How much is in a square? Calculating functional programs with squares

JFP journal-first paper

ICFP 2025 — Singapore, 15 October 2025

J.N. Oliveira

Univ. Minho & Inesc tec (Portugal)

Summary of the talk

- **Simplicity** (eventually) wins
- **Widening** scope (usually) helps

Summary of the talk

- **Simplicity** (eventually) wins
- **Widening** scope (usually) helps

Simplicity (eventually) wins

"Simplicity does not precede complexity, but follows it."

(Edsger Dijkstra)

Widening context helps

Complex exponentials can simplify trigonometry, because they are mathematically easier to manipulate than their sine and cosine components. One technique is simply to convert sines and cosines into equivalent expressions in terms of exponentials sometimes called *complex sinusoids*.[13] After the manipulations, the simplified result is still real-valued. For example:

$$\begin{aligned}\cos x \cos y &= \frac{e^{ix}+e^{-ix}}{2}\cdot\frac{e^{iy}+e^{-iy}}{2}\\ &= \frac{1}{2}\cdot\frac{e^{i(x+y)}+e^{i(x-y)}+e^{i(-x+y)}+e^{i(-x-y)}}{2}\\ &= \frac{1}{2}\left(\frac{e^{i(x+y)}+e^{-i(x+y)}}{2}+\frac{e^{i(x-y)}+e^{-i(x-y)}}{2}\right)\\ &= \frac{1}{2}\left(\cos(x+y)+\cos(x-y)\right).\end{aligned}$$

(Ref: Wikipedia $\mapsto$ Euler's formula)

Freyd & Ščedrov, 1990

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

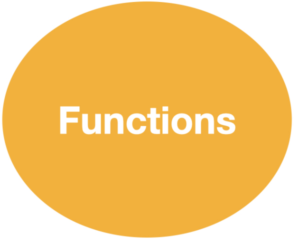
This calculus offers another, **often more amenable** framework for concepts and methods discussed in part one."

Freyd & Ščedrov, 1990

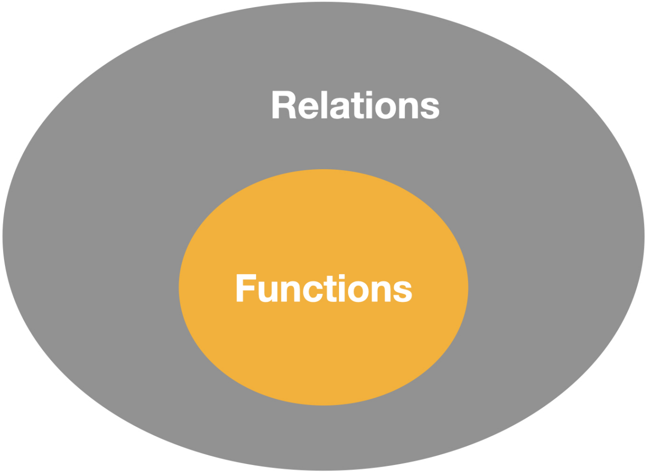
"(...) A special feature of our approach is a general **calculus of relations** presented in part two.

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

Functions



Functions $\subseteq$ Relations



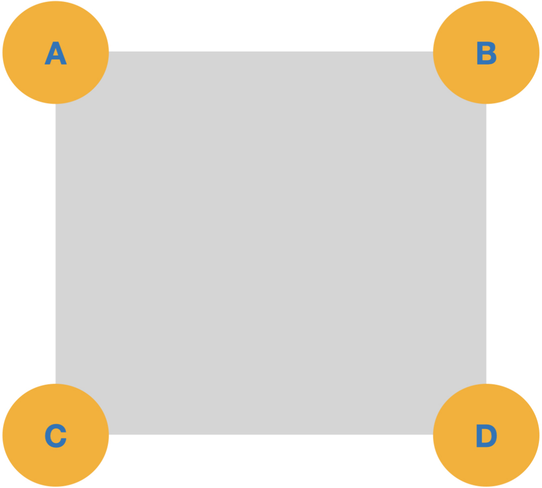
Braga, June 2003

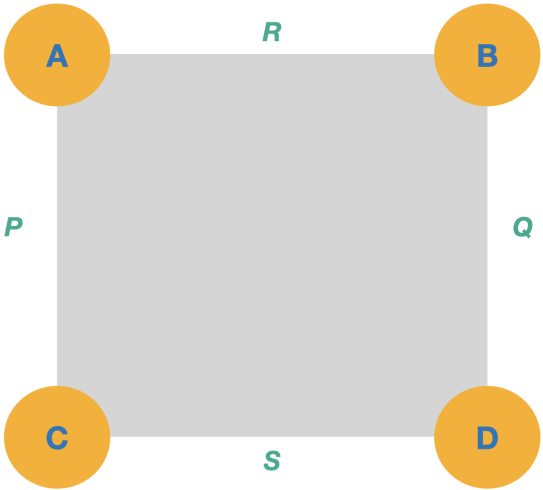
JNO - *"(...) What I find lacking in functional programming practice is formal specification..."*

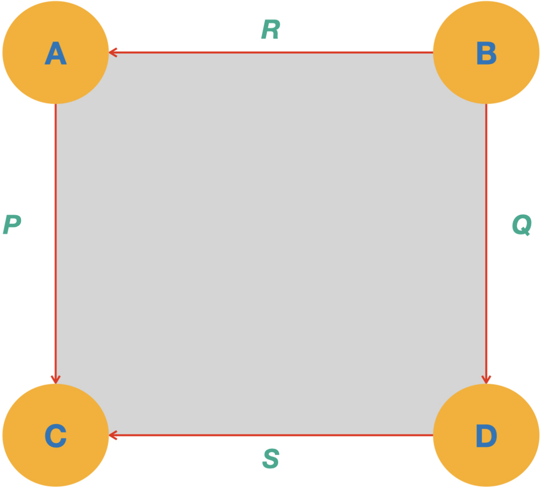
SPJ - *"But, types are... the formal specifications, aren't they?"*

It took me 20+ years to fully appreciate this answer!

Squares


A
B
C
D


R
A
B
P
Q
C
D
S


R
A
B
P
Q
C
D
S

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

"Magic" square

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

"Magic" square

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

$$R \subseteq S \;\equiv\; \langle \forall\, b,a \;::\; b\,R\,a \Rightarrow b\,S\,a \rangle$$

$$b(R \cdot S)c \;\equiv\; \langle \exists\, a \;::\; b\,R\,a \;\wedge\; a\,S\,c \rangle$$

“Magic” square

Pointfree:

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

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

Pointwise:

$$\underset{\forall}{\exists} \quad \underset{c}{P} \overset{a}{\cdot} \underset{b}{R} \Rightarrow \underset{c}{S} \overset{d}{\cdot} \underset{b}{Q}$$

“Magic” square

Pointfree:

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

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

Pointwise:

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

Vertical composition

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

Horizontal composition

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

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

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

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

$$x \; R^\circ \; y \Leftrightarrow y \; R \; x$$

Functorial squares

Functor $\mathbb{F}$:

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

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

Squares with functions

Some relations f fit into the following squares:

$$
\begin{tikzcd}
A \arrow[d, "id"'] \arrow[dr, phantom, "\subseteq"] & A \arrow[l, "id"'] \arrow[d, "f"] \arrow[dr, phantom, "\subseteq"] & B \arrow[l, "f^\circ"'] \arrow[d, "id"] \\
A & B \arrow[l, "f^\circ"] & B \arrow[l, "id"]
\end{tikzcd}
\tag{3}
$$

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:

$$\begin{tikzcd} A \arrow[d, "id"'] \arrow[dr, phantom, "\subseteq"] & A \arrow[l, "id"'] \arrow[d, "f"] \arrow[dr, phantom, "\subseteq"] & B \arrow[l, "f^\circ"'] \arrow[d, "id"] \\ A & B \arrow[l, "f^\circ"] & B \arrow[l, "id"] \end{tikzcd} \qquad (3)$$

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:

$$\begin{tikzcd} A & A \arrow[l, "id"'] \arrow[d, "f"] & B \arrow[l, "f^\circ"'] \arrow[d, "id"] \\ A \arrow[from=u, "id"'] \arrow[ur, phantom, "\subseteq"] & B \arrow[l, "f^\circ"] \arrow[ur, phantom, "\subseteq"] & B \arrow[l, "id"] \end{tikzcd} \tag{3}$$

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:

$$
\begin{tikzcd}
A \arrow[d, "id"'] \arrow[dr, phantom, "\subseteq"] & A \arrow[l, "id"'] \arrow[d, "f"] \arrow[dr, phantom, "\subseteq"] & B \arrow[l, "f^\circ"'] \arrow[d, "id"] \\
A & B \arrow[l, "f^\circ"] & B \arrow[l, "id"]
\end{tikzcd}
\tag{3}
$$

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:

$$\begin{tikzcd} A & C \arrow[l, "R"'] \arrow[d, "Q"] \arrow[dl, phantom, "\subseteq"] \\ B & B \arrow[l, "id"] \end{tikzcd} \arrow[from=1-1, to=2-1, "f"'] \quad \Leftrightarrow \quad \begin{tikzcd} A \arrow[d, "id"'] & C \arrow[l, "R"'] \arrow[d, "Q"] \arrow[dl, phantom, "\subseteq"] \\ A & B \arrow[l, "f^\circ"] \end{tikzcd}$$

This is the **shunting** rule:

$$f \cdot R \subseteq Q \quad \Leftrightarrow \quad R \subseteq f^\circ \cdot Q \tag{4}$$

Taking converses:

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

Squares with functions

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

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

This is the **shunting** rule:

$$f \cdot R \subseteq Q \quad \Leftrightarrow \quad R \subseteq f^\circ \cdot Q \tag{4}$$

Taking converses:

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

"Nice" rules about functions

Functional **equality**:

$$f \subseteq g \quad \Leftrightarrow \quad f = g \quad \Leftrightarrow \quad g \subseteq f \tag{6}$$

∃-quantifiers go away:

$$b\,(f^{\circ} \cdot R \cdot g)\,a \quad \Leftrightarrow \quad (f\ b)\ R\ (g\ a) \tag{7}$$

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

☞ Very useful in practice ☺

"Nice" rules about functions

Functional **equality**:

$$f \subseteq g \quad \Leftrightarrow \quad f = g \quad \Leftrightarrow \quad g \subseteq f \tag{6}$$

$\exists$**-quantifiers go away**:

$$b\,(f^{\circ} \cdot R \cdot g)\; a \quad \Leftrightarrow \quad (f\ b)\ R\ (g\ a) \tag{7}$$

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

☞ Very useful in practice ☺

Squares with functions

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

$$
\begin{tikzcd}
A \arrow[d, "f"'] & B \arrow[l, "R"'] \arrow[d, "g"] \arrow[dl, phantom, "\subseteq"] \\
C & D \arrow[l, "S"]
\end{tikzcd}
\qquad f \cdot R \subseteq S \cdot g \tag{8}
$$

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

$$f \; S^R \; g \quad \Leftrightarrow \quad f \cdot R \subseteq S \cdot g \tag{9}$$

In words:

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

Squares with functions

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

$$\begin{tikzcd} A & B \arrow[l, "R"'] \arrow[d, "g"] \\ C \arrow[u, leftarrow, "f"'] \arrow[ur, phantom, "\subseteq"] & D \arrow[l, "S"] \end{tikzcd} \qquad f \cdot R \subseteq S \cdot g \tag{8}$$

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

$$f \; S^R \; g \quad \Leftrightarrow \quad f \cdot R \subseteq S \cdot g \tag{9}$$

In words:

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

"Higher-order" squares

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

$$\begin{tikzcd} A & B \arrow[l, "R"'] \arrow[d, "g"] \\ C \arrow[u, leftarrow, "f"] \arrow[ur, phantom, "\subseteq"] & D \arrow[l, "S"] \end{tikzcd}$$

that is to say,

$$\frac{A \xleftarrow{R} B \qquad C \xleftarrow{S} D}{C^A \xleftarrow{S^R} D^B}$$

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

$$(S^R)^\circ = (S^\circ)^{(R^\circ)} \tag{10}$$

$$id^{id} = id \tag{11}$$

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

“Higher-order” squares

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

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

that is to say,

$$\frac{A \xleftarrow{R} B \qquad C \xleftarrow{S} D}{C^A \xleftarrow{S^R} D^B}$$

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

$$(S^R)^\circ = (S^\circ)^{(R^\circ)} \tag{10}$$

$$id^{id} = id \tag{11}$$

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:

$$f\ (k^h)\ g \quad \Leftrightarrow \quad f \cdot h = k \cdot g \tag{12}$$

In case of h° instead of h,

$$f\ (k^{h^\circ})\ g \Leftrightarrow f \cdot h^\circ \subseteq k \cdot g \tag{13}$$

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

$$(k^{h^\circ})\ g = k \cdot g \cdot h \tag{14}$$

"Higher-order" squares

Functions-only Reynolds squares:

$$f\ (k^h)\ g \quad \Leftrightarrow \quad f \cdot h = k \cdot g \tag{12}$$

In case of h° instead of h,

$$f\ (k^{h^\circ})\ g \Leftrightarrow f \cdot h^\circ \subseteq k \cdot g \tag{13}$$

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

$$(k^{h^\circ})\ g = k \cdot g \cdot h \tag{14}$$

"Higher-order" squares

Then:

$$(id \rightarrow k)\ g \quad = \quad k \cdot g \tag{15}$$

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

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

In fully pointfree notation, (15,16) become

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

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

Then, by (10):

$$id^h = (\cdot h)^\circ \tag{17}$$

and so on and so forth.

☞ Rich construction! ☺

Higher-order Reynolds squares

Relational exponentials S^R can involve other exponentials, for instance $(S^Q)^R$ i.e. $R \to S^Q$:

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

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

Higher-order Reynolds squares

$$f\ (R \to S^Q)\ g \tag{18}$$

$$\Leftrightarrow \quad \{\ \text{Reynolds square (8)}\ \}$$

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

$$\Leftrightarrow \quad \{\ \text{shunting (4) followed by “nice rule” (7)}\ \}$$

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

$$\Leftrightarrow \quad \{\ \text{(8) again}\ \}$$

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

$$\Leftrightarrow \quad \{\ \text{(4) followed by (7) again}\ \}$$

$$a\ R\ b \Rightarrow c\ Q\ d \Rightarrow (f\ \ a\ \ c)\ S\ (g\ \ b\ \ d) \tag{19}$$

Relational types

Let $f = g$ in Reynolds square (8):

$$\begin{tikzcd} A \arrow[d, "f"'] \arrow[dr, phantom, "\subseteq"] & A \arrow[l, "R"'] \arrow[d, "f"] \\ C & C \arrow[l, "S"] \end{tikzcd} \qquad f \cdot R \subseteq S \cdot f \tag{20}$$

We often abbreviate $f\ S^R\ f$ 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)*.

Relational types by example

$f : (\leqslant) \to (\preceq)$ f is **monotonic**

$$\begin{tikzcd} A \arrow[d, "f"'] & A \arrow[l, "(\leqslant)"'] \arrow[d, "f"] \arrow[dl, phantom, "\subseteq"] \\ B & B \arrow[l, "(\preceq)"] \end{tikzcd}$$

$f : (t_0+) \to id$ f is **periodic**

$$\begin{tikzcd} \mathbb{R} \arrow[d, "f"'] & \mathbb{R} \arrow[l, "(t_0+)"'] \arrow[d, "f"] \arrow[dl, phantom, "\subseteq"] \\ B & B \arrow[l, "id"] \end{tikzcd}$$

$f : \top \to id$ f is **constant**

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

Category

Objects — binary relations R, S, ...

Morphisms — $R \xrightarrow{f} S$ as above (20)

☞ 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**: $(y, x)\ (R \times S)\ (b, a) \Leftrightarrow y\ R\ b \wedge x\ S\ a$.

Category

Objects — binary relations R, S, ...

Morphisms — $R \xrightarrow{f} S$ as above (20)

☞ 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**: $(y,x)\,(R \times S)\,(b,a) \Leftrightarrow y\ R\ b \wedge x\ S\ a$.

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**

$$f : \mathbb{F}\, R \to \mathbb{G}\, R \tag{21}$$

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

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

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

Free theorem squares

Example: Haskell constant function $\mathtt{const} : a \to b \to a$.

By (21), $\mathtt{const}$ has relational type $R \to R^S$, that is:

$$\begin{tikzcd} A \arrow[d, "\mathtt{const}"'] & C \arrow[l, "R"'] \arrow[d, "\mathtt{const}"] \arrow[dl, phantom, "\subseteq"] \\ A^B & C^D \arrow[l, "R^S"] \end{tikzcd} \qquad \mathtt{const} \cdot R \subseteq R^S \cdot \mathtt{const} \tag{22}$$

Free theorem squares

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

$$\Updownarrow (9)$$

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

Free theorem squares

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

$$\Big\updownarrow \; (9)$$

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

Free theorem squares

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

$$\updownarrow \; (9)$$

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

Free theorem squares

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

$$
\updownarrow \; {\color{red}(9)}
$$

$$
\begin{tikzcd}
B \arrow[d, "id"'] \arrow[dr, phantom, "\subseteq"] & D \arrow[l, "S"'] \arrow[d, "\mathtt{const}\ a"] \\
B & A \arrow[l, "(\mathtt{const}\ a)^\circ"]
\end{tikzcd}
\quad \xleftrightarrow{\color{red}\text{shunting}} \quad
\begin{tikzcd}
B \arrow[d, "\mathtt{const}\ a"'] \arrow[dr, phantom, "\subseteq"] & D \arrow[l, "S"'] \arrow[d, "\mathtt{const}\ a"] \\
A & A \arrow[l, "id"]
\end{tikzcd}
$$

Free theorem squares

$$\begin{tikzcd} B & D \arrow[l, "S"'] \arrow[d, "\mathtt{const}\ a"] \\ B \arrow[from=1-1, "id"'] \arrow[ur, phantom, "\subseteq"] & A \arrow[l, "(\mathtt{const}\ a)^\circ"] \end{tikzcd}$$

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

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

$$(\mathtt{const}\ a)^\circ \cdot (\mathtt{const}\ a) = \top \qquad (23)$$

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

On Injectivity

NB: **injective** functions are those that fit the square:

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

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

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

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

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

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

Free theorem squares

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

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

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

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

Free theorem squares

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

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

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

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

Free theorem squares

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

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

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

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

Free theorem squares

Example:

$$\texttt{flip} :: (a \to b \to c) \to b \to a \to c \tag{24}$$

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 (where $\widetilde{f}$ abbreviates $\texttt{flip}\ f$):

$$\begin{tikzcd} A \arrow[d, "g"'] & X \arrow[l, "R"'] \arrow[d, "f"] \arrow[dl, phantom, "\subseteq"] \\ C^B & Z^Y \arrow[l, "Q^S"] \end{tikzcd} \quad \Rightarrow \quad \begin{tikzcd} B \arrow[d, "\widetilde{g}"'] & Y \arrow[l, "S"'] \arrow[d, "\widetilde{f}"] \arrow[dl, phantom, "\subseteq"] \\ C^A & Z^X \arrow[l, "Q^R"] \end{tikzcd} \tag{25}$$

Free theorem squares

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

$$
\begin{array}{ll}
& f \; (r \rightarrow id) \; g \Rightarrow \widetilde{f} \; (id \rightarrow id^{r}) \; \widetilde{g} \\
\Leftrightarrow & \qquad \{ \;\; (12) \; ; (15) \;\; \} \\
& f \cdot r = g \;\; \Rightarrow \;\; \widetilde{f} \;\; \subseteq \;\; id^{r} \cdot \widetilde{g} \\
\Leftrightarrow & \qquad \{ \;\; id^{r} = (\cdot r)^{\circ} \; (17) \; ; \text{substitution of } g; \text{ shunting } (4) \;\; \} \\
& (\cdot r) \cdot \widetilde{f} = \widetilde{f \cdot r}
\end{array}
$$

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:

$$\begin{aligned} \mathbf{foldl} &:: \mathit{Foldable}\ t \Rightarrow (b \to a \to b) \to b \to t\ a \to b \\ \mathbf{foldr} &:: \mathit{Foldable}\ t \Rightarrow (a \to b \to b) \to b \to t\ a \to b \end{aligned} \tag{26}$$

(Hackage's Data.Foldable)

Free theorem squares

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

... *"how much is in a* **type"** *?*

Quite a lot.

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

$$\begin{aligned} \textbf{foldl} &:: Foldable\ t \Rightarrow (b \to a \to b) \to b \to t\ a \to b \\ \textbf{foldr} &:: Foldable\ t \Rightarrow (a \to b \to b) \to b \to t\ a \to b \end{aligned} \tag{26}$$

(Hackage's Data.Foldable)

foldl and **foldr** squares

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

$$\textbf{foldl} : (S \to S^R) \to (S \to S^{\mathbb{T}\,R}) \tag{27}$$

$$\textbf{foldr} : (R \to S^S) \to (S \to S^{\mathbb{T}\,R}) \tag{28}$$

As seen above:

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

foldl squares

The squares of

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

are:

$$\begin{tikzcd} B \arrow[d, "g"'] & Y \arrow[l, "s"'] \arrow[d, "f"] \arrow[dl, phantom, "\subseteq"] \\ B^A & Y^X \arrow[l, "s^R"] \end{tikzcd} \Rightarrow \begin{tikzcd} B \arrow[d, "\mathbf{foldl}\ g"'] & Y \arrow[l, "s"'] \arrow[d, "\mathbf{foldl}\ f"] \arrow[dl, phantom, "\subseteq"] \\ B^{\mathbb{T}\,A} & Y^{\mathbb{T}\,X} \arrow[l, "s^{\mathbb{T}\,R}"] \end{tikzcd} \tag{29}$$

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 (15).

So the squares become equalities:

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

Pointwise:

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

---

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

---

foldr

Repeating the above exercise for **foldr** (28):

$$\begin{tikzcd} A \arrow[d, "g"'] \arrow[dr, phantom, "\subseteq"] & X \arrow[l, "R"'] \arrow[d, "f"] \\ B^B & Y^Y \arrow[l, "S^S"] \end{tikzcd} \Rightarrow \begin{tikzcd} B \arrow[d, "\mathbf{foldr}\ g"'] \arrow[dr, phantom, "\subseteq"] & Y \arrow[l, "S"'] \arrow[d, "\mathbf{foldr}\ f"] \\ B^{\mathbb{T}\ A} & Y^{\mathbb{T}\ X} \arrow[l, "S^{\mathbb{T}\ R}"] \end{tikzcd} \tag{30}$$

Same right square as in (29), 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:

$$
\begin{aligned}
& g\ (id \to h^h)\ f \\
\Leftrightarrow \quad & \{\ (47)\ \} \\
& (g\ x)\ h^h\ (f\ x) \\
\Leftrightarrow \quad & \{\ (12)\ \} \\
& (g\ x) \cdot h = h \cdot (f\ x)
\end{aligned}
$$

foldr squares

Altogether, one has

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

that is:

$$(g\ x) \cdot h = h \cdot (f\ x) \quad \Rightarrow \quad \mathbf{foldr}\ g \cdot h = (h\cdot) \cdot \mathbf{foldr}\ f \tag{31}$$

i.e. the fully pointwise:

$$g\ x\ (h\ y) = h\ (f\ x\ y) \quad \Rightarrow \quad h\ (\mathbf{foldr}\ f\ \ e\ xs) = \mathbf{foldr}\ g\ \ (h\ e)\ xs \tag{32}$$

---

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

Corollary of **foldr**-fusion

In (31), let f and g be the same function, say s, and let $h := s\ a$

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

Then (31) becomes:

$$(s\ x) \cdot (s\ a) = (s\ a) \cdot (s\ x) \quad \Rightarrow \quad \mathbf{foldr}\ s \cdot (s\ a) = (s\ a\cdot) \cdot \mathbf{foldr}\ s \tag{33}$$

Permutativity squares

Square

$$\begin{tikzcd} B \arrow[d, "s\ x"'] & B \arrow[l, "s\ a"'] \arrow[d, "s\ x"] \\ B & B \arrow[l, "s\ a"] \end{tikzcd} \tag{34}$$

captures the (left) **permutativity** property of (Danvy, 2023):

$$(s\ x) \cdot (s\ a) = (s\ a) \cdot (s\ x) \tag{35}$$

— i.e. the fully pointwise $s\ x\ (s\ a\ y) = s\ a\ (s\ x\ y)$

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

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

Looking at

$$\begin{aligned} \mathbf{foldl} &:: \mathit{Foldable}\ t \Rightarrow (b \to a \to b) \to b \to t\ a \to b \\ \mathbf{foldr} &:: \mathit{Foldable}\ t \Rightarrow (a \to b \to b) \to b \to t\ a \to b \end{aligned}$$

the *type-wise distance* between **foldr** and **foldl** is the flip (24) of the first parameter.

So the "best fit" one can aim at is

$$\mathbf{foldl}\ f \ \stackrel{?}{=}\ \mathbf{foldr}\ \tilde{f} \tag{36}$$

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 (32):

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

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

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

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

$$\mathbf{foldr}\ \alpha\ \gamma = id \tag{37}$$

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 (32):

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

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

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

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

$$\mathbf{foldr}\ \alpha\ \gamma = id \tag{37}$$

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:

$$\begin{cases} h\ \gamma = z \\ h\ (\alpha\ x\ y) = g\ x\ (h\ y) \end{cases} \quad \Rightarrow \quad h = \mathbf{foldr}\ g\ z \tag{38}$$

So, **foldr** $g\ z$ is the unique solution for h of 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**:

$$\begin{cases} \mathbf{foldr}\ g\ z\ \gamma = z \\ \mathbf{foldr}\ g\ z\ (\alpha\ x\ y) = g\ x\ (\mathbf{foldr}\ g\ z\ y) \end{cases} \tag{39}$$

Towards **foldr**-universal

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

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

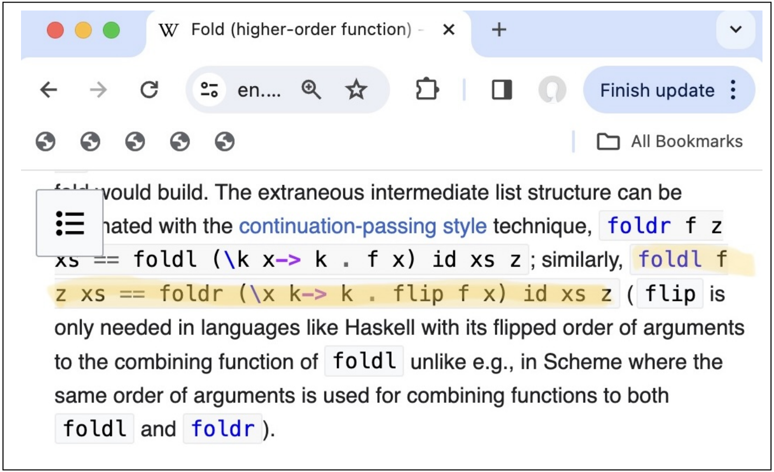
Altogether, (38) and (40) make up a universal property:

$$h = \textbf{foldr}\ g\ z \quad \Leftrightarrow \quad \begin{cases} h\ \gamma = z \\ h\ (\alpha\ x\ xs) = g\ x\ (h\ xs) \end{cases} \tag{41}$$

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

What about **foldl**?

Wikipedia



fold 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,

$$\widetilde{\textbf{foldl}\ f} = \textbf{foldr}\ (\lambda x\ k \to k \cdot \tilde{f}\ x)\ id \tag{42}$$

or

$$\begin{aligned} \widetilde{\textbf{foldl}\ f} &= \textbf{foldr}\ (\theta\ f)\ id \\ &\quad \textbf{where}\ (\theta\ f)\ x\ k = k \cdot (\tilde{f}\ x) \end{aligned} \tag{43}$$

cf. the (functional) square

$$\begin{tikzcd} B^B \arrow[d, "(\cdot k)"'] & A \arrow[l, "\theta\ f"'] \arrow[d, "\tilde{f}"] \\ B^B & B^B \arrow[l, "(k\cdot)"] \end{tikzcd} \tag{44}$$

Universal-**foldl**

An advantage of defining **foldl** *"as a **foldr**"* (43) is that the universal property of the latter induces the universal property of the former:

$$
\begin{array}{ll}
 & k = \mathbf{foldl}\ f \\
\Leftrightarrow & \quad \{\ \mathbf{foldl}\ f = \overbrace{\mathbf{foldr}\ (\theta\ f)\ id}\ (43)\ ;\ \text{flipping}\ \} \\
 & \widetilde{k} = \mathbf{foldr}\ (\theta\ f)\ id \\
\Leftrightarrow & \quad \{\ \text{universal-}\mathbf{foldr}\ (41)\ \text{etc}\ \} \\
 & \left\{ \begin{array}{l} \widetilde{k}\ \gamma = id \\ \widetilde{k}\ (\alpha\ x\ xs) = (\theta\ f)\ x\ (\widetilde{k}\ xs) \end{array} \right.
\end{array}
$$

Universal-**foldl**

$\Leftrightarrow$ { introduce z and flip }

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

$\Leftrightarrow$ { square (44) — $(\theta\ f)\ x\ g = g \cdot (\widetilde{f}\ x)$ }

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

$\Leftrightarrow$ { 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**:

$$k = \mathbf{foldl}\ f \quad \Leftrightarrow \quad \begin{cases} k\ z\ \gamma = z \\ k\ z\ (\alpha\ x\ xs) = k\ (f\ z\ x)\ xs \end{cases} \tag{45}$$

Good — we already know something about **foldl** and **foldr**

But question (36) remains:

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

Universal-**foldl**

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

$$k = \mathbf{foldl}\ f \quad \Leftrightarrow \quad \begin{cases} k\ z\ \gamma = z \\ k\ z\ (\alpha\ x\ xs) = k\ (f\ z\ x)\ xs \end{cases} \tag{45}$$

Good — we already know something about **foldl** and **foldr** ☺

But question (36) remains:

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

Universal-**foldl**

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

$$k = \textbf{foldl}\ f \quad \Leftrightarrow \quad \begin{cases} k\ z\ \gamma = z \\ k\ z\ (\alpha\ x\ xs) = k\ (f\ z\ x)\ xs \end{cases} \tag{45}$$

Good — we already know something about **foldl** and **foldr**

But question (36) remains:

Under what conditions does $\textbf{foldl}\ f = \textbf{foldr}\ \widetilde{f}$ *hold?*

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

A 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**

A popular assumption is that **foldl** f e and **foldr** $\widetilde{f}$ 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 (45) to find an answer:

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

$$\Leftrightarrow \qquad \{\ \text{universal property (45)}\ \}$$

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

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

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

Back to the permutativity squares

Recall (33)

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

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

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

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

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

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

$\Leftrightarrow$ { (33) 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** (39) }

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 (37), provided that $\widetilde{f}$ is **permutative**.

Back to e.g.

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

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

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

Galois connection squares

The specification of $x \div y$ is a **Galois connection**:

$$\begin{tikzcd} A \arrow[d, "(\times y)^\circ"'] \arrow[dr, phantom, "="] & A \arrow[l, "(\leqslant)"'] \arrow[d, "(\div y)"] \\ B & B \arrow[l, "(\leqslant)"] \end{tikzcd} \qquad a \times y \leqslant x \quad \Leftrightarrow \quad a \leqslant x \div y \tag{46}$$

We can use (46) 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

$$
\begin{array}{ll}
& y \leqslant (x \div b) \div a \\
\Leftrightarrow & \quad \{ \text{ Galois connection (46) twice } \} \\
& (y \times a) \times b \leqslant x \\
\Leftrightarrow & \quad \{ \text{ } (\times) \text{ is } \textbf{associative} \text{ and } \textbf{commutative} \text{ } \} \\
& (y \times b) \times a \leqslant x \\
\Leftrightarrow & \quad \{ \text{ Galois connection (46) twice in the opposite direction } \} \\
& y \leqslant (x \div a) \div b \\
:: & \quad \{ \text{ by indirect equality (Dijkstra, 2001) } \} \\
& (x \div b)\ a = (x \div a) \div b
\end{array}
$$

Comments

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

Danvy's reasoning is, however, quite different: permutativity is **postulated** as side condition and then **proved** in Coq by **list induction**.

☞ Above, **permutativity** arose (generically) by **free-theorem** calculation.

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

Summary

- **Simplicity** (eventually) wins

  ☞ "Magic" squares ☺

- **Widening** scope (usually) helps

  ☞ (Binary) relations!

Summary

- **Simplicity** (eventually) wins

  "Magic" squares

- **Widening** scope (usually) helps

  (Binary) relations!

FPCA 1989

"From the **type** *of a polymorphic function we can derive a* **theorem** *that it satisfies. (...) How useful are the theorems so generated?*

Only **time** *and* **experience** *will tell (...)"*

Indeed — many years later, experience is still telling us how useful such a fantastic result is!

Acknowledgements

This work is funded by national funds through
FCT – Fundação para a Ciência e a Tecnologia, I.P.,
under the support UID/50014/2023
(https://doi.org/10.54499/UID/50014/2023)

Annex

Permutativity matters

Insertion

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

on a linearly **ordered** list is a **permutative** operation.

Thus **insertion sort**

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

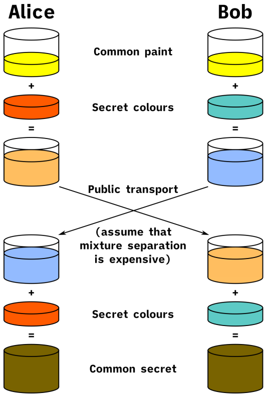
computes the same as

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

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

Permutativity matters

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



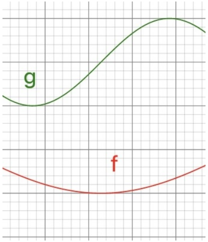
$$(+red) \cdot (+cyan) = (+cyan) \cdot (+red)$$

[1] Source: Wikipedia

Pointwise ordering squares

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

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



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

$$f\ (\leqslant)^{id}\ g \quad \Leftrightarrow \quad \langle \forall\ a\ ::\ f\ a \leqslant g\ a \rangle \tag{47}$$

In words:

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

Logical relation 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 (6) 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:

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

Hoare triple squares

Let $\Phi_p : A \to A$ be such that $b\ \Phi_p\ a \Leftrightarrow b = a \wedge p\ a$ in:

$$\begin{tikzcd} A \arrow[d, "f"'] \arrow[dr, phantom, "\subseteq"] & A \arrow[l, "\Phi_p"'] \arrow[d, "f"] \\ B & B \arrow[l, "\Phi_q"] \end{tikzcd}$$

This square captures the **Hoare triple**:

$$\langle \forall\ a\ ::\ p\ a \Rightarrow q\ (f\ a) \rangle$$

Big-$\mathcal{O}$ squares

Define

$$y \leqslant_C x \Leftrightarrow y \leqslant C\ x$$

for some scalar C, in:

$$\begin{tikzcd} \mathbb{N}_0 \arrow[d, "f"'] \arrow[dr, phantom, "\subseteq"] & \mathbb{N}_0 \arrow[l, "\Phi_{(\geqslant n_0)}"'] \arrow[d, "g"] \\ \mathbb{R} & \mathbb{R} \arrow[l, "\leqslant_C"] \end{tikzcd}$$

Meaning:

$$\langle \forall\ n\ ::\ n \geqslant n_0 \Rightarrow f\ n \leqslant C\ g\ n \rangle$$

Checking that (43) defines **foldl**

We unfold (43) via universal property (41):

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

$$\Leftrightarrow \qquad \{\ \text{universal-}\textbf{foldr}\ (41)\ \}$$

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

$$\Leftrightarrow \qquad \{\ \text{definition of } \theta\ (44)\ \}$$

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

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

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

On relational exponentials S^R

By vertical composition (1) 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 $id^{id} = id$ (11).

By horizontal composition (2) we get

$$S^R \cdot {S'}^{R'} \subseteq (S \cdot S')^{(R \cdot R')} \tag{48}$$

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 (48) 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 $\textbf{foldr}\ \alpha\ \gamma$:

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

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

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. ISBN: 9780444703682.

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.

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.C. Reynolds. Types, abstraction and parametric polymorphism. *Information Processing 83*, pages 513–523, 1983.

J. Voigtländer. Free theorems simply, via dinaturality, 2019. arXiv cs.PL 1908.07776.

P.L. Wadler. Theorems for free! In *4th Int. Symp. on Functional Programming Languages and Computer Architecture*, pages 347–359, London, Sep. 1989. ACM.