finite lists.

Permutativity squares

Let f and g be the same function in (34), say s , and $h := s a$

$$\begin{array}{ccc}
 B & \xleftarrow{s a} & B \\
 s x \downarrow & & \downarrow s x \\
 B & \xleftarrow{s a} & B
 \end{array}
 \Rightarrow
 \begin{array}{ccc}
 B & \xleftarrow{s a} & B \\
 \text{foldr } s \downarrow & & \downarrow \text{foldr } s \\
 B^{\mathbb{T}} A & \xleftarrow{(s a \cdot)} & B^{\mathbb{T}} A
 \end{array}$$

Then (34) becomes:

$$(s x) \cdot (s a) = (s a) \cdot (s x) \Rightarrow \text{foldr } s \cdot (s a) = (s a \cdot) \cdot \text{foldr } s \quad (36)$$

Property

$$(s x) \cdot (s a) = (s a) \cdot (s x) \quad (37)$$

is called (left) **permutativity** in (Danvy, 2023).



If s is *associative* and *commutative* then it is *permutative*.

Is **foldl** equal to **foldr**?

Looking at

foldl :: *Foldable* *t* \Rightarrow (*b* \rightarrow *a* \rightarrow *b*) \rightarrow *b* \rightarrow *t* *a* \rightarrow *b*

foldr :: *Foldable* *t* \Rightarrow (*a* \rightarrow *b* \rightarrow *b*) \rightarrow *b* \rightarrow *t* *a* \rightarrow *b*

the *type-wise distance* between **foldr** and **foldl** is the flip (27) of the first parameter.

So the “best fit” one can aim at is

$$\mathbf{foldl} \ f \stackrel{?}{=} \mathbf{foldr} \ \tilde{f} \tag{38}$$

possibly valid for a (as wide as possible) class of functions *f* and instances of class *Foldable*.

But... no law relating both

Free theorems only relate pairs of folds, e.g. in (35):

$$g \times (h \ y) = h \ (f \times y) \Rightarrow h \ (\text{foldr } f \ e \ xs) = \text{foldr } g \ (h \ e) \ xs$$

Perhaps a **universal property** could be found?

For this we need to get rid of one of the **foldr**.

One way is to *assume* that, for some α and γ ,

$$\text{foldr } \alpha \ \gamma = id \tag{39}$$

holds. Then $(f, e := \alpha, \gamma)$:

$$g \times (h \ y) = h \ (\alpha \times y) \Rightarrow h \ xs = \text{foldr } g \ (h \ \gamma) \ xs$$

But... no law relating both

Free theorems only relate pairs of folds, e.g. in (35):

$$g \times (h \ y) = h \ (f \times y) \Rightarrow h \ (\text{foldr } f \ e \ xs) = \text{foldr } g \ (h \ e) \ xs$$

Perhaps a **universal property** could be found?

For this we need to get rid of one of the **foldr**.

One way is to *assume* that, for some α and γ ,

$$\text{foldr } \alpha \ \gamma = id \tag{39}$$

holds. Then $(f, e := \alpha, \gamma)$:

$$g \times (h \ y) = h \ (\alpha \times y) \Rightarrow h \ xs = \text{foldr } g \ (h \ \gamma) \ xs$$

Towards **foldr**-universal

Let us introduce $z = h \gamma$ and drop xs :

$$\begin{cases} h \gamma = z \\ h (\alpha \times y) = g \times (h y) \end{cases} \Rightarrow h = \mathbf{foldr} \ g \ z \quad (40)$$

So, $\mathbf{foldr} \ g \ z$ is the unique solution for h in the equations

$$\begin{cases} h \gamma = z \\ h (\alpha \times y) = g \times (h y) \end{cases}$$

By substituting this solution in the equations we get a definition for **foldr**:

$$\begin{cases} \mathbf{foldr} \ g \ z \ \gamma = z \\ \mathbf{foldr} \ g \ z \ (\alpha \times y) = g \times (\mathbf{foldr} \ g \ z \ y) \end{cases} \quad (41)$$

Towards **foldr**-universal

Moreover, this definition is mathematically equivalent to (just replace h by **foldr** g z and simplify):

$$h = \mathbf{foldr} \ g \ z \Rightarrow \begin{cases} h \ \gamma = z \\ h \ (\alpha \times y) = g \times (h \ y) \end{cases} \quad (42)$$

Altogether, (40) and (42) make up a universal property:

$$h = \mathbf{foldr} \ g \ z \Leftrightarrow \begin{cases} h \ \gamma = z \\ h \ (\alpha \times xs) = g \times (h \ xs) \end{cases} \quad (43)$$

(For lists, we can easily identify $\gamma = []$ and $\alpha \times xs = x : xs$.)

What about **foldl**?

Wikipedia

foldl would build. The extraneous intermediate list structure can be
 :≡ nated with the [continuation-passing style](#) technique, `foldr f z`
`xs == foldl (\k x-> k . f x) id xs z`; similarly, `foldl f`
`z xs == foldr (\x k-> k . flip f x) id xs z` (`flip` is
 only needed in languages like Haskell with its flipped order of arguments
 to the combining function of `foldl` unlike e.g., in Scheme where the
 same order of arguments is used for combining functions to both
`foldl` and `foldr`).

[https://en.wikipedia.org/wiki/Fold_\(higher-order_function\)](https://en.wikipedia.org/wiki/Fold_(higher-order_function))

Wikipedia

That is,

$$\widetilde{\text{foldl}} f = \text{foldr} (\lambda x k \rightarrow k \cdot \tilde{f} x) id \quad (44)$$

or

$$\begin{aligned} \widetilde{\text{foldl}} f &= \text{foldr} (\theta f) id \\ \text{where } (\theta f) x k &= k \cdot (\tilde{f} x) \end{aligned} \quad (45)$$

cf. the square

$$\begin{array}{ccc} B^B & \xleftarrow{\theta f} & A \\ (\cdot k) \downarrow & & \downarrow \tilde{f} \\ B^B & \xleftarrow{(k \cdot)} & B^B \end{array} \quad (46)$$

Ok?

Let us unfold (45) via universal property (43):

$$\widetilde{\text{foldl } f} = \text{foldr } (\theta f) \text{ id}$$

$$\Leftrightarrow \{ \text{universal-foldr (43)} \}$$

$$\left\{ \begin{array}{l} \widetilde{\text{foldl } f} \gamma = \text{id} \\ \widetilde{\text{foldl } f} (\alpha \times xs) z = (\theta f) \times (\widetilde{\text{foldl } f} xs) \end{array} \right.$$

$$\Leftrightarrow \{ \text{definition of } \theta \text{ (46)} \}$$

$$\left\{ \begin{array}{l} \widetilde{\text{foldl } f} \gamma = \text{id} \\ \widetilde{\text{foldl } f} (\alpha \times xs) = \widetilde{\text{foldl } f} xs (\tilde{f} \times z) \end{array} \right.$$

$$\Leftrightarrow \{ \text{go pointwise on } z \text{ and unfold the flips} \}$$

$$\left\{ \begin{array}{l} \text{foldl } f \ z \ \gamma = z \\ \text{foldl } f \ z \ (\alpha \times xs) = \text{foldl } f \ (f \ z \ x) \ xs \end{array} \right.$$

Universal-**foldl**

An advantage of defining **foldl** “as a **foldr**” (45) is that the universal property of the latter induces the universal property of the former:

$$\begin{aligned}
 k &= \mathbf{foldl} \ f \\
 \Leftrightarrow & \quad \{ \mathbf{foldl} \ f = \overbrace{\mathbf{foldr} \ (\theta \ f) \ id \ (45)} \text{ ; flipping } \} \\
 \tilde{k} &= \mathbf{foldr} \ (\theta \ f) \ id \\
 \Leftrightarrow & \quad \{ \text{universal-}\mathbf{foldr} \ (43) \text{ etc } \} \\
 & \left\{ \begin{array}{l} \tilde{k} \ \gamma = id \\ \tilde{k} \ (\alpha \times xs) = (\theta \ f) \times (\tilde{k} \ xs) \end{array} \right.
 \end{aligned}$$

Universal-foldl

$$\Leftrightarrow \quad \{ \text{introduce } z \text{ and flip} \}$$

$$\begin{cases} k \ z \ \gamma = z \\ k \ z \ (\alpha \times xs) = (\theta \ f) \times (\tilde{k} \ xs) \ z \end{cases}$$

$$\Leftrightarrow \quad \{ (\theta \ f) \times g = g \cdot (\tilde{f} \ x) \ (46) \}$$

$$\begin{cases} k \ z \ \gamma = z \\ k \ z \ (\alpha \times xs) = \tilde{k} \ xs \ (f \ z \ x) \end{cases}$$

$$\Leftrightarrow \quad \{ \text{flipping} \}$$

$$\begin{cases} k \ z \ \gamma = z \\ k \ z \ (\alpha \times xs) = k \ (f \ z \ x) \ xs \end{cases}$$

Universal-**foldl**

Thus we get the universal-property of **foldl**:

$$k = \mathbf{foldl} \ f \Leftrightarrow \begin{cases} k \ z \ \gamma = z \\ k \ z \ (\alpha \ x \ xs) = k \ (f \ z \ x) \ xs \end{cases} \quad (47)$$

Ok — now we know something else about **foldl** and **foldr**.

Let us then address the question (38) that motivated this case-study:

*Under what conditions does **foldl** $f = \mathbf{foldr} \ \tilde{f}$ hold?*

Universal-**foldl**

Thus we get the universal-property of **foldl**:

$$k = \mathbf{foldl} \ f \Leftrightarrow \begin{cases} k \ z \ \gamma = z \\ k \ z \ (\alpha \ x \ xs) = k \ (f \ z \ x) \ xs \end{cases} \quad (47)$$

Ok — now we know something else about **foldl** and **foldr**.

Let us then address the question (38) that motivated this case-study:

Under what conditions does $\mathbf{foldl} \ f = \mathbf{foldr} \ \tilde{f}$ hold?

Equating **foldl** and **foldr**

The popular assumption is that **foldl** f e and **foldr** \tilde{f} e compute the same output for f **associative** and e its **unit**, see e.g. exercise 1.10 of (Bird and Gibbons, 2020).

However, we have that, for instance (\div is **div**),

$$\mathbf{foldl} \ (\div) \ 100000 \ [99, 2, 7] = 72 = \mathbf{foldr} \ (\hat{\div}) \ 100000 \ [99, 2, 7]$$

$$\mathbf{foldl} \ (\div) \ 10000 \ [99, 2, 7] = 7 = \mathbf{foldr} \ (\hat{\div}) \ 10000 \ [99, 2, 7]$$

and yet

- neither (\div) nor $(\hat{\div})$ are associative
- the other parameter can be any number.

How do we explain this and similar examples?

Equating **foldl** and **foldr**

We can use **foldl**-universal (47) to find an answer:

$$\begin{aligned}
 & \mathbf{foldl} \ f = \mathbf{foldr} \ \tilde{f} \\
 \Leftrightarrow & \quad \{ \text{(47)} \} \\
 & \left\{ \begin{array}{l} \mathbf{foldr} \ \tilde{f} \ z \ \gamma = z \\ \mathbf{foldr} \ \tilde{f} \ z \ (\alpha \ x \ xs) = \mathbf{foldr} \ \tilde{f} \ (f \ z \ x) \ xs \end{array} \right. \\
 \Leftrightarrow & \quad \{ \text{flipping } f \ z \ x \} \\
 & \left\{ \begin{array}{l} \mathbf{foldr} \ \tilde{f} \ z \ \gamma = z \\ \mathbf{foldr} \ \tilde{f} \ z \ (\alpha \ x \ xs) = \mathbf{foldr} \ \tilde{f} \ (\tilde{f} \ x \ z) \ xs \end{array} \right.
 \end{aligned}$$

Back to the permutativity squares

Recall (36)

$$\begin{array}{ccc}
 B & \xleftarrow{s \times} & B \\
 s \ a \downarrow & & \downarrow s \ a \\
 B & \xleftarrow{s \times} & B
 \end{array}
 \Rightarrow
 \begin{array}{ccc}
 B & \xleftarrow{s \times} & B \\
 \text{foldr } s \downarrow & & \downarrow \text{foldr } s \\
 B^{\mathbb{T}} A & \xleftarrow{(s \times \cdot)} & B^{\mathbb{T}} A
 \end{array}$$

which, for $s := \tilde{f}$, becomes

$$\begin{array}{ccc}
 B & \xleftarrow{\tilde{f} \times} & B \\
 \tilde{f} \ a \downarrow & & \downarrow \tilde{f} \ a \\
 B & \xleftarrow{\tilde{f} \times} & B
 \end{array}
 \Rightarrow
 \begin{array}{ccc}
 B & \xleftarrow{\tilde{f} \times} & B \\
 \text{foldr } \tilde{f} \downarrow & & \downarrow \text{foldr } \tilde{f} \\
 B^{\mathbb{T}} A & \xleftarrow{(\tilde{f} \times \cdot)} & B^{\mathbb{T}} A
 \end{array}$$

This suits us because permuting **foldr** \tilde{f} with $\tilde{f} \times$ will be useful.
Let us see why:

Equating **foldl** and **foldr**

$$\left\{ \begin{array}{l} \mathbf{foldr} \tilde{f} z \gamma = z \\ \mathbf{foldr} \tilde{f} z (\alpha \times xs) = \mathbf{foldr} \tilde{f} (\tilde{f} \times z) xs \end{array} \right.$$

$$\Leftrightarrow \quad \{ \text{(36) assuming permutativity: } (\tilde{f} \times) \cdot (\tilde{f} a) = (\tilde{f} a) \cdot (\tilde{f} \times) \}$$

$$\left\{ \begin{array}{l} \mathbf{foldr} \tilde{f} z \gamma = z \\ \mathbf{foldr} \tilde{f} z (\alpha \times xs) = \tilde{f} \times (\mathbf{foldr} \tilde{f} z xs) \end{array} \right.$$

$$\Leftrightarrow \quad \{ \text{definition of } \mathbf{foldr} \text{ (41)} \}$$

True

Conclusion

We conclude that **foldl** $f = \text{foldr } \tilde{f}$ holds for the instances of class *Foldable* such that **foldr** $\alpha \gamma = id$ for some α and γ (39), provided that \tilde{f} is **permutative**.

Back to

$$\text{foldl } (\div) 100000 [99, 2, 7] = 72 = \text{foldr } (\widetilde{\div}) 100000 [99, 2, 7]$$

$$\text{foldl } (\div) 10000 [99, 2, 7] = 7 = \text{foldr } (\widetilde{\div}) 10000 [99, 2, 7]$$

how can we be sure $(\widetilde{\div})$ is permutative?

Equating **foldl** and **foldr**

Recall that the specification of $x \div y$ is a **Galois connection**:

$$\begin{array}{ccc}
 A & \xleftarrow{(\leq)} & A \\
 (\times y)^\circ \downarrow & = & \downarrow (\div y) \\
 B & \xleftarrow{(\leq)} & B
 \end{array}
 \qquad
 a \times y \leq x \iff a \leq x \div y \qquad (48)$$

We can use (48) and **indirect equality** over (\leq) to prove

$$(\widetilde{\div} a) \cdot (\widetilde{\div} b) = (\widetilde{\div} b) \cdot (\widetilde{\div} a)$$

that is:

$$(x \div b) \div a = (x \div a) \div b$$

Never underestimate indirect equality

$$y \leqslant (x \div b) \div a$$

$$\Leftrightarrow \quad \{ \text{Galois connection (48) twice} \}$$

$$(y \times a) \times b \leqslant x$$

$$\Leftrightarrow \quad \{ (\times) \text{ is } \mathbf{associative} \text{ and } \mathbf{commutative} \}$$

$$(y \times b) \times a \leqslant x$$

$$\Leftrightarrow \quad \{ \text{Galois connection (48) twice in the opposite direction} \}$$

$$y \leqslant (x \div a) \div b$$

$$\therefore \quad \{ \text{by indirect equality (Dijkstra, 2001)} \}$$

$$(x \div b) a = (x \div a) \div b$$

Summary

Knowing that **permutativity** is enough for foldr/foldl “equality” is not new — see e.g. (Danvy, 2023).

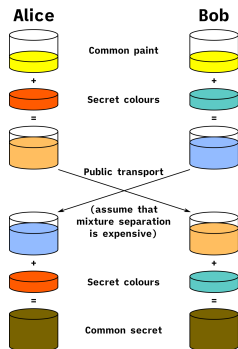
Danvy’s reasoning is, however, quite different: he **postulates** permutativity as side condition and then **proves** it in Coq by **list induction**.

Above, permutativity arose by **free-theorem** calculation.

Moreover, we’ve shown that a commutative + associative **lower adjoint** f in $f \dashv g$ ensures a permutative g , widening Olivier Danvy’s result.

Permutativity

Diffie-Hellman key exchange (Merkle, 1978)¹:



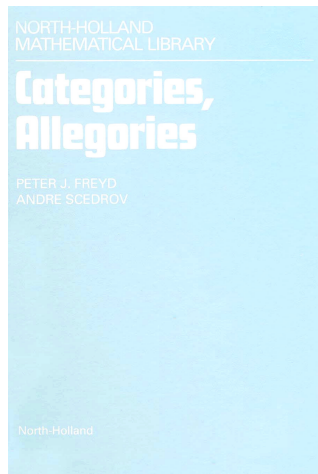
$$(+red) \cdot (+cyan) = (+cyan) \cdot (+red)$$

¹Source: Wikipedia

Summary

Following the advice of Freyd & Šcedrov:

*"This calculus [of relations] offers another, **often more amenable** framework for concepts and methods discussed in part one."*



Afterthought

A nice example of permutative operation popped up in the discussion after this talk — **insertion** on a linearly **ordered** list:

$$\text{insert} :: \text{Ord } a \Rightarrow a \rightarrow [a] \rightarrow [a]$$

Thus **insertion sort**

foldr *insert* []

computes the same as

foldl $\widetilde{\text{insert}}$ [].

(This is assumed in the example of (Gibbons, 1996).)

Annex

On relational exponentials S^R

By vertical composition (2) one immediately infers:

$$\left\{ \begin{array}{l} R' \subseteq R \\ S \subseteq S' \end{array} \right\} \Rightarrow S^R \subseteq S'^{R'}$$

We also know that $id^{id} = id$ (12).

By horizontal composition (3) we get

$$S^R \cdot S'^{R'} \subseteq (S \cdot S')^{(R \cdot R')} \quad (49)$$

However, the converse inclusion does not hold and so relational exponentiation is not in general a (bi)relator — in a sense, it can be regarded as a “lax (bi)relator”.

Backhouse and Backhouse (2004) give conditions for strengthening (49) to an equality that include the cases involving functions and converses of functions used above.

Data.Foldable

```
instance Foldable  $\mathbb{M}$  where  
  foldMap = maybe mempty  
  foldr _ z Nothing = z  
  foldr f z (Just x) = f x z  
  foldl _ z Nothing = z  
  foldl f z (Just x) = f z x
```

Let $\alpha x _ = \text{Just } x$ and $\gamma = \text{Nothing}$ and unfold **foldr** $\alpha \gamma$:

```
foldr  $\alpha$  Nothing Nothing = Nothing  
foldr  $\alpha$  Nothing (Just x) =  $\alpha x z = \text{Just } x$ 
```

So **foldr** $\alpha \gamma = \text{id}$.

References

- K. Backhouse and R.C. Backhouse. Safety of abstract interpretations for free, via logical relations and Galois connections. *SCP*, 15(1–2):153–196, 2004.
- R. Bird and J. Gibbons. *Algorithm Design with Haskell*. Cambridge University Press, 2020.
- O. Danvy. Folding left and right matters: Direct style, accumulators, and continuations. *Journal of Functional Programming*, 33:e2, 2023.
- E.W. Dijkstra. Indirect equality enriched, 2001. Technical note EWD 1315-0.
- P.J. Freyd and A. Scedrov. *Categories, Allegories*, volume 39 of *Mathematical Library*. North-Holland, 1990.
- J. Gibbons. The third homomorphism theorem. *J. Funct. Program.*, 6(4):657–665, 1996. doi: 10.1017/S0956796800001908. URL <https://doi.org/10.1017/S0956796800001908>.
- R.C. Merkle. Secure communications over insecure channels. *Commun. ACM*, 21(4):294–299, 1978.

- J.N. Oliveira. A note on the under-appreciated for-loop. Technical Report TR-HASLab:01:2020 (PDF), HASLab/U.Minho and INESC TEC, 2020.
- J.N. Oliveira and M.A. Ferreira. Alloy meets the algebra of programming: A case study. *IEEE Trans. Soft. Eng.*, 39(3): 305–326, 2013. .
- G. Plotkin, J. Power, D. Sannella, and R. Tennent. Lax logical relations. In Ugo Montanari, José D. P. Rolim, and Emo Welzl, editors, *Automata, Languages and Programming*, pages 85–102, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.
- J. Voigtländer. Free theorems simply, via dinaturality, 2019. arXiv cs.PL 1908.07776.