“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

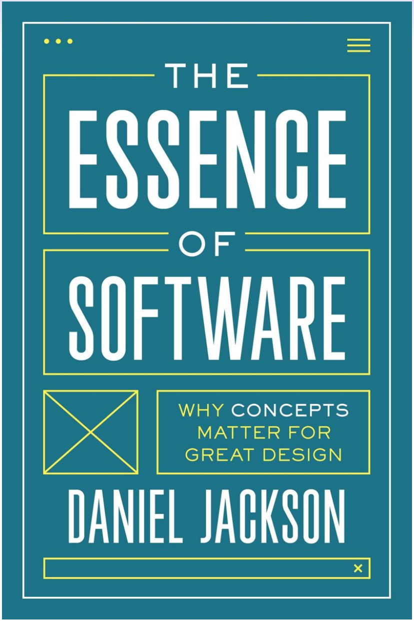
University of Minho & INESC TEC

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 old 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
WHY CONCEPTS
MATTER FOR
GREAT DESIGN
DANIEL JACKSON

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."

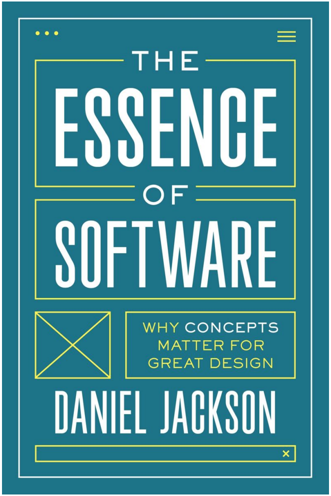
The Essence of Software

... **small number**

... **well defined**

... **easy to understand**

... **easy to use**



Freyd & Ščedrov, 1990

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

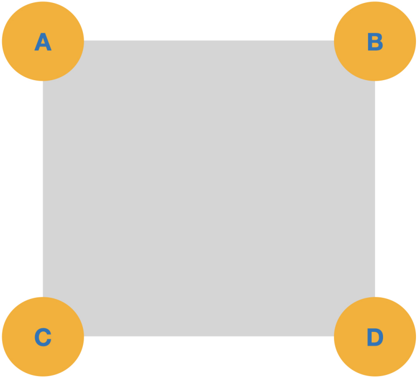
This calculus offers another, **often more amenable** framework for concepts and methods discussed in part one."

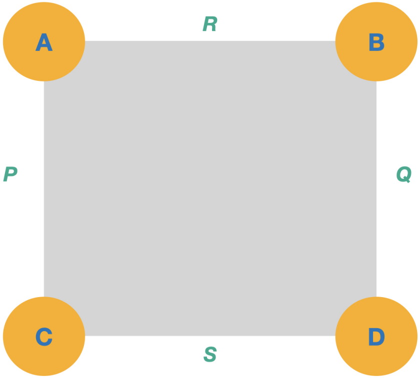
Freyd & Ščedrov, 1990

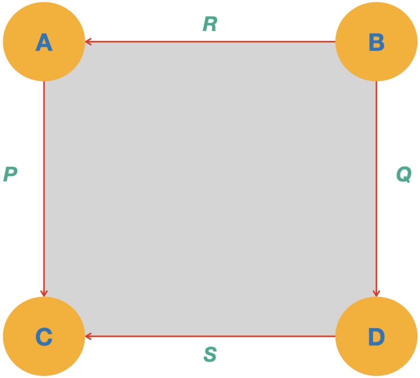
"(...) A special feature of our approach is a general **calculus of relations** presented in part two.

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

Squares


A
B
C
D


A
R
B
P
Q
C
S
D


A
R
B
P
Q
C
S
D

$$
\begin{tikzcd}
A \arrow[d, "P"'] \arrow[dr, phantom, "\subseteq"] & B \arrow[l, "R"'] \arrow[d, "Q"] \\
C & D \arrow[l, "S"]
\end{tikzcd}
$$

"Magic" square

$$\begin{tikzcd} A \arrow[d, "P"'] \arrow[dr, phantom, "P \cdot R \subseteq S \cdot Q"] & B \arrow[l, "R"'] \arrow[d, "Q"] \\ C & D \arrow[l, "S"] \end{tikzcd}$$

"Magic" square

Four binary relations:

$$\begin{tikzcd} A \arrow[d, "P"'] & B \arrow[l, "R"'] \arrow[d, "Q"] \arrow[dl, phantom, "\subseteq"] \\ C & D \arrow[l, "S"] \end{tikzcd} \qquad P \cdot R \subseteq S \cdot Q \tag{1}$$

Terminology:

R — *pre-producer*
P — *pre-consumer*
Q — *post-producer*
S — *post-consumer*

"Magic" square

Pointfree:

$$\begin{tikzcd}
A \arrow[d, "P"'] & B \arrow[l, "R"'] \arrow[d, "Q"] \arrow[dl, phantom, "\subseteq"] \\
C & D \arrow[l, "S"]
\end{tikzcd}
\qquad\qquad
P \cdot R \subseteq S \cdot Q$$

Pointwise:

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

"Magic" square

Pointfree:

$$\begin{tikzcd} A \arrow[d, "P"'] & B \arrow[l, "R"'] \arrow[d, "Q"] \arrow[dl, phantom, "\subseteq"] \\ C & D \arrow[l, "S"] \end{tikzcd}$$

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

Pointwise:

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

"Magic" square

Perhaps not what you were expecting...

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

$$\begin{tikzcd} A \arrow[d, "P"'] & B \arrow[l, "R"'] \arrow[d, "Q"] \arrow[dl, phantom, "\subseteq"] \\ C & D \arrow[l, "S"] \end{tikzcd}$$

Vertical composition

$$
\begin{tikzcd}
A & C \arrow[l, "R"'] \arrow[d, "Q"] \\
B \arrow[from=u, "P"'] \arrow[ur, phantom, "\subseteq"] & D \arrow[l, "S"] \arrow[d, "Q'"] \\
B' \arrow[from=u, "P'"'] \arrow[ur, phantom, "\subseteq"] & D' \arrow[l, "S'"]
\end{tikzcd}
\quad \Rightarrow \quad
\begin{tikzcd}
A & C \arrow[l, "R"'] \arrow[d, "Q' \cdot Q"] \\
B' \arrow[from=u, "P' \cdot P"'] \arrow[ur, phantom, "\subseteq"] & D' \arrow[l, "S'"]
\end{tikzcd}
\tag{2}
$$

Horizontal composition

$$
\begin{tikzcd}
A \arrow[d, "P"'] & C \arrow[l, "R"'] \arrow[d, "Q"] & C' \arrow[l, "R'"'] \arrow[d, "Q'"] \\
B \arrow[ur, phantom, "\subseteq"] & D \arrow[l, "S"] \arrow[ur, phantom, "\subseteq"] & D' \arrow[l, "S'"]
\end{tikzcd}
\Rightarrow
\begin{tikzcd}
A \arrow[d, "P"'] & C \arrow[l, "R \cdot R'"'] \arrow[d, "Q'"] \\
B' \arrow[ur, phantom, "\subseteq"] & D' \arrow[l, "S \cdot S'"]
\end{tikzcd}
\tag{3}
$$

Identity

$$\begin{tikzcd}
A \arrow[d, "id"'] \arrow[dr, phantom, "\subseteq"] & C \arrow[l, "R"'] \arrow[d, "id"] \\
A \arrow[d, "P"'] \arrow[dr, phantom, "\subseteq"] & C \arrow[l, "R"'] \arrow[d, "Q"] \\
B \arrow[d, "id"'] \arrow[dr, phantom, "\subseteq"] & D \arrow[l, "S"] \arrow[d, "id"] \\
B & D \arrow[l, "S"]
\end{tikzcd}
\quad \Leftrightarrow \quad
\begin{tikzcd}
A \arrow[d, "P"'] \arrow[dr, phantom, "\subseteq"] & C \arrow[l, "R"'] \arrow[d, "Q"] \\
B & D \arrow[l, "S"]
\end{tikzcd}$$

(Similarly for horizontal.)

Converse

$$
\begin{tikzcd}
A \arrow[d, "P"'] & C \arrow[l, "R"'] \arrow[d, "Q"] \\
B \arrow[ur, phantom, "\subseteq"] & D \arrow[l, "S"]
\end{tikzcd}
\quad \Leftrightarrow \quad
\begin{tikzcd}
A \arrow[d, "R^\circ"'] & B \arrow[l, "P^\circ"'] \arrow[d, "S^\circ"] \\
C \arrow[ur, phantom, "\subseteq"] & D \arrow[l, "Q^\circ"]
\end{tikzcd}
$$

The converse of a square is its "*passive voice*"

Square (direct) sums

$$
\begin{array}{c}
\begin{tikzcd}
A & C \arrow[l, "R"'] \arrow[d, "Q"] \\
B & D \arrow[l, "S"]
\arrow[from=1-1, to=2-1, "P"']
\arrow[from=1-1, to=2-2, phantom, "\subseteq"]
\end{tikzcd} \\
+ \\
\begin{tikzcd}
A' & C' \arrow[l, "R'"'] \arrow[d, "Q'"] \\
B' & D' \arrow[l, "S"]
\arrow[from=1-1, to=2-1, "P'"']
\arrow[from=1-1, to=2-2, phantom, "\subseteq"]
\end{tikzcd}
\end{array}
\quad = \quad
\begin{tikzcd}
A + A' & C + C' \arrow[l, "R+R'"'] \arrow[d, "Q+Q'"] \\
B + B' & D + D' \arrow[l, "S+S'"]
\arrow[from=1-1, to=2-1, "P+P'"']
\arrow[from=1-1, to=2-2, phantom, "\subseteq"]
\end{tikzcd}
$$

Functorial squares

Functor $\mathbb{F}$:

$$\begin{tikzcd} A \arrow[d, "P"'] \arrow[dr, phantom, "\subseteq"] & C \arrow[l, "R"'] \arrow[d, "Q"] \\ B & D \arrow[l, "S"] \end{tikzcd} \quad \Rightarrow \quad \begin{tikzcd} \mathbb{F}\ A \arrow[d, "\mathbb{F}\ P"'] \arrow[dr, phantom, "\subseteq"] & \mathbb{F}\ C \arrow[l, "\mathbb{F}\ R"'] \arrow[d, "\mathbb{F}\ Q"] \\ \mathbb{F}\ B & \mathbb{F}\ D \arrow[l, "\mathbb{F}\ S"] \end{tikzcd}$$

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

Squares with functions

Some relations f fit into the following squares:

$$
\begin{tikzcd}
A & A \arrow[l, "id"'] \arrow[d, "f"] & B \arrow[l, "f^\circ"'] \arrow[d, "id"] \\
A \arrow[u, leftarrow, "id"] \arrow[ur, phantom, "\subseteq"] & B \arrow[l, "f^\circ"] \arrow[ur, phantom, "\subseteq"] & B \arrow[l, "id"]
\end{tikzcd}
\tag{4}
$$

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:

$$\begin{tikzcd} A & A \arrow[l, "id"'] \arrow[d, "f"] & B \arrow[l, "f^\circ"'] \arrow[d, "id"] \\ A \arrow[from=u, "id"'] \arrow[ur, phantom, "\subseteq"] & B \arrow[l, "f^\circ"] \arrow[ur, phantom, "\subseteq"] & B \arrow[l, "id"] \end{tikzcd} \tag{4}$$

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:

$$\begin{tikzcd} A & A \arrow[l, "id"'] \arrow[d, "f"] & B \arrow[l, "f^\circ"'] \arrow[d, "id"] \\ A \arrow[from=u, "id"'] \arrow[ur, phantom, "\subseteq"] & B \arrow[l, "f^\circ"] \arrow[ur, phantom, "\subseteq"] & B \arrow[l, "id"] \end{tikzcd} \tag{4}$$

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:

$$\begin{tikzcd} A \arrow[d, "id"'] & A \arrow[l, "id"'] \arrow[d, "f"] \arrow[dl, phantom, "\subseteq"] & B \arrow[l, "f^\circ"'] \arrow[d, "id"] \arrow[dl, phantom, "\subseteq"] \\ A & B \arrow[l, "f^\circ"] & B \arrow[l, "id"] \end{tikzcd} \tag{4}$$

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:

$$
\begin{tikzcd}
A \arrow[d, "f"'] & C \arrow[l, "R"'] \arrow[d, "Q"] \arrow[dl, phantom, "\subseteq"] \\
B & B \arrow[l, "id"]
\end{tikzcd}
\quad \Leftrightarrow \quad
\begin{tikzcd}
A \arrow[d, "id"'] & C \arrow[l, "R"'] \arrow[d, "Q"] \arrow[dl, phantom, "\subseteq"] \\
A & B \arrow[l, "f^\circ"]
\end{tikzcd}
$$

This 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**:

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

Existential quantifiers go away:

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

$$\begin{tikzcd} B \arrow[r, "f"] & C & D \arrow[l, "R"'] & A \arrow[l, "g"'] \arrow[lll, bend left, "f^\circ \cdot R \cdot g"] \end{tikzcd}$$

☞ Very useful in practice ☺

Squares with functions

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

$$\begin{tikzcd} A \arrow[d, "f"'] \arrow[dr, phantom, "\subseteq"] & B \arrow[l, "R"'] \arrow[d, "g"] \\ C & D \arrow[l, "S"] \end{tikzcd} \qquad f \cdot R \subseteq S \cdot g \tag{9}$$

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

$$f \; S^R \; g \quad \Leftrightarrow \quad f \cdot R \subseteq S \cdot g \tag{10}$$

In words:

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

Squares with functions

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

$$\begin{tikzcd} A & A \arrow[l, "id"'] \arrow[d, "g"] \\ C & D \arrow[l, "(\leqslant)"] \arrow[from=1-1, to=2-1, "f"'] \arrow[from=1-1, to=2-2, phantom, "\subseteq"] \end{tikzcd} \qquad f \subseteq (\leqslant) \cdot g$$

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**:

$$\begin{tikzcd} A \arrow[d, "f"'] \arrow[dr, phantom, "\subseteq"] & B \arrow[l, "R"'] \arrow[d, "g"] \\ C & D \arrow[l, "S"] \end{tikzcd} \quad \text{that is to say,} \quad \frac{A \xleftarrow{R} B \qquad C \xleftarrow{S} D}{C^A \xleftarrow{S^R} D^B}$$

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

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

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

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:

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

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):

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

and so and so forth.

☞ Rich construction! ☺

Prosaic applications

"Chase" the squares:

$$
\begin{tikzcd}
\mathit{Client} & \mathit{Sale} \arrow[l, "\mathit{client}"'] \arrow[ddr, "\mathit{date}"] & \mathit{SoldProduct} \arrow[l, "\mathit{sale}"'] \arrow[d, "\mathit{product}"'] \\
 & \mathit{Cupon} \arrow[ul, "\mathit{owner}"] \arrow[u, "\mathit{Used}"'] \arrow[dr, "\mathit{expiry}"'] & \mathit{Product} \\
 & & \mathit{Date} \arrow[loop below, "(\leqslant)"]
\end{tikzcd}
$$

"Chase" the squares

Pick

$$\begin{tikzcd} Client & & Sale \arrow[ll, "client"'] \\ & & Cupon \arrow[ull, "owner"] \arrow[u, "Used"'] \end{tikzcd}$$

and try orienting it:

$$\begin{tikzcd} Sale \arrow[d, "client"'] & Cupon \arrow[l, "Used"'] \arrow[d, "owner"] \arrow[dl, phantom, "\subseteq"] \\ Client & Client \arrow[l, "id"] \end{tikzcd}$$

Indeed:

Coupons can only be used by clients who own them.

"Chase" the squares

Now this one:

$$
\begin{tikzcd}
Sale \arrow[rrdd, "date"] & & \\
Cupon \arrow[u, "Used"'] \arrow[rrd, "expiry"'] & & \\
& & Date \arrow[loop below, "(\leqslant)"]
\end{tikzcd}
\qquad
\begin{tikzcd}
Sale \arrow[d, "date"'] \arrow[dr, phantom, "\subseteq"] & Cupon \arrow[l, "Used"'] \arrow[d, "expiry"] \\
Date & Date \arrow[l, "(\leqslant)"]
\end{tikzcd}
$$

Coupons cannot be used beyond their expiry date.

Formal modelling

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

$$
\begin{tikzcd}
OpenFileDescriptor \arrow[d, "path"'] & FileHandle \arrow[l, "table\ \ s"'] \arrow[d, "\top"] \arrow[dl, phantom, "\subseteq"] & \\
Path \arrow[r, shift right, "fileStore\ \ s"'] & File \arrow[l, shift right, "(fileStore\ \ s)^{\circ}"'] \arrow[r, "attributes"] & Attributes \arrow[d, "fileType"] \\
Path \arrow[u, "dirName"] \arrow[r, "fileStore\ \ s"'] & File \arrow[u, phantom, "\rotatebox[origin=c]{90}{$\subseteq$}"] \arrow[r, "\underline{Directory}"'] & FileType
\end{tikzcd}
$$

(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. $R \to S^Q$:

$$\begin{tikzcd} A \arrow[d, "f"'] & B \arrow[l, "R"'] \arrow[d, "g"] \\ X^C \arrow[ur, phantom, "\subseteq"] & Y^D \arrow[l, "S^Q"] \end{tikzcd} \qquad f\ (R \to S^Q)\ g$$

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

Higher-order Reynolds squares

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

$\Leftrightarrow$ { Reynolds square (9) }

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

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

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

$\Leftrightarrow$ { (9) again }

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

$\Leftrightarrow$ { (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 $f = g$:

$$\begin{tikzcd} A \arrow[d, "f"'] \arrow[dr, phantom, "\subseteq"] & A \arrow[l, "R"'] \arrow[d, "f"] \\ C & C \arrow[l, "S"] \end{tikzcd} \qquad f \cdot R \subseteq S \cdot f \tag{21}$$

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 $R \rightarrow S$ 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**:

$$\begin{tikzcd}
\mathbb{F}\,A \arrow[d, "\alpha"'] \arrow[dr, phantom, "\subseteq"] & \mathbb{F}\,B \arrow[l, "\mathbb{F}\,R"'] \arrow[d, "\beta"] \\
A & B \arrow[l, "R"]
\end{tikzcd}$$

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 ($R := h$),

$$\begin{tikzcd} \mathbb{F}\ A \arrow[d, "\alpha"'] \arrow[dr, phantom, "="] & \mathbb{F}\ B \arrow[l, "\mathbb{F}\ h"'] \arrow[d, "\beta"] \\ B & A \arrow[l, "h"] \end{tikzcd}$$

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**:

$$\begin{tikzcd} A \arrow[d, "\gamma"'] \arrow[dr, phantom, "\subseteq"] & B \arrow[l, "R"'] \arrow[d, "\phi"] \\ \mathbb{F}\ A & \mathbb{F}\ B \arrow[l, "\mathbb{F}\ R"] \end{tikzcd}$$

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:

$$\begin{tikzcd} A \arrow[d, "f^\circ"'] \arrow[dr, phantom, "="] & A \arrow[l, "(\sqsubseteq)"'] \arrow[d, "g"] \\ B & B \arrow[l, "(\leqslant)"] \end{tikzcd}$$

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

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

In this very special situation, f and g in

$$\begin{tikzcd} (A, \sqsubseteq) \arrow[r, bend left, "g"] & (B, \leqslant) \arrow[l, bend left, "f"] \end{tikzcd}$$

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:

$$\begin{tikzcd} A \arrow[d, "f^\circ"'] \arrow[dr, phantom, "="] & A \arrow[l, "(\sqsubseteq)"'] \arrow[d, "g"] \\ B & B \arrow[l, "(\leqslant)"] \end{tikzcd}$$

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

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

In this very special situation, f and g in

$$\begin{tikzcd} (A, \sqsubseteq) \arrow[r, bend left, "g"] & (B, \leqslant) \arrow[l, bend left, "f"] \end{tikzcd}$$

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:

$$\begin{tikzcd}
\mathbb{F}\ A \arrow[d, "f"'] & \mathbb{F}\ B \arrow[l, "\mathbb{F}\ R"'] \arrow[d, "f"] \\
\mathbb{G}\ A \arrow[ur, phantom, "\subseteq"] & \mathbb{G}\ B \arrow[l, "\mathbb{G}\ R"]
\end{tikzcd}$$

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

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:

$$
\begin{tikzcd}
A \arrow[d, "\mathtt{const}"'] \arrow[dr, phantom, "\subseteq"] & C \arrow[l, "R"'] \arrow[d, "\mathtt{const}"] \\
A^B & C^D \arrow[l, "R^S"]
\end{tikzcd}
\qquad \mathtt{const} \cdot R \subseteq R^S \cdot \mathtt{const} \tag{25}
$$

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

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:

$$
\begin{tikzcd}
A & C \arrow[l, "R"'] \arrow[d, "\mathtt{const}"] \\
A^B \arrow[from=u, "\mathtt{const}"'] \arrow[ur, phantom, "\subseteq"] & C^D \arrow[l, "R^S"]
\end{tikzcd}
\qquad \mathtt{const} \cdot R \;\subseteq\; R^S \cdot \mathtt{const} \qquad (25)
$$

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

$$
\begin{tikzcd}
A \arrow[d, "\mathtt{const}"'] \arrow[dr, phantom, "\subseteq"] & C \arrow[l, "R"'] \arrow[d, "\mathtt{const}"] \\
A^B & C^D \arrow[l, "R^S"]
\end{tikzcd}
\quad \xrightarrow{R := id} \quad
\begin{tikzcd}
A \arrow[d, "\mathtt{const}"'] \arrow[dr, phantom, "\subseteq"] & A \arrow[l, "id"'] \arrow[d, "\mathtt{const}"] \\
A^B & A^D \arrow[l, "id^S"]
\end{tikzcd}
$$

$$\updownarrow$$

$$
\begin{tikzcd}
B \arrow[d, "id"'] \arrow[dr, phantom, "\subseteq"] & D \arrow[l, "S"'] \arrow[d, "\mathtt{const}\ a"] \\
B & A \arrow[l, "\mathtt{const}\ a^{\circ}"]
\end{tikzcd}
\quad \longleftrightarrow \quad
\begin{tikzcd}
B \arrow[d, "\mathtt{const}\ a"'] \arrow[dr, phantom, "\subseteq"] & D \arrow[l, "S"'] \arrow[d, "\mathtt{const}\ a"] \\
A & A \arrow[l, "id"]
\end{tikzcd}
$$

Free theorem squares

$$
\begin{tikzcd}
A \arrow[d, "\mathtt{const}"'] \arrow[dr, phantom, "\subseteq"] & C \arrow[l, "R"'] \arrow[d, "\mathtt{const}"] \\
A^B & C^D \arrow[l, "R^S"]
\end{tikzcd}
\quad \xrightarrow{R := id} \quad
\begin{tikzcd}
A \arrow[d, "\mathtt{const}"'] \arrow[dr, phantom, "\subseteq"] & A \arrow[l, "id"'] \arrow[d, "\mathtt{const}"] \\
A^B & A^D \arrow[l, "id^S"]
\end{tikzcd}
$$

$$
\updownarrow
$$

$$
\begin{tikzcd}
B \arrow[d, "id"'] \arrow[dr, phantom, "\subseteq"] & D \arrow[l, "S"'] \arrow[d, "\mathtt{const}\ a"] \\
B & A \arrow[l, "\mathtt{const}\ a^{\circ}"]
\end{tikzcd}
\quad \longleftrightarrow \quad
\begin{tikzcd}
B \arrow[d, "\mathtt{const}\ a"'] \arrow[dr, phantom, "\subseteq"] & D \arrow[l, "S"'] \arrow[d, "\mathtt{const}\ a"] \\
A & A \arrow[l, "id"]
\end{tikzcd}
$$

Free theorem squares

$$\begin{tikzcd} A \arrow[d, "\mathtt{const}"'] \arrow[dr, phantom, "\subseteq"] & C \arrow[l, "R"'] \arrow[d, "\mathtt{const}"] \\ A^B & C^D \arrow[l, "R^S"] \end{tikzcd} \quad \xrightarrow{R := id} \quad \begin{tikzcd} A \arrow[d, "\mathtt{const}"'] \arrow[dr, phantom, "\subseteq"] & A \arrow[l, "id"'] \arrow[d, "\mathtt{const}"] \\ A^B & A^D \arrow[l, "id^S"] \end{tikzcd}$$

$$\updownarrow$$

$$\begin{tikzcd} B \arrow[d, "id"'] \arrow[dr, phantom, "\subseteq"] & D \arrow[l, "S"'] \arrow[d, "\mathtt{const}\ a"] \\ B & A \arrow[l, "\mathtt{const}\ a^{\circ}"] \end{tikzcd} \quad \longleftrightarrow \quad \begin{tikzcd} B \arrow[d, "\mathtt{const}\ a"'] \arrow[dr, phantom, "\subseteq"] & D \arrow[l, "S"'] \arrow[d, "\mathtt{const}\ a"] \\ A & A \arrow[l, "id"] \end{tikzcd}$$

Free theorem squares

$$
\begin{tikzcd}
A \arrow[d, "\texttt{const}"'] & C \arrow[l, "R"'] \arrow[d, "\texttt{const}"] \arrow[dl, phantom, "\subseteq"] \\
A^B & C^D \arrow[l, "R^S"]
\end{tikzcd}
\quad \xrightarrow{R := id} \quad
\begin{tikzcd}
A \arrow[d, "\texttt{const}"'] & A \arrow[l, "id"'] \arrow[d, "\texttt{const}"] \arrow[dl, phantom, "\subseteq"] \\
A^B & A^D \arrow[l, "id^S"]
\end{tikzcd}
$$

$$\updownarrow$$

$$
\begin{tikzcd}
B \arrow[d, "id"'] & D \arrow[l, "S"'] \arrow[d, "\texttt{const}\ a"] \arrow[dl, phantom, "\subseteq"] \\
B & A \arrow[l, "\texttt{const}\ a^{\circ}"]
\end{tikzcd}
\quad \longleftrightarrow \quad
\begin{tikzcd}
B \arrow[d, "\texttt{const}\ a"'] & D \arrow[l, "S"'] \arrow[d, "\texttt{const}\ a"] \arrow[dl, phantom, "\subseteq"] \\
A & A \arrow[l, "id"]
\end{tikzcd}
$$

Free theorem squares

$$\begin{tikzcd} B \arrow[d, "id"'] \arrow[dr, phantom, "\subseteq"] & D \arrow[l, "S"'] \arrow[d, "\mathtt{const}\ a"] \\ B & A \arrow[l, "\mathtt{const}\ a^{\circ}"] \end{tikzcd}$$

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

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

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

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

On Injectivity

Injective functions fit in the following square:

$$\begin{tikzcd} B \arrow[d, "f^\circ"'] & A \arrow[l, "f"'] \arrow[d, "id"] \arrow[dl, phantom, "\subseteq"] \\ A & A \arrow[l, "id"] \end{tikzcd}$$

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

$$
\begin{tikzcd}
B \arrow[d, "\mathtt{const}\ a"'] \arrow[dr, phantom, "\subseteq"] & D \arrow[l, "S"'] \arrow[d, "\mathtt{const}\ a"] \\
A & A \arrow[l, "id"]
\end{tikzcd}
\xrightarrow{S := h}
\begin{tikzcd}
B \arrow[d, "\mathtt{const}\ a"'] \arrow[dr, phantom, "="] & D \arrow[l, "h"'] \arrow[d, "\mathtt{const}\ a"] \\
A & A \arrow[l, "id"]
\end{tikzcd}
$$

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

$$
\begin{tikzcd}
A \arrow[d, "\mathtt{const}"'] \arrow[dr, phantom, "\subseteq"] & C \arrow[l, "R"'] \arrow[d, "\mathtt{const}"] \\
A^B & C^D \arrow[l, "R^S"]
\end{tikzcd}
\xrightarrow{R, S := h, id}
\begin{tikzcd}
A \arrow[d, "\mathtt{const}"'] \arrow[dr, phantom, "="] & C \arrow[l, "h"'] \arrow[d, "\mathtt{const}"] \\
A^B & C^D \arrow[l, "h^{id}"]
\end{tikzcd}
$$

$$h \cdot (\mathtt{const}\ a) = \mathtt{const}\ (h\ a)$$

Free theorem squares

$$\begin{tikzcd} B & D \arrow[l, "S"'] \arrow[d, "\mathtt{const}\ a"] \\ A \arrow[from=u, "\mathtt{const}\ a"'] \arrow[ur, phantom, "\subseteq"] & A \arrow[l, "id"] \end{tikzcd} \quad \xrightarrow{S := h} \quad \begin{tikzcd} B & D \arrow[l, "h"'] \arrow[d, "\mathtt{const}\ a"] \\ A \arrow[from=u, "\mathtt{const}\ a"'] \arrow[ur, phantom, "="] & A \arrow[l, "id"] \end{tikzcd}$$

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

$$\begin{tikzcd} A & C \arrow[l, "R"'] \arrow[d, "\mathtt{const}"] \\ A^B \arrow[from=u, "\mathtt{const}"'] \arrow[ur, phantom, "\subseteq"] & C^D \arrow[l, "R^S"] \end{tikzcd} \quad \xrightarrow{R, S := h, id} \quad \begin{tikzcd} A & C \arrow[l, "h"'] \arrow[d, "\mathtt{const}"] \\ A^B \arrow[from=u, "\mathtt{const}"'] \arrow[ur, phantom, "="] & C^D \arrow[l, "h^{id}"] \end{tikzcd}$$

$$h \cdot (\mathtt{const}\ a) = \mathtt{const}\ (h\ a)$$

Free theorem squares

$$\begin{tikzcd} B & D \arrow[l, "S"'] \arrow[d, "\mathtt{const}\ a"] \\ A \arrow[from=u, "\mathtt{const}\ a"'] \arrow[ur, phantom, "\subseteq"] & A \arrow[l, "id"] \end{tikzcd} \quad \xrightarrow{S := h} \quad \begin{tikzcd} B & D \arrow[l, "h"'] \arrow[d, "\mathtt{const}\ a"] \\ A \arrow[from=u, "\mathtt{const}\ a"'] \arrow[ur, phantom, "="] & A \arrow[l, "id"] \end{tikzcd}$$

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

$$\begin{tikzcd} A & C \arrow[l, "R"'] \arrow[d, "\mathtt{const}"] \\ A^B \arrow[from=u, "\mathtt{const}"'] \arrow[ur, phantom, "\subseteq"] & C^D \arrow[l, "R^S"] \end{tikzcd} \quad \xrightarrow{R, S := h, id} \quad \begin{tikzcd} A & C \arrow[l, "h"'] \arrow[d, "\mathtt{const}"] \\ A^B \arrow[from=u, "\mathtt{const}"'] \arrow[ur, phantom, "="] & C^D \arrow[l, "h^{id}"] \end{tikzcd}$$

$$h \cdot (\mathtt{const}\ a) = \mathtt{const}\ (h\ a)$$

Free theorem squares

$$\begin{tikzcd} B \arrow[d, "\mathtt{const}\ a"'] & D \arrow[l, "S"'] \arrow[d, "\mathtt{const}\ a"] \arrow[dl, phantom, "\subseteq"] \\ A & A \arrow[l, "id"] \end{tikzcd} \xrightarrow{S := h} \begin{tikzcd} B \arrow[d, "\mathtt{const}\ a"'] & D \arrow[l, "h"'] \arrow[d, "\mathtt{const}\ a"] \arrow[dl, phantom, "="] \\ A & A \arrow[l, "id"] \end{tikzcd}$$

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

$$\begin{tikzcd} A \arrow[d, "\mathtt{const}"'] & C \arrow[l, "R"'] \arrow[d, "\mathtt{const}"] \arrow[dl, phantom, "\subseteq"] \\ A^B & C^D \arrow[l, "R^S"] \end{tikzcd} \xrightarrow{R, S := h, id} \begin{tikzcd} A \arrow[d, "\mathtt{const}"'] & C \arrow[l, "h"'] \arrow[d, "\mathtt{const}"] \arrow[dl, phantom, "="] \\ A^B & C^D \arrow[l, "h^{id}"] \end{tikzcd}$$

$$h \cdot (\mathtt{const}\ a) = \mathtt{const}\ (h\ a)$$

Free theorem squares

Example:

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

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

$$g\ (R \to Q^S)\ f \quad \Rightarrow \quad (\texttt{flip}\ g)\ (S \to Q^R)\ (\texttt{flip}\ f)$$

that is:

$$\begin{tikzcd} A \arrow[d, "g"'] & X \arrow[l, "R"'] \arrow[d, "f"] \arrow[dl, phantom, "\subseteq"] \\ C^B & Z^Y \arrow[l, "Q^S"] \end{tikzcd} \quad \Rightarrow \quad \begin{tikzcd} B \arrow[d, "\widetilde{g}"'] & Y \arrow[l, "S"'] \arrow[d, "\widetilde{f}"] \arrow[dl, phantom, "\subseteq"] \\ C^A & Z^X \arrow[l, "Q^R"] \end{tikzcd} \tag{28}$$

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

Free theorem squares

For $Q := id, S := id \text{ and } R := r$ (a function):

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

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:

$$\begin{array}{l} \textbf{foldl} :: \mathit{Foldable}\ t \Rightarrow (b \rightarrow a \rightarrow b) \rightarrow b \rightarrow t\ a \rightarrow b \\ \textbf{foldr} :: \mathit{Foldable}\ t \Rightarrow (a \rightarrow b \rightarrow b) \rightarrow b \rightarrow t\ a \rightarrow b \end{array} \tag{29}$$

(Hackage's Data.Foldable)

foldl and **foldr** squares

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

$$\textbf{foldl} : (S \rightarrow S^R) \rightarrow (S \rightarrow S^{\mathbb{T}\ R}) \tag{30}$$
$$\textbf{foldr} : (R \rightarrow S^S) \rightarrow (S \rightarrow S^{\mathbb{T}\ R}) \tag{31}$$

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:

$$\begin{tikzcd} B \arrow[d, "g"'] & Y \arrow[l, "s"'] \arrow[d, "f"] \arrow[dl, phantom, "\subseteq"] \\ B^{A} & Y^{X} \arrow[l, "s^{R}"] \end{tikzcd} \quad \Rightarrow \quad \begin{tikzcd} B \arrow[d, "\textbf{foldl}\ g"'] & Y \arrow[l, "s"'] \arrow[d, "\textbf{foldl}\ f"] \arrow[dl, phantom, "\subseteq"] \\ B^{\mathbb{T}\ A} & Y^{\mathbb{T}\ X} \arrow[l, "s^{\mathbb{T}\ R}"] \end{tikzcd} \tag{32}$$

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:

$$\begin{tikzcd} B & Y \arrow[l, "h"'] \arrow[d, "f"] \\ B^A \arrow[from=u, "g"'] & Y^A \arrow[l, "(h\cdot)"] \end{tikzcd} \quad \Rightarrow \quad \begin{tikzcd} B \arrow[d, "\mathbf{foldl}\ g"'] & Y \arrow[l, "h"'] \arrow[d, "\mathbf{foldl}\ f"] \\ B^{\mathbb{T}\ A} & Y^{\mathbb{T}\ A} \arrow[l, "(h\cdot)"] \end{tikzcd}$$

Pointwise:

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

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

---

foldr

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

$$
\begin{tikzcd}
A \arrow[d, "g"'] \arrow[dr, phantom, "\subseteq"] & X \arrow[l, "R"'] \arrow[d, "f"] \\
B^B & Y^Y \arrow[l, "S^S"]
\end{tikzcd}
\quad \Rightarrow \quad
\begin{tikzcd}
B \arrow[d, "\mathsf{foldr}\ g"'] \arrow[dr, phantom, "\subseteq"] & Y \arrow[l, "S"'] \arrow[d, "\mathsf{foldr}\ f"] \\
B^{\mathbb{T}\ A} & Y^{\mathbb{T}\ X} \arrow[l, "S^{\mathbb{T}\ R}"]
\end{tikzcd}
\tag{33}
$$

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

$$\begin{tikzcd} A \arrow[d, "g"'] & A \arrow[l, "id"'] \arrow[d, "f"] \\ B^B & Y^Y \arrow[l, "h^h"] \end{tikzcd} \quad \Rightarrow \quad \begin{tikzcd} B \arrow[d, "\mathbf{foldr}\ g"'] & Y \arrow[l, "h"'] \arrow[d, "\mathbf{foldr}\ f"] \\ B^{\mathbb{T}\ A} & Y^{\mathbb{T}\ A} \arrow[l, "(h\cdot)"] \end{tikzcd}$$

where the side-condition square unfolds to:

$$\begin{array}{ll} & g\ (id \to h^h)\ f \\ \Leftrightarrow & \{\ (11)\ \} \\ & (g\ x)\ h^h\ (f\ x) \\ \Leftrightarrow & \{\ (13)\ \} \\ & (g\ x) \cdot h = h \cdot (f\ x) \end{array}$$

foldr squares

Altogether, one gets:

$$\begin{tikzcd} B & Y \arrow[l, "h"'] \\ B \arrow[u, leftarrow, "g\ x"'] & Y \arrow[l, "h"] \arrow[u, leftarrow, "f\ x"] \end{tikzcd} \quad \Rightarrow \quad \begin{tikzcd} B & Y \arrow[l, "h"'] \\ B^{\mathbb{T}\ A} \arrow[u, leftarrow, "\mathbf{foldr}\ g"'] & Y^{\mathbb{T}\ A} \arrow[l, "(h\cdot)"] \arrow[u, leftarrow, "\mathbf{foldr}\ f"] \end{tikzcd}$$

that is:

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

Going fully pointwise,

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

---

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 $h := s\ a$

$$
\begin{tikzcd}
B \arrow[d, "s\ x"'] & B \arrow[l, "s\ a"'] \arrow[d, "s\ x"] \\
B & B \arrow[l, "s\ a"]
\end{tikzcd}
\quad \Rightarrow \quad
\begin{tikzcd}
B \arrow[d, "\textbf{foldr}\ s"'] & B \arrow[l, "s\ a"'] \arrow[d, "\textbf{foldr}\ s"] \\
B^{\mathbb{T}\ A} & B^{\mathbb{T}\ A} \arrow[l, "(s\ a\cdot)"]
\end{tikzcd}
$$

Then (34) becomes:

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

Property

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

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

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

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

Looking at

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

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

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

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):

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

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,

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

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

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

But... no law relating both

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

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

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,

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

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

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

Towards **foldr**-universal

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

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

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

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

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

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

Towards **foldr**-universal

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

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

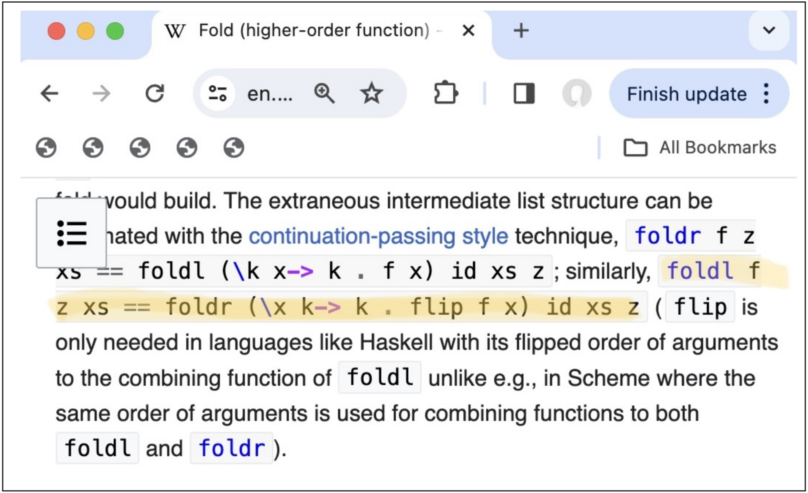
Altogether, (40) and (42) make up a universal property:

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

(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 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)

Wikipedia

That is,

$$\widetilde{\textbf{foldl}\ f} = \textbf{foldr}\ (\lambda x\ k \to k \cdot \tilde{f}\ x)\ id \tag{44}$$

or

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

cf. the square

$$\begin{tikzcd} B^B \arrow[d, "(\cdot k)"'] & A \arrow[l, "\theta\ f"'] \arrow[d, "\tilde{f}"] \\ B^B & B^B \arrow[l, "(k\cdot)"] \end{tikzcd} \tag{46}$$

Ok?

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

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

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

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

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

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

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

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

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:

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

Universal-**foldl**

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

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

$\Leftrightarrow$ $\{$ $(\theta\ f)\ x\ g = g \cdot (\widetilde{f}\ x)$ (46) $\}$

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

$\Leftrightarrow$ $\{$ 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**:

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

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 $\mathbf{foldl}\ f = \mathbf{foldr}\ \tilde{f}$ *hold?*

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

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

and yet

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

$$
\begin{array}{ll}
& \mathbf{foldl}\ f = \mathbf{foldr}\ \tilde{f} \\
\Leftrightarrow & \qquad \{\ (47)\ \} \\
& \begin{cases} \mathbf{foldr}\ \tilde{f}\ z\ \gamma = z \\ \mathbf{foldr}\ \tilde{f}\ z\ (\alpha\ x\ xs) = \mathbf{foldr}\ \tilde{f}\ (f\ z\ x)\ xs \end{cases} \\
\Leftrightarrow & \qquad \{\ \text{flipping } f\ z\ x\ \} \\
& \begin{cases} \mathbf{foldr}\ \tilde{f}\ z\ \gamma = z \\ \mathbf{foldr}\ \tilde{f}\ z\ (\alpha\ x\ xs) = \mathbf{foldr}\ \tilde{f}\ (\tilde{f}\ x\ z)\ xs \end{cases}
\end{array}
$$

Back to the permutativity squares

Recall (36)

$$
\begin{tikzcd}
B & B \arrow[l, "s\ x"'] \\
B \arrow[from=u, "s\ a"'] & B \arrow[l, "s\ x"] \arrow[from=u, "s\ a"]
\end{tikzcd}
\quad \Rightarrow \quad
\begin{tikzcd}
B & B \arrow[l, "s\ x"'] \\
B^{\mathbb{T}\ A} \arrow[from=u, "\textbf{foldr}\ s"'] & B^{\mathbb{T}\ A} \arrow[l, "(s\ x \cdot)"] \arrow[from=u, "\textbf{foldr}\ s"]
\end{tikzcd}
$$

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

$$
\begin{tikzcd}
B & B \arrow[l, "\widetilde{f}\ x"'] \\
B \arrow[from=u, "\widetilde{f}\ a"'] & B \arrow[l, "\widetilde{f}\ x"] \arrow[from=u, "\widetilde{f}\ a"]
\end{tikzcd}
\quad \Rightarrow \quad
\begin{tikzcd}
B & B \arrow[l, "\widetilde{f}\ x"'] \\
B^{\mathbb{T}\ A} \arrow[from=u, "\textbf{foldr}\ \widetilde{f}"'] & B^{\mathbb{T}\ A} \arrow[l, "(\widetilde{f}\ x \cdot)"] \arrow[from=u, "\textbf{foldr}\ \widetilde{f}"]
\end{tikzcd}
$$

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

Equating foldl and foldr

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

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

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

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

True

Conclusion

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

Back to

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

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

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**:

$$\begin{tikzcd} A \arrow[d, "(\times y)^\circ"'] \arrow[dr, phantom, "="] & A \arrow[l, "(\leqslant)"'] \arrow[d, "(\div y)"] \\ B & B \arrow[l, "(\leqslant)"] \end{tikzcd} \qquad a \times y \leqslant x \quad \Leftrightarrow \quad a \leqslant x \div y \qquad (48)$$

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

$$
\begin{array}{ll}
& y \leqslant (x \div b) \div a \\
\Leftrightarrow & \qquad \{ \text{ Galois connection (48) twice } \} \\
& (y \times a) \times b \leqslant x \\
\Leftrightarrow & \qquad \{ \text{ } (\times) \text{ is } \textbf{associative} \text{ and } \textbf{commutative} \text{ } \} \\
& (y \times b) \times a \leqslant x \\
\Leftrightarrow & \qquad \{ \text{ Galois connection (48) twice in the opposite direction } \} \\
& y \leqslant (x \div a) \div b \\
:: & \qquad \{ \text{ by indirect equality (Dijkstra, 2001) } \} \\
& (x \div b)\ a = (x \div a) \div b
\end{array}
$$

Summary

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

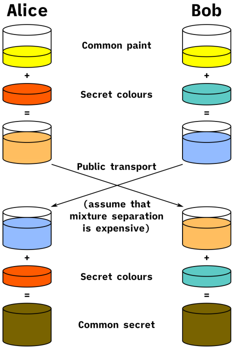
Danvy's reasoning is, however, quite different: he **postulates** permutativity as side condition and then **proves** it in Coq by **list induction**.

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."*

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**

$$\mathbf{foldr}\ insert\ [\,]$$

computes the same as

$$\mathbf{foldl}\ \widetilde{insert}\ [\,].$$

(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

$$
\begin{aligned}
&\textbf{instance } \mathit{Foldable}\ \mathbb{M}\ \textbf{where}\\
&\quad \mathit{foldMap} = \mathit{maybe}\ \mathit{mempty}\\
&\quad \textbf{foldr}\ \_\ z\ \mathit{Nothing} = z\\
&\quad \textbf{foldr}\ f\ z\ (\mathit{Just}\ x) = f\ x\ z\\
&\quad \textbf{foldl}\ \_\ z\ \mathit{Nothing} = z\\
&\quad \textbf{foldl}\ f\ z\ (\mathit{Just}\ x) = f\ z\ x
\end{aligned}
$$

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

$$
\begin{aligned}
&\textbf{foldr}\ \alpha\ \mathit{Nothing}\ \mathit{Nothing} = \mathit{Nothing}\\
&\textbf{foldr}\ \alpha\ \mathit{Nothing}\ (\mathit{Just}\ x) = \alpha\ x\ z = \mathit{Just}\ x
\end{aligned}
$$

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

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.