"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
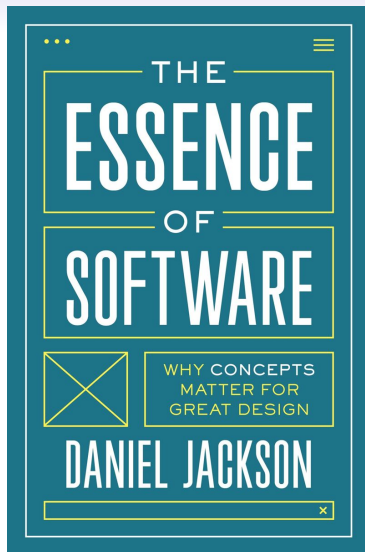

University of Minho & INESC TEC

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**.
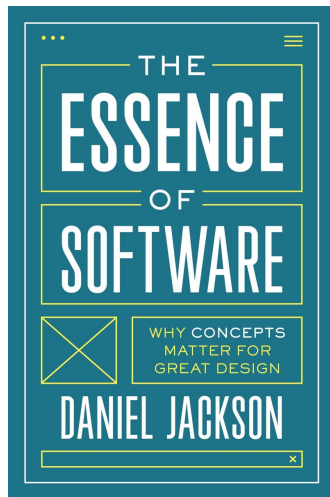
The Essence of Software

"(...) 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."

NORTH-HOLLAND
MATHEMATICAL LIBRARY

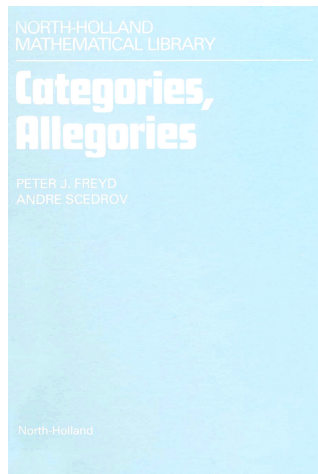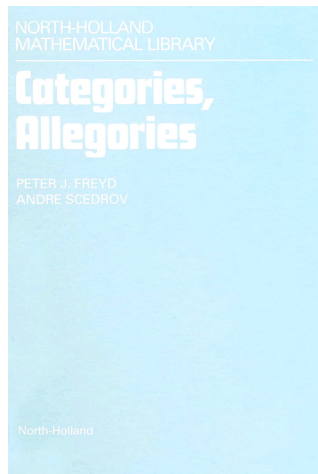**Categories, Allegories**

PETER J. FREYD
ANDRE SCEDROV

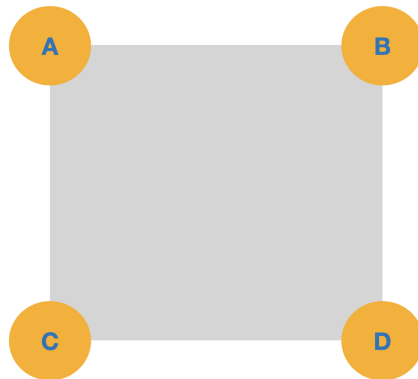North-Holland

Freyd & Ščedrov, 1990

"(...) A special feature of our approach is a general **calculus of relations** presented in part two.
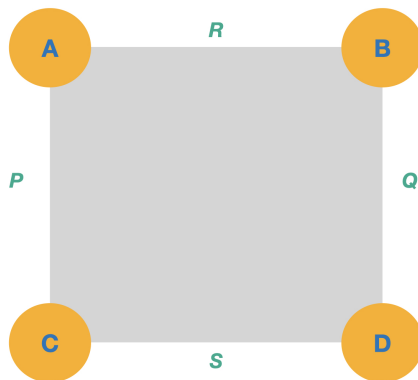
This calculus offers another, **often more amenable** framework for concepts and methods discussed in part one."

NORTH-HOLLAND
MATHEMATICAL LIBRARY

**Categories,
Allegories**

PETER J. FREYD
ANDRE SCEDROV

North-Holland
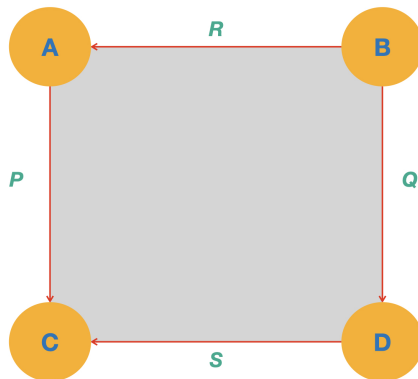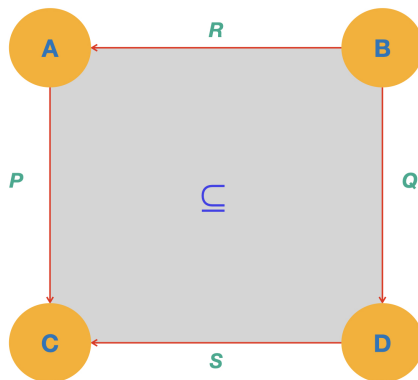
Squares
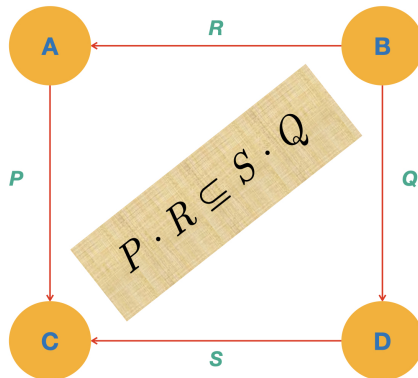
"Magic" square

"Magic" square

Four binary relations:

$$A \xleftarrow{\;R\;} B$$
$$P \downarrow \quad \subseteq \quad \downarrow Q \qquad\qquad P \cdot R \subseteq S \cdot Q \qquad\qquad (1)$$
$$C \xleftarrow{\;S\;} D$$
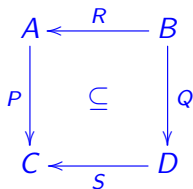
Terminology:

R — *pre-producer*

P — *pre-consumer*

Q — *post-producer*

S — *post-consumer*

"Magic" square

Pointfree:

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

Pointwise:

"Magic" square

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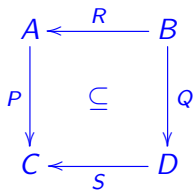
$$P \cdot R \subseteq S \cdot Q$$

Pointwise:

"Magic" square

Perhaps not what you were expecting...

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

$$A \xleftarrow{\ R\ } B$$

$$P \downarrow \quad \subseteq \quad \downarrow Q$$

$$C \xleftarrow{\ S\ } D$$
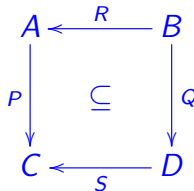
Vertical composition

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

Horizontal composition

$$A \xleftarrow{R} C \xleftarrow{R'} C' \qquad A \xleftarrow{R \cdot R'} C'$$

(3)

Identity
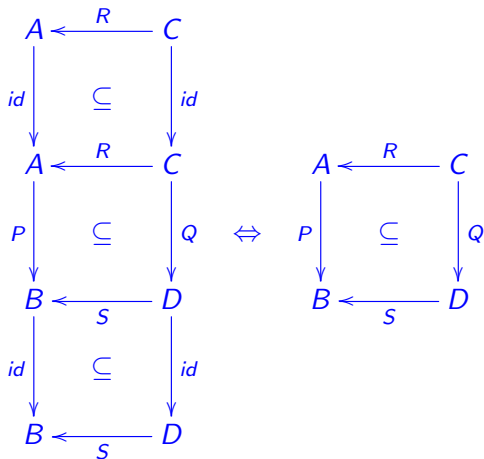
(Similarly for horizontal.)

Converse

$$
\begin{array}{ccc}
A & \xleftarrow{\ R\ } & C \\
{\scriptstyle P}\big\downarrow & \subseteq & \big\downarrow{\scriptstyle Q} \\
B & \xleftarrow{\ S\ } & D
\end{array}
\qquad \Leftrightarrow \qquad
\begin{array}{ccc}
A & \xleftarrow{\ P^{\circ}\ } & B \\
{\scriptstyle R^{\circ}}\big\downarrow & \subseteq & \big\downarrow{\scriptstyle S^{\circ}} \\
C & \xleftarrow{\ Q^{\circ}\ } & D
\end{array}
$$
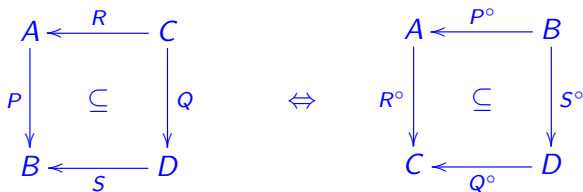
The converse of a square is its "*passive voice*" ☺

Square (direct) sums

Functorial squares

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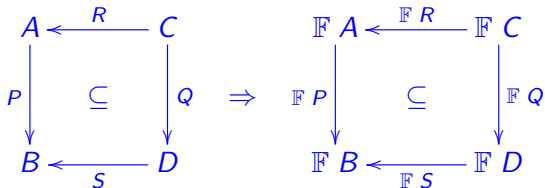
$\mathbb{F}$ should be monotonic and preserve converses — a **relator** (Freyd and Scedrov, 1990).

Squares with functions

Some relations f fit into the following squares:

$$
\begin{array}{ccccc}
A & \xleftarrow{\;id\;} & A & \xleftarrow{\;f^{\circ}\;} & B \\[2pt]
{\scriptstyle id}\downarrow & \subseteq & {\scriptstyle f}\downarrow & \subseteq & \downarrow{\scriptstyle id} \\[2pt]
A & \xleftarrow{\;f^{\circ}\;} & B & \xleftarrow{\;id\;} & B
\end{array}
\qquad (4)
$$

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

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

$\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ f is **univocal**.

Such relations f are called **functions**.

Squares with functions

Some relations f fit into the following squares:

$$
\begin{array}{ccccc}
A & \xleftarrow{\ id\ } & A & \xleftarrow{\ f^\circ\ } & B \\
{\scriptstyle id}\downarrow & \subseteq & {\scriptstyle f}\downarrow & \subseteq & \downarrow{\scriptstyle id} \\
A & \xleftarrow{\ f^\circ\ } & B & \xleftarrow{\ id\ } & B
\end{array}
\qquad (4)
$$

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

$$
\begin{array}{ccccc}
A & \xleftarrow{\;id\;} & A & \xleftarrow{\;f^{\circ}\;} & B \\[2pt]
\scriptstyle id \downarrow & \subseteq & \scriptstyle f \downarrow & \subseteq & \scriptstyle id \downarrow \\[2pt]
A & \xleftarrow{\;f^{\circ}\;} & B & \xleftarrow{\;id\;} & B
\end{array}
\tag{4}
$$

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:

$$
\begin{array}{ccccc}
A & \xleftarrow{\;id\;} & A & \xleftarrow{\;f^\circ\;} & B \\
{\scriptstyle id}\downarrow & \subseteq & \downarrow{\scriptstyle f} & \subseteq & \downarrow{\scriptstyle id} \\
A & \xleftarrow{\;f^\circ\;} & B & \xleftarrow{\;id\;} & B
\end{array}
\qquad (4)
$$

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:

$$A \xleftarrow{R} C \qquad A \xleftarrow{R} C$$

$$f \downarrow \quad \subseteq \quad \downarrow Q \qquad \Leftrightarrow \qquad id \downarrow \quad \subseteq \quad \downarrow Q$$

$$B \xleftarrow{id} B \qquad A \xleftarrow{f^\circ} B$$

This is the **shunting** rule

$$f \cdot R \subseteq Q \quad \Leftrightarrow \quad R \subseteq f^\circ \cdot Q \tag{5}$$

which, by taking converses, becomes:

$$R \cdot f^\circ \subseteq Q \quad \Leftrightarrow \quad R \subseteq Q \cdot f \tag{6}$$

"Nice" rules about functions

(Functional) **equality**:

$$f \subseteq g \;\Leftrightarrow\; f = g \;\Leftrightarrow\; g \subseteq f \tag{7}$$

Existential quantifiers go away:

$$b \, (f^{\circ} \cdot R \cdot g) \, a \;\Leftrightarrow\; (f \, b) \, R \, (g \, a) \tag{8}$$

☞ Very useful in practice ☺

Squares with functions

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

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

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

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

In words:

"R-related inputs are mapped to S-related outputs".

Squares with functions

Let $R := id$, $S := (\leqslant)$:
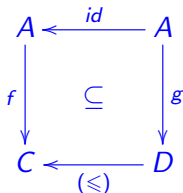
$$f \subseteq (\leqslant) \cdot g$$
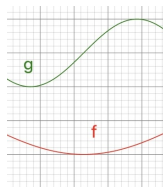
This square captures the $(\leqslant)$-pointwise-ordering of functions:

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

In words:

"The same input is mapped to $(\leqslant)$-related outputs".

"Higher-order" squares

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

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

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

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

$$id^{id} = id \tag{12}$$

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:

$$f \ (h \to k) \ g \quad \Leftrightarrow \quad f \cdot h = k \cdot g \tag{13}$$

In case of h° instead of h,

$$f \ (h^\circ \to k) \ g \Leftrightarrow f \cdot h^\circ \subseteq k \cdot g \tag{14}$$

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

$$(h^\circ \to k) \ g = k \cdot g \cdot h \tag{15}$$

Then:

$$(id \to k) \ g \ = \ k \cdot g \tag{16}$$
$$(h^\circ \to id) \ g \ = \ g \cdot h \tag{17}$$

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):

$$id^h = (\cdot h)^\circ \qquad (18)$$

and so and so forth.

☞　　Rich construction!　　🙂

Prosaic applications

"Chase" the squares:
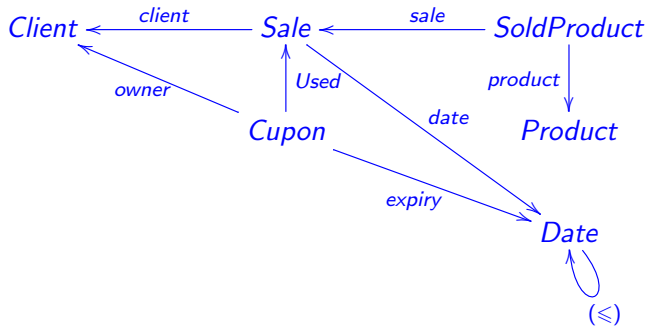
"Chase" the squares

Pick
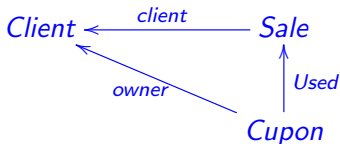
and try orienting it:
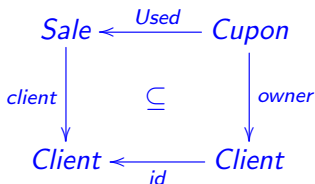
Indeed:

Coupons can only be used by clients who own them.

"Chase" the squares

Now this one:
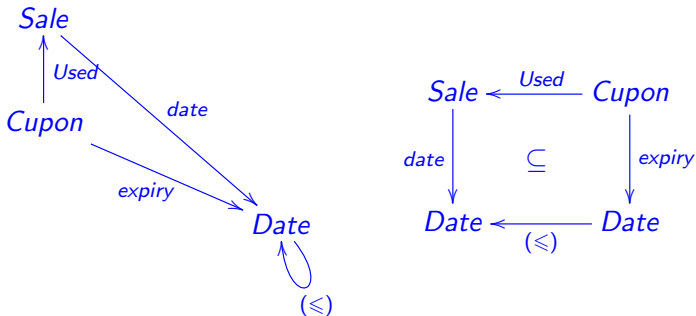
Coupons cannot be used beyond their expiry date.

Formal modelling

Model **invariants** are squares, cf. e.g.
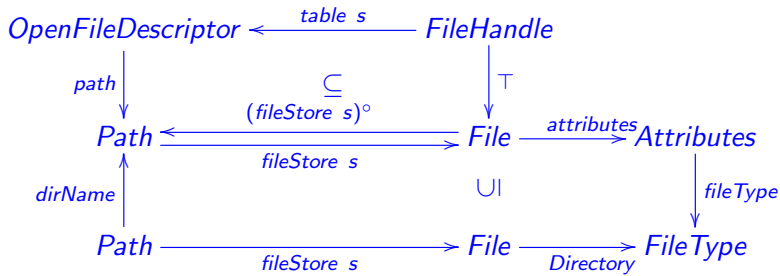
(Verified File System (VFS) case study in (Oliveira and Ferreira, 2013).)

Higher-order Reynolds squares

Exponential relations S^R can involve other exponentials, for instance $(S^Q)^R$ i.e. $R \to S^Q$:
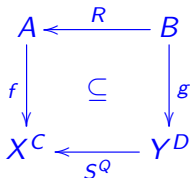
$$
\begin{array}{ccc}
A & \xleftarrow{\quad R \quad} & B \\
{\scriptstyle f} \downarrow & \subseteq & \downarrow {\scriptstyle g} \\
X^C & \xleftarrow{\quad S^Q \quad} & Y^D
\end{array}
\qquad\qquad f \,(R \to S^Q)\, g
$$

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

Higher-order Reynolds squares

$$f \ (R \to S^Q) \ g \qquad\qquad (19)$$

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

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

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

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

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

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

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

$$a \ R \ b \Rightarrow c \ Q \ d \Rightarrow (f \ a \ c) \ S \ (g \ b \ d) \qquad\qquad (20)$$

Relational types

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

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

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

> *Note how type variables A and C in $f : A \to C$ are straightforwardly replaced by relations R and S in $f : R \to S$.*
>
>   ☞   **Types** *"are"* **relations** *(Voigtländer, 2019).*

Category

Objects — binary relations R, S, ...

Morphisms — $R \xrightarrow{f} S$ as above (21)
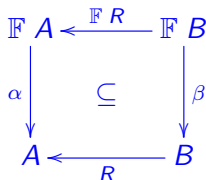
☞ This category is named Rel_2 in (Plotkin et al., 2000).

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

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

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**:

$$
\begin{array}{ccc}
\mathbb{F}\,A & \xleftarrow{\;\mathbb{F}\,R\;} & \mathbb{F}\,B \\
{\scriptstyle\alpha}\downarrow & \subseteq & \downarrow{\scriptstyle\beta} \\
A & \xleftarrow{\quad R \quad} & B
\end{array}
$$

In a succint 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 $f \in M_{\sigma \to \tau}$ and $g \in N_{\sigma \to \tau}$, we have $f\,R_{\sigma \to \tau}\,g$ 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
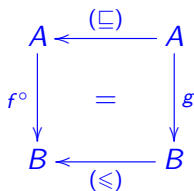
Assume two preorders $(\sqsubseteq)$ and $(\leqslant)$ in:

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

$$f\ b \sqsubseteq a \Leftrightarrow b \leqslant g\ a \qquad (22)$$

In this very special situation, f and g in

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

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

Very special squares
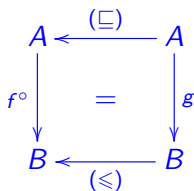
Assume two preorders $(\sqsubseteq)$ and $(\leqslant)$ in:

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

$$f\ b \sqsubseteq a \Leftrightarrow b \leqslant g\ a \quad (22)$$
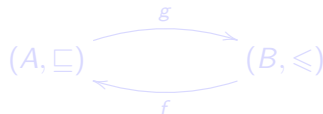
In this very special situation, f and g in
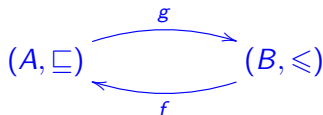
are said to be **Galois connected** (GC) and we write

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

Free theorem squares

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

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

$$f : \mathbb{F} R \to \mathbb{G} R \tag{24}$$

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

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

Free theorem squares

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

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

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

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 $\mathtt{const} : a \to b \to a$, that is $\mathtt{const} : A \to A^B$.

By (24), $\mathtt{const}$ has relational type $R \to R^S$, that is:

$$
\begin{array}{ccc}
A & \xleftarrow{\quad R \quad} & C \\[2pt]
{\scriptstyle\mathtt{const}}\Big\downarrow & \subseteq & \Big\downarrow{\scriptstyle\mathtt{const}} \\[2pt]
A^B & \xleftarrow{\quad R^S \quad} & C^D
\end{array}
\qquad
\mathtt{const} \cdot R \;\subseteq\; R^S \cdot \mathtt{const}
\qquad (25)
$$

Pointwise equivalent, recall (19,20):

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

for all a, b, c, d.

Free theorem squares

Free theorem squares
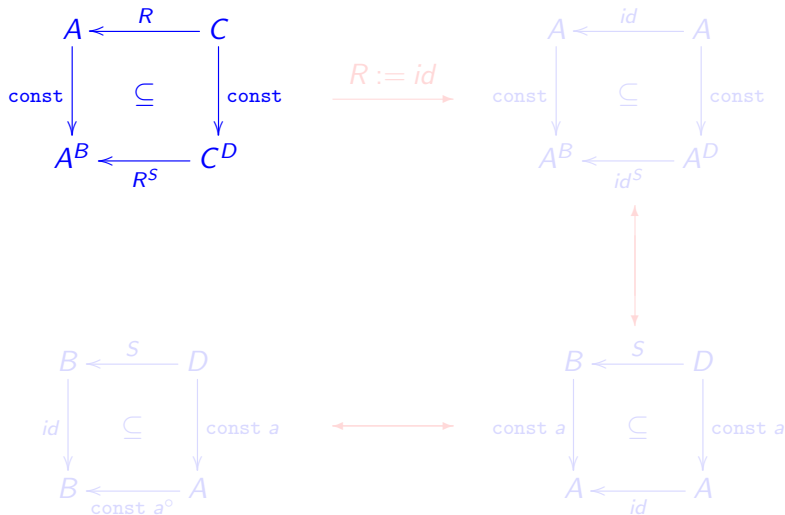
Free theorem squares
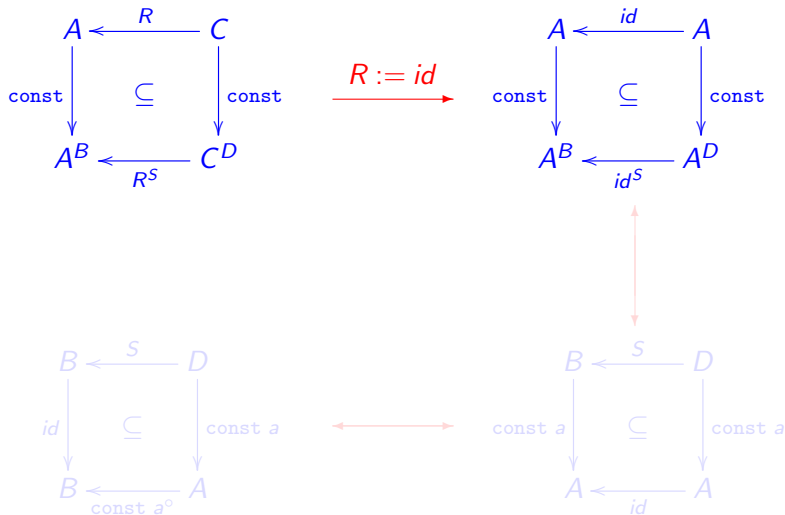
Free theorem squares
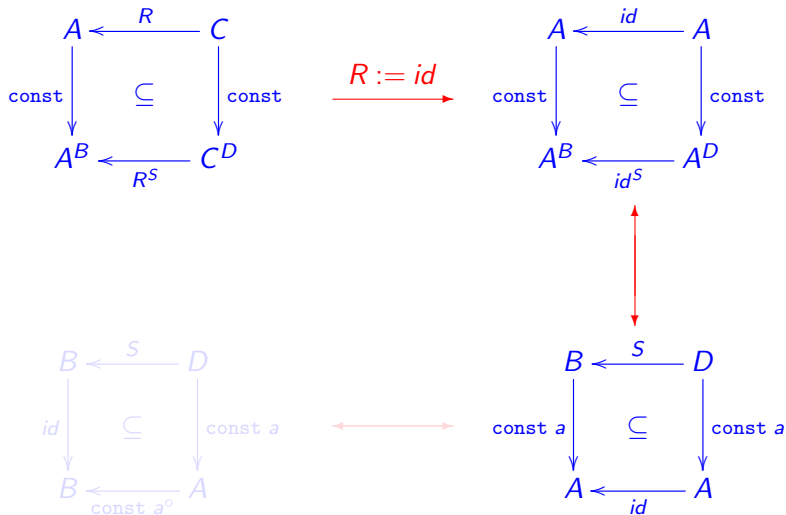
Free theorem squares
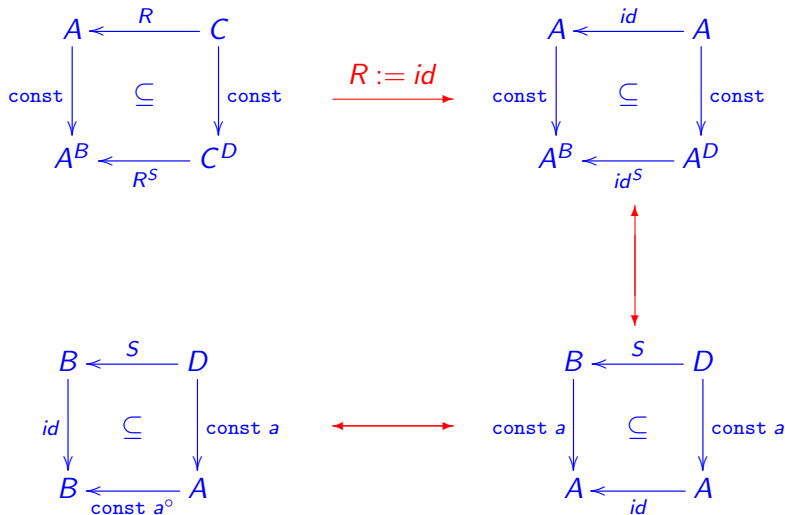
$$B \xleftarrow{\quad S \quad} D \qquad\qquad S \subseteq (\mathrm{const}\ a)^{\circ} \cdot (\mathrm{const}\ a)$$
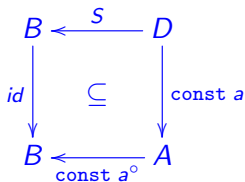
$$id \downarrow \qquad \subseteq \qquad \downarrow \mathrm{const}\ a$$

$$B \xleftarrow[\mathrm{const}\ a^{\circ}]{} A$$

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

$$(\mathrm{const}\ a)^{\circ} \cdot (\mathrm{const}\ a) = \top \qquad\qquad (26)$$

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

On Injectivity

Injective functions fit in the following square:

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

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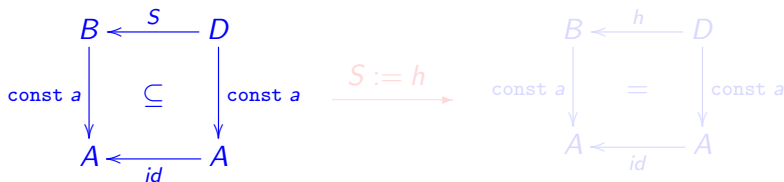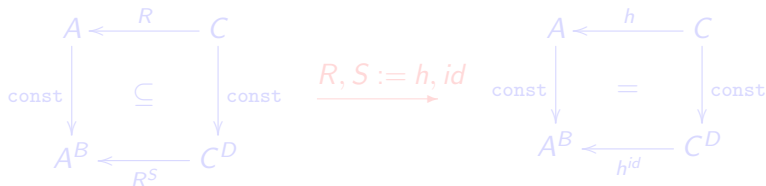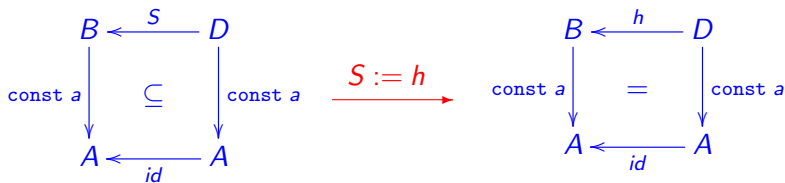
$$\mathrm{const}\ a \cdot h = \mathrm{const}\ a$$

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

Free theorem squares
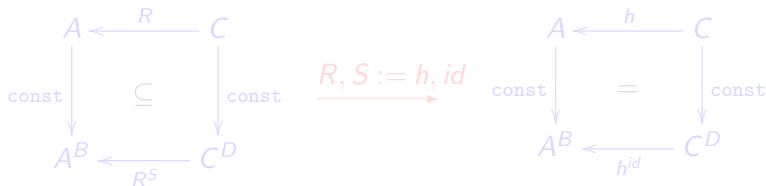
$$\text{const } a \cdot h = \text{const } a$$

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

Free theorem squares
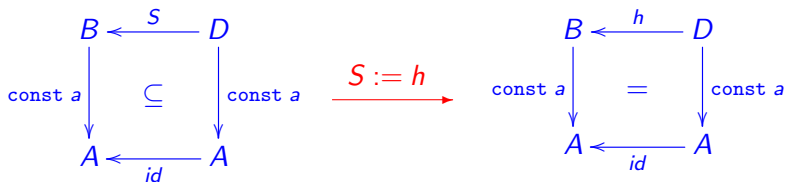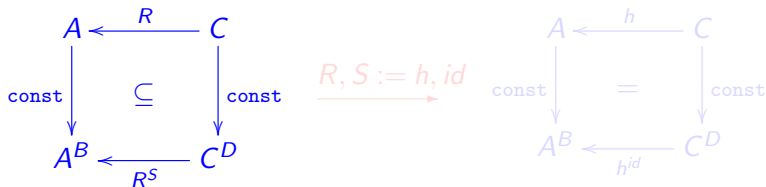
$$\text{const } a \cdot h = \text{const } a$$

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

Free theorem squares
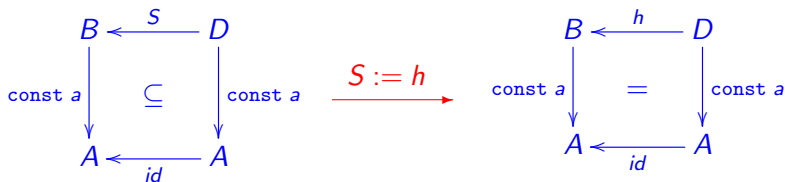
$$\text{const } a \cdot h = \text{const } a$$
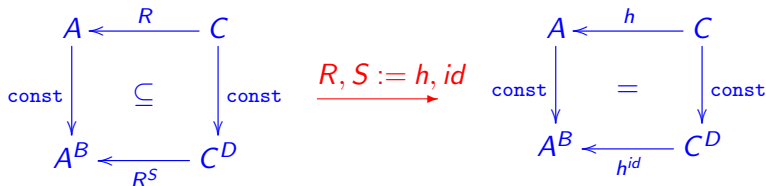
$$h \cdot (\text{const } c) = \text{const } (h\, c)$$

Free theorem squares

Example:

$$\texttt{flip} :: (a \rightarrow b \rightarrow c) \rightarrow b \rightarrow a \rightarrow c \tag{27}$$

Free theorem: $\texttt{flip} : Q^{S^R} \rightarrow Q^{R^S}$, i.e.

$$g \left(R \rightarrow Q^S \right) f \;\; \Rightarrow \;\; (\texttt{flip}\, g) \left(S \rightarrow Q^R \right) (\texttt{flip}\, f)$$

that is:

$$
\begin{array}{ccccc}
A \xleftarrow{\;R\;} X & & & B \xleftarrow{\;S\;} Y & \\
{\scriptstyle g}\downarrow \quad \subseteq \quad \downarrow {\scriptstyle f} & \Rightarrow & {\scriptstyle \widetilde{g}}\downarrow \quad \subseteq \quad \downarrow {\scriptstyle \widetilde{f}} & \\
C^B \xleftarrow{\;Q^S\;} Z^Y & & & C^A \xleftarrow{\;Q^R\;} Z^X &
\end{array}
\tag{28}
$$

Notation: $\widetilde{f}$ abbreviates $\texttt{flip}\, f$ where convenient.

Free theorem squares

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

$$f\ (r \rightarrow id)\ g \Rightarrow \widetilde{\widehat{f}}\ (id \rightarrow id^r)\ \widetilde{g}$$

$$\Leftrightarrow \qquad \{\ \ (13)\ ;\ (16)\ \ \}$$

$$f \cdot r = g\ \Rightarrow\ \widetilde{\widehat{f}}\ \subseteq\ id^r \cdot \widetilde{g}$$

$$\Leftrightarrow \qquad \{\ \ id^r = (\cdot r)^\circ\ (18)\ ;\ \text{substitution of } g;\ \text{shunting } (5)\ \ \}$$

$$(\cdot r) \cdot \widetilde{\widehat{f}} = \widetilde{f \cdot r}$$

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:

$$\textbf{foldl} :: Foldable\ t \Rightarrow (b \to a \to b) \to b \to t\ a \to b$$
$$\textbf{foldr} :: Foldable\ t \Rightarrow (a \to b \to b) \to b \to t\ a \to b$$

(29)

(Hackage's Data.Foldable)

foldl and **foldr** squares

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

$$\textbf{foldl} : (S \to S^R) \to (S \to S^{\mathbb{T}\,R}) \tag{30}$$

$$\textbf{foldr} : (R \to S^S) \to (S \to S^{\mathbb{T}\,R}) \tag{31}$$

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

$$\textbf{foldl} : (S \to S^R) \to (S \to S^{\mathbb{T}\,R})$$

are:

$$(32)$$

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:

$$
\begin{array}{ccc}
B \xleftarrow{\;h\;} Y & & B \xleftarrow{\;h\;} Y \\
{\scriptstyle g}\downarrow \quad \downarrow{\scriptstyle f} & \Rightarrow \quad {\color{red}\textbf{foldl}}\;{\scriptstyle g}\downarrow \quad \downarrow{\color{red}\textbf{foldl}}\;{\scriptstyle f} \\
B^A \xleftarrow[(h\cdot)]{} Y^A & & B^{\mathbb{T} A} \xleftarrow[(h\cdot)]{} Y^{\mathbb{T} A}
\end{array}
$$

Pointwise:

$$h \, (f \, y \, x) = g \, (h \, y) \, x \;\; \Rightarrow \;\; h \, (\textbf{foldl} \, f \, y \, xs) = \textbf{foldl} \, g \, (h \, y) \, xs$$

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

---

foldr

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

$$
\begin{array}{ccc}
A \xleftarrow{R} X & & B \xleftarrow{S} Y \\
g \downarrow \quad \subseteq \quad \downarrow f & \Rightarrow \quad \textbf{foldr } g \downarrow \quad \subseteq \quad \downarrow \textbf{foldr } f & \\
B^B \xleftarrow{S^S} Y^Y & & B^{\mathbb{T}\, A} \xleftarrow{S^{\mathbb{T}\, R}} Y^{\mathbb{T}\, X}
\end{array}
\tag{33}
$$

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

where the side-condition square unfolds to:

$$g \ (id \rightarrow h^h) \ f$$

$\Leftrightarrow \qquad \{ \ (11) \ \}$

$$(g \ x) \ h^h \ (f \ x)$$

$\Leftrightarrow \qquad \{ \ (13) \ \}$

$$(g \ x) \cdot h = h \cdot (f \ x)$$

foldr squares

Altogether, one has

that is:

$$(g\ x) \cdot h = h \cdot (f\ x) \quad \Rightarrow \quad \textbf{foldr}\ g \cdot h = (h \cdot) \cdot \textbf{foldr}\ f \qquad (34)$$

Going fully pointwise:

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

---

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 $h := s\,a$

$$
\begin{array}{ccc}
B & \xleftarrow{\ s\,a\ } & B \\
{\scriptstyle s\,x}\downarrow & & \downarrow{\scriptstyle s\,x} \\
B & \xleftarrow[s\,a]{} & B
\end{array}
\quad\Rightarrow\quad
\begin{array}{ccc}
B & \xleftarrow{\ s\,a\ } & B \\
{\scriptstyle \textbf{foldr } s}\downarrow & & \downarrow{\scriptstyle \textbf{foldr } s} \\
B^{\mathbb{T}\,A} & \xleftarrow[(s\,a\cdot)]{} & B^{\mathbb{T}\,A}
\end{array}
$$

Then (34) becomes:

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

Property

$$(s\,x)\cdot(s\,a) = (s\,a)\cdot(s\,x) \qquad\qquad (37)$$

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

$$\textbf{foldl}\ f \quad \overset{?}{=} \quad \textbf{foldr}\ \widetilde{f} \tag{38}$$

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):

$$g\ x\ (h\ y) = h\ (f\ x\ y) \quad \Rightarrow \quad h\ (\textbf{foldr}\ f\ e\ xs) = \textbf{foldr}\ g\ (h\ e)\ xs$$

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,

$$\textbf{foldr}\ \alpha\ \gamma = id \qquad\qquad\qquad\qquad\qquad (39)$$

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

$$g\ x\ (h\ y) = h\ (\alpha\ x\ y) \quad \Rightarrow \quad h\ xs = \textbf{foldr}\ g\ (h\ \gamma)\ xs$$

But... no law relating both

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

$$g \; x \; (h \; y) = h \; (f \; x \; y) \;\; \Rightarrow \;\; h \; (\textbf{foldr} \; f \; e \; xs) = \textbf{foldr} \; g \; (h \; e) \; xs$$

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,

$$\textbf{foldr} \; \alpha \; \gamma = id \tag{39}$$

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

$$g \; x \; (h \; y) = h \; (\alpha \; x \; y) \;\; \Rightarrow \;\; h \; xs = \textbf{foldr} \; g \; (h \; \gamma) \; xs$$

Towards **foldr**-universal

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

$$\left\{ \begin{array}{l} h\,\gamma = z \\ h\,(\alpha\,x\,y) = g\,x\,(h\,y) \end{array} \right. \quad \Rightarrow \quad h = \textbf{foldr}\;g\;z \qquad (40)$$

So, **foldr** $g\;z$ is the unique solution for h in the equations

$$\left\{ \begin{array}{l} h\,\gamma = z \\ h\,(\alpha\,x\,y) = g\,x\,(h\,y) \end{array} \right.$$

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

$$\left\{ \begin{array}{l} \textbf{foldr}\;g\;z\;\gamma = z \\ \textbf{foldr}\;g\;z\;(\alpha\,x\,y) = g\,x\,(\textbf{foldr}\;g\;z\;y) \end{array} \right. \qquad (41)$$

Towards **foldr**-universal

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

$$h = \textbf{foldr } g \; z \quad \Rightarrow \quad \left\{ \begin{array}{l} h \, \gamma = z \\ h \, (\alpha \; x \; y) = g \; x \; (h \; y) \end{array} \right. \tag{42}$$

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

$$h = \textbf{foldr } g \; z \quad \Leftrightarrow \quad \left\{ \begin{array}{l} h \, \gamma = z \\ h \, (\alpha \; x \; xs) = g \; x \; (h \; xs) \end{array} \right. \tag{43}$$

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

What about **foldl**?

Wikipedia

... would build. The extraneous intermediate list structure can be eliminated 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)

Wikipedia

That is,

$$\widetilde{\textbf{foldl } f} = \textbf{foldr } (\lambda x\ k \rightarrow k \cdot \hat{\tilde{f}}\ x)\ id \qquad (44)$$

or

$$\widetilde{\textbf{foldl } f} = \textbf{foldr } (\theta\ f)\ id \qquad (45)$$
$$\textbf{where } (\theta\ f)\ x\ k = k \cdot (\hat{\tilde{f}}\ x)$$

cf. the square

$$
\begin{array}{ccc}
B^B & \xleftarrow{\ \theta\ f\ } & A \\
{\scriptstyle (\cdot k)}\Big\downarrow & & \Big\downarrow{\scriptstyle \hat{\tilde{f}}} \\
B^B & \xleftarrow[\ (k\cdot)\ ]{} & B^B
\end{array}
\qquad (46)
$$

Ok?

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

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

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

$$\begin{cases} \widetilde{\mathbf{foldl}\ f}\ \gamma = id \\ \widetilde{\mathbf{foldl}\ f}\ (\alpha\ x\ xs)\ z = (\theta\ f)\ x\ (\widetilde{\mathbf{foldl}\ f}\ xs) \end{cases}$$

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

$$\begin{cases} \widetilde{\mathbf{foldl}\ f}\ \gamma = id \\ \widetilde{\mathbf{foldl}\ f}\ (\alpha\ x\ xs) = \widetilde{\mathbf{foldl}\ f}\ xs\ (\overset{\sim}{\hat{f}}\ x\ z) \end{cases}$$

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

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

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:

$$k = \textbf{foldl } f$$

$$\Leftrightarrow \qquad \{ \ \textbf{foldl } f = \overbrace{\textbf{foldr } (\theta \ f) \ id} \ (45) \ ; \ \text{flipping} \ \}$$

$$\tilde{k} = \textbf{foldr } (\theta \ f) \ id$$

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

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

Universal-**foldl**

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

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

$\Leftrightarrow \qquad \{ \;\; (\theta \; f) \; x \; g = g \cdot (\tilde{f} \; x) \; (46) \;\; \}$

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

$\Leftrightarrow \qquad \{ \;\; \text{flipping} \;\; \}$

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

Universal-**foldl**

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

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

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 = $ **foldr** $\overset{\sim}{\tilde{f}}$ *hold?*

Universal-**foldl**

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

$$k = \textbf{foldl} \ f \quad \Leftrightarrow \quad \left\{ \begin{array}{l} k \ z \ \gamma = z \\ k \ z \ (\alpha \ x \ xs) = k \ (f \ z \ x) \ xs \end{array} \right. \tag{47}$$

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** $f =$ **foldr** $\widehat{\widetilde{f}}$ *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`),

$$\textbf{foldl } (\div) \ 100000 \ [99, 2, 7] = 72 = \textbf{foldr } (\tilde{\div}) \ 100000 \ [99, 2, 7]$$
$$\textbf{foldl } (\div) \ 10000 \ [99, 2, 7] = 7 = \textbf{foldr } (\tilde{\div}) \ 10000 \ [99, 2, 7]$$

and yet

- neither $(\div)$ nor $(\tilde{\div})$ 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:

$$\textbf{foldl } f = \textbf{foldr } \tilde{f}$$

$$\Leftrightarrow \qquad \{ \ (47) \ \}$$

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

$$\Leftrightarrow \qquad \{ \ \text{flipping } f \ z \ x \ \}$$

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

Back to the permutativity squares

Recall (36)

$$
\begin{array}{ccc}
B & \xleftarrow{\;s\,x\;} & B \\
{\scriptstyle s\,a}\downarrow & & \downarrow{\scriptstyle s\,a} \\
B & \xleftarrow{\;s\,x\;} & B
\end{array}
\quad\Rightarrow\quad
\begin{array}{ccc}
B & \xleftarrow{\;s\,x\;} & B \\
\textbf{foldr }s\downarrow & & \downarrow\textbf{foldr }s \\
B^{\mathbb{T}\,A} & \xleftarrow[(s\,x\cdot)]{} & B^{\mathbb{T}\,A}
\end{array}
$$

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

$$
\begin{array}{ccc}
B & \xleftarrow{\;\tilde{f}\,x\;} & B \\
{\scriptstyle \tilde{f}\,a}\downarrow & & \downarrow{\scriptstyle \tilde{f}\,a} \\
B & \xleftarrow[\tilde{f}\,x]{} & B
\end{array}
\quad\Rightarrow\quad
\begin{array}{ccc}
B & \xleftarrow{\;\tilde{f}\,x\;} & B \\
\textbf{foldr }\tilde{f}\downarrow & & \downarrow\textbf{foldr }\tilde{f} \\
B^{\mathbb{T}\,A} & \xleftarrow[(\tilde{f}\,x\cdot)]{} & B^{\mathbb{T}\,A}
\end{array}
$$

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

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

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

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

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

$\Leftrightarrow$ 　　　$\{$　definition of **foldr** (41)　$\}$

　　　True

Conclusion

We conclude that **foldl** $f =$ **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

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

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

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

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

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

$$
\begin{array}{ccc}
A & \xleftarrow{(\leqslant)} & A \\
{\scriptstyle (\times y)^{\circ}} \downarrow & = & \downarrow {\scriptstyle (\div y)} \\
B & \xleftarrow[(\leqslant)]{} & B
\end{array}
\qquad\qquad
a \times y \leqslant x \;\;\Leftrightarrow\;\; a \leqslant x \div y
\qquad (48)
$$

We can use (48) and **indirect equality** over $(\leqslant)$ 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

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

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

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

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

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

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

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

$::$ $\quad\{$ 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

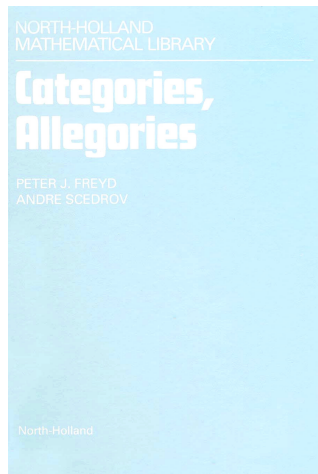Diffie-Hellman key exchange (Merkle, 1978)[1]:
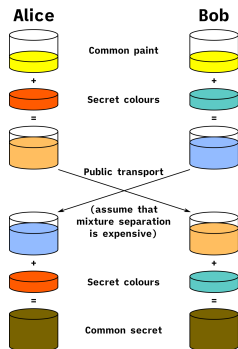
$$(+red) \cdot (+cyan) = (+cyan) \cdot (+red)$$

---

[1]Source: Wikipedia

Summary

Following the advice of Freyd & Ščedrov:

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

NORTH-HOLLAND
MATHEMATICAL LIBRARY

Categories, Allegories

PETER J. FREYD
ANDRE SCEDROV

North-Holland

Afterthought

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

$insert :: Ord\ a \Rightarrow a \rightarrow [a] \rightarrow [a]$

Thus **insertion sort**

foldr $insert$ []

computes the same as

foldl $\widetilde{insert}$ [].

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

Annex

On relational exponentials S^R

By vertical composition (2) one immediately infers:

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

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

By horizontal composition (3) we get

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

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

> **foldr** α *Nothing Nothing* = *Nothing*
> **foldr** α *Nothing* (*Just x*) = $\alpha\ x\ z$ = *Just x*

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

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.