"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

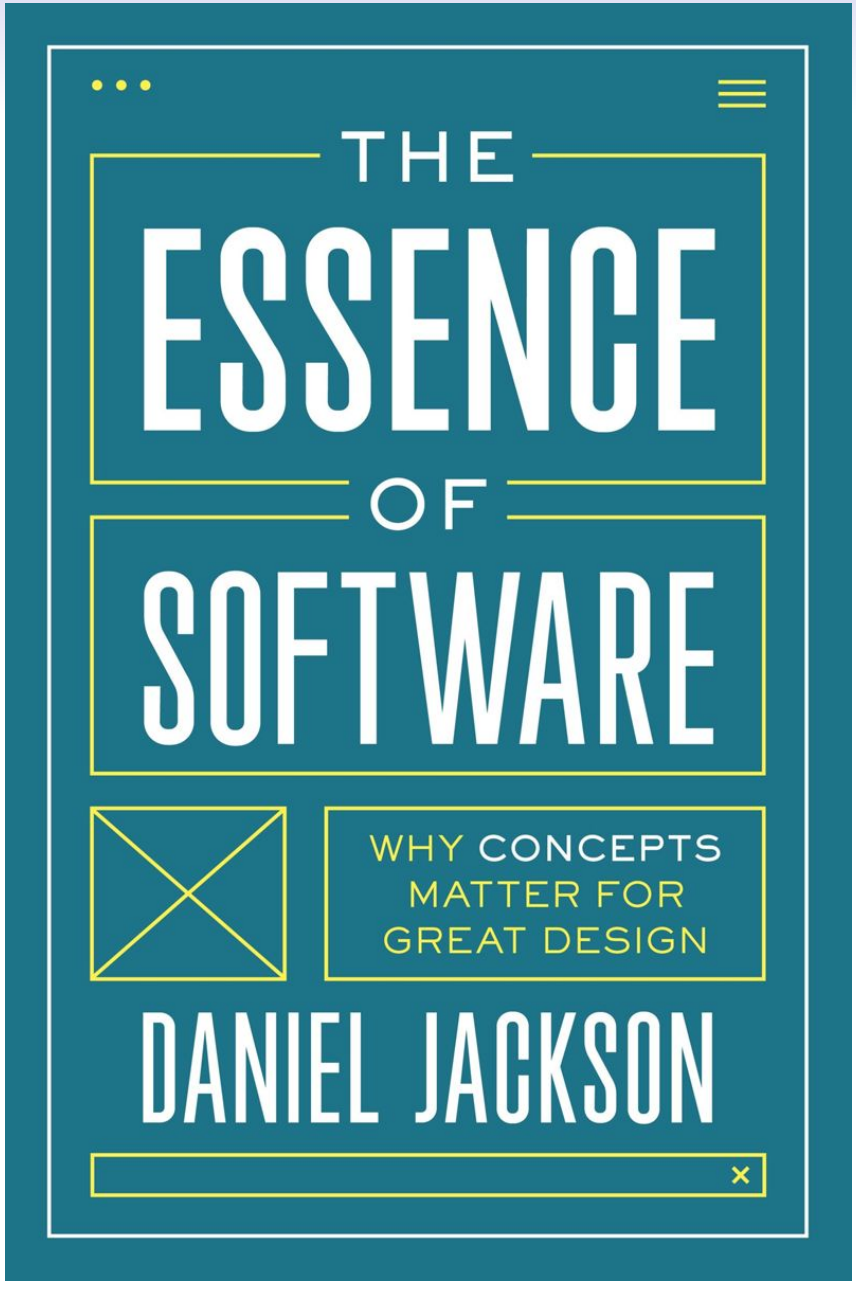
University of Minho & INESC TEC

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**.

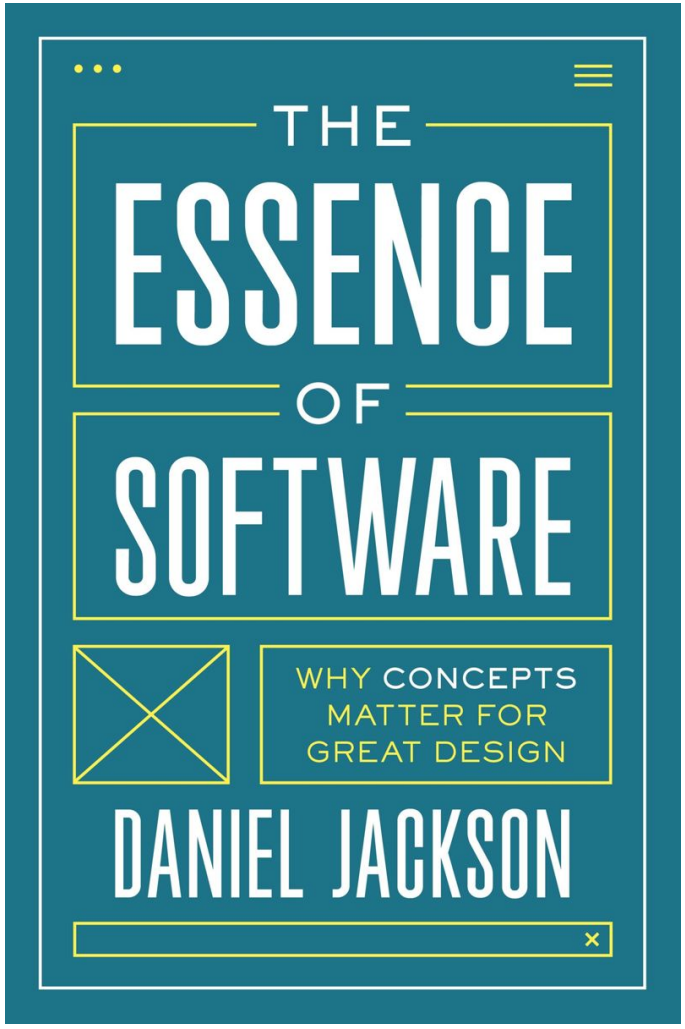

THE
ESSENCE
OF
SOFTWARE
WHY CONCEPTS MATTER FOR GREAT DESIGN
DANIEL JACKSON

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.

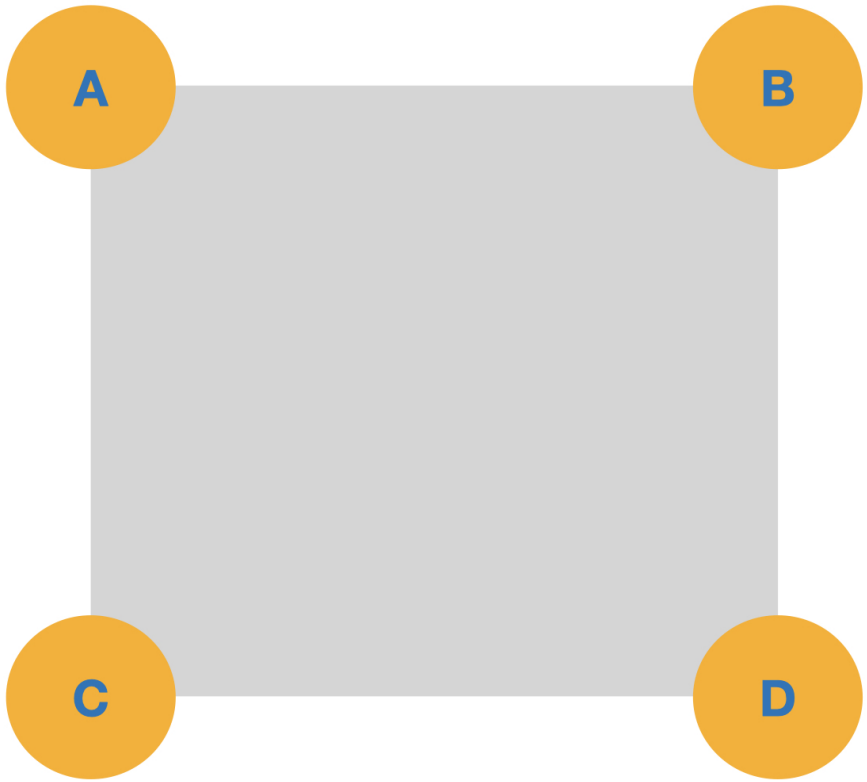
This calculus offers another, **often more amenable** framework for concepts and methods discussed in part one."

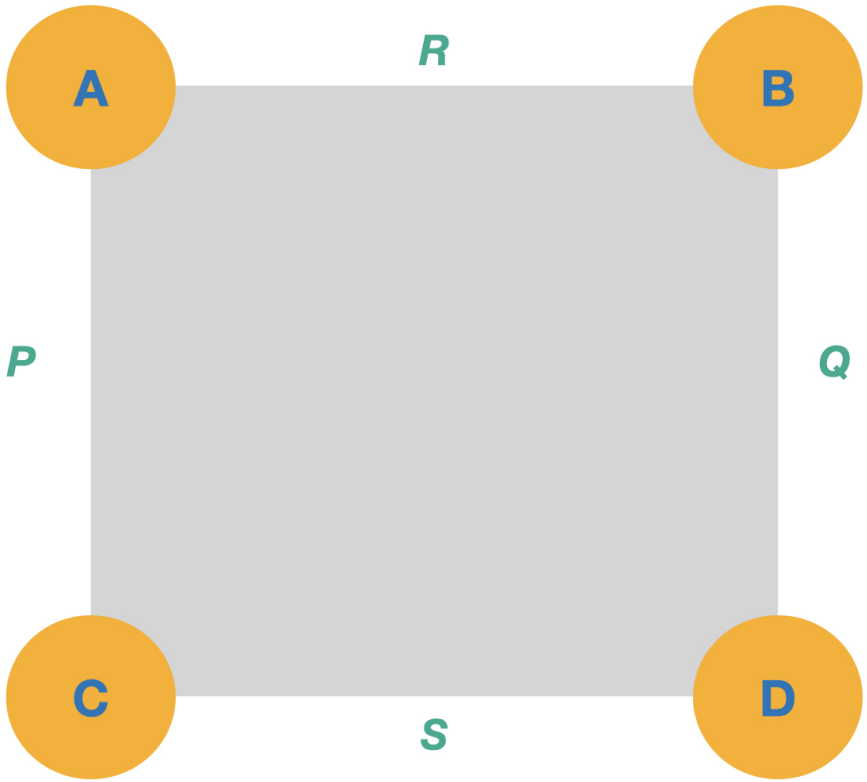
Freyd & Ščedrov, 1990

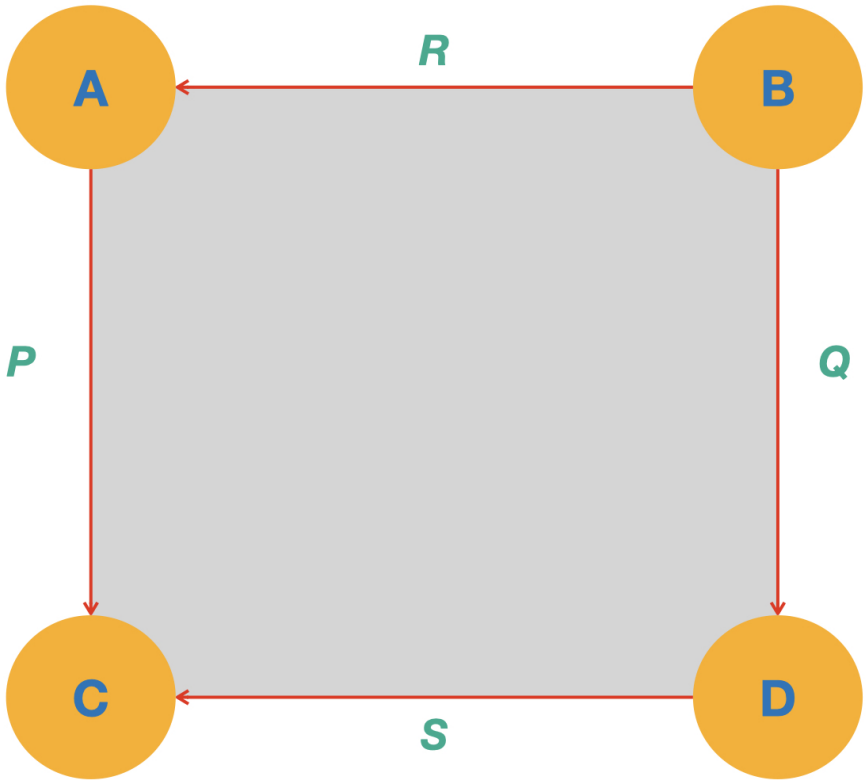
"(...) A special feature of our approach is a general **calculus of relations** presented in part two.

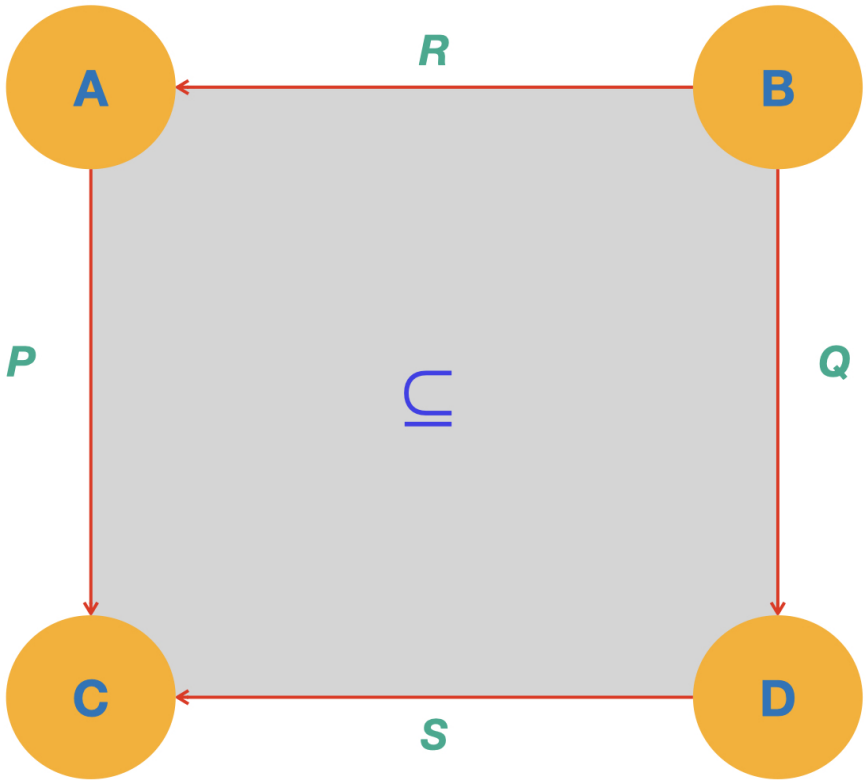
This calculus offers another, **often more amenable** framework for concepts and methods discussed in part one."

Squares

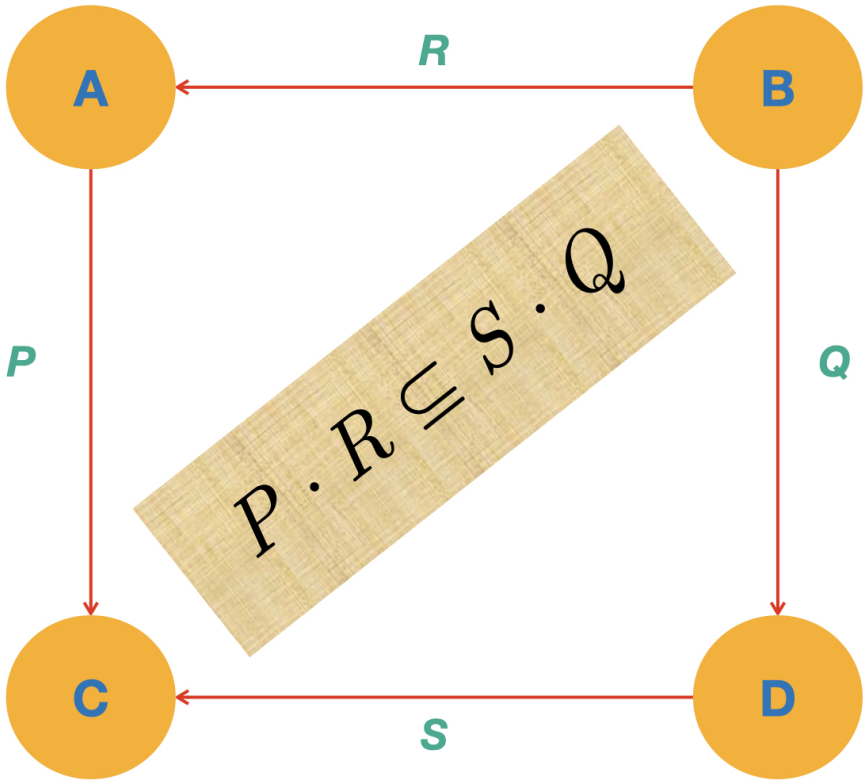

A
B
C
D


A
R
B
P
Q
C
S
D


A
B
C
D
R
S
P
Q

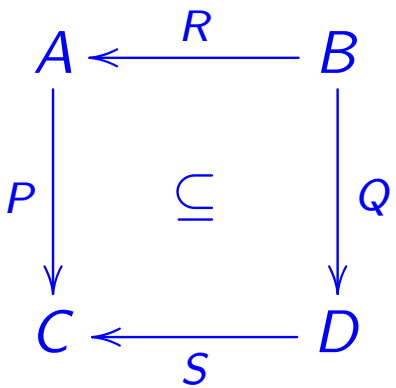

A
R
B
P
⊆
Q
C
S
D

“Magic” square



"Magic" square

Four binary relations:



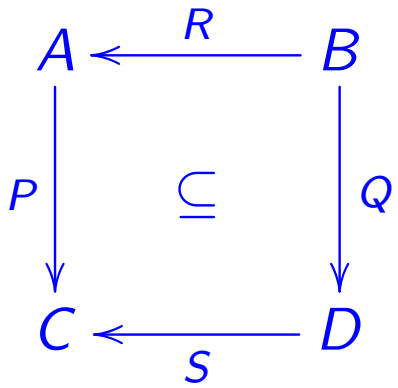
$$P \cdot R \subseteq S \cdot Q \tag{1}$$

Terminology:

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

“Magic” square

Pointfree:



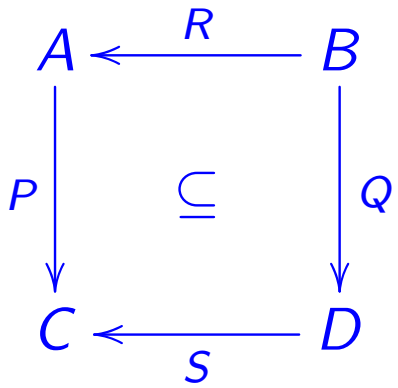
$$P \cdot R \subseteq S \cdot Q$$

Pointwise:

$$\forall\, c,\ \exists\, a, b \;.\; P\ a\ b \wedge R\ b\ c \Rightarrow \forall\, d, b \;.\; S\ d\ b \wedge Q\ b\ c$$

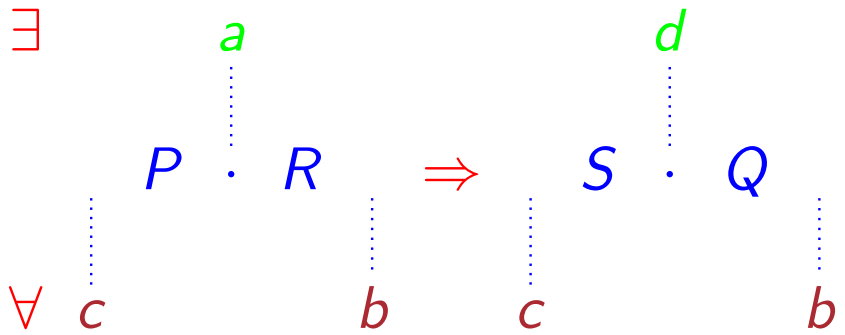
"Magic" square

Pointfree:



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

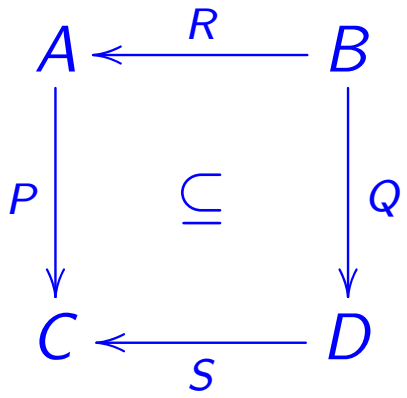
Pointwise:



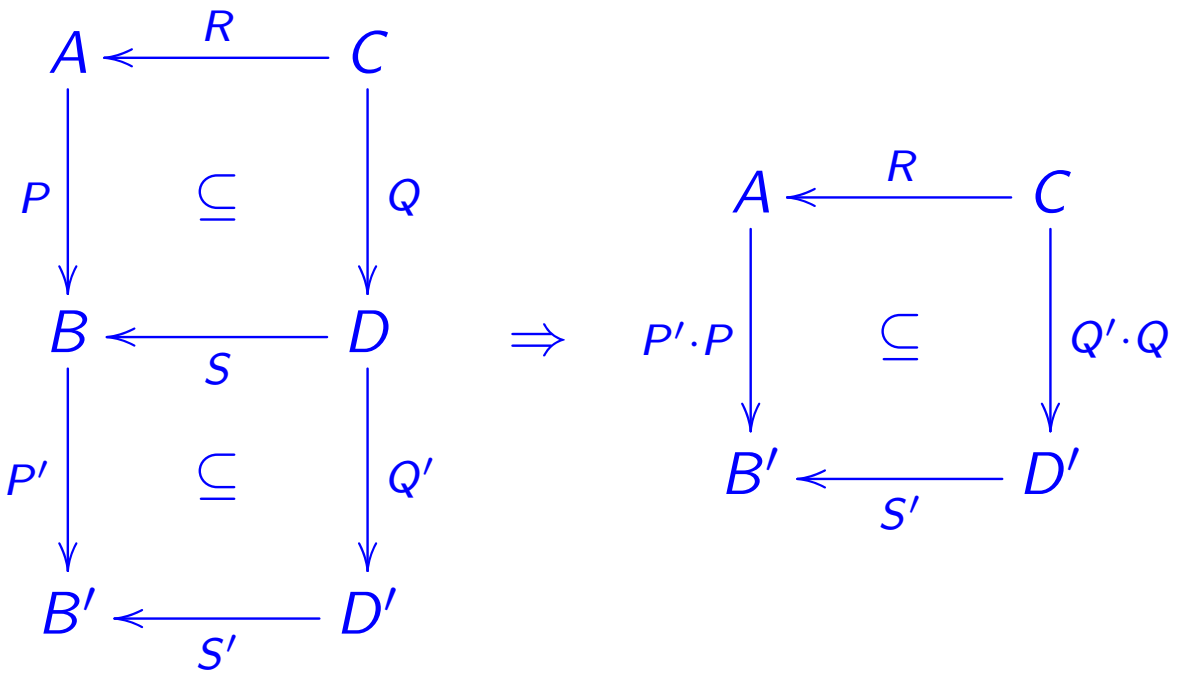
"Magic" square

Perhaps not what you were expecting...

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

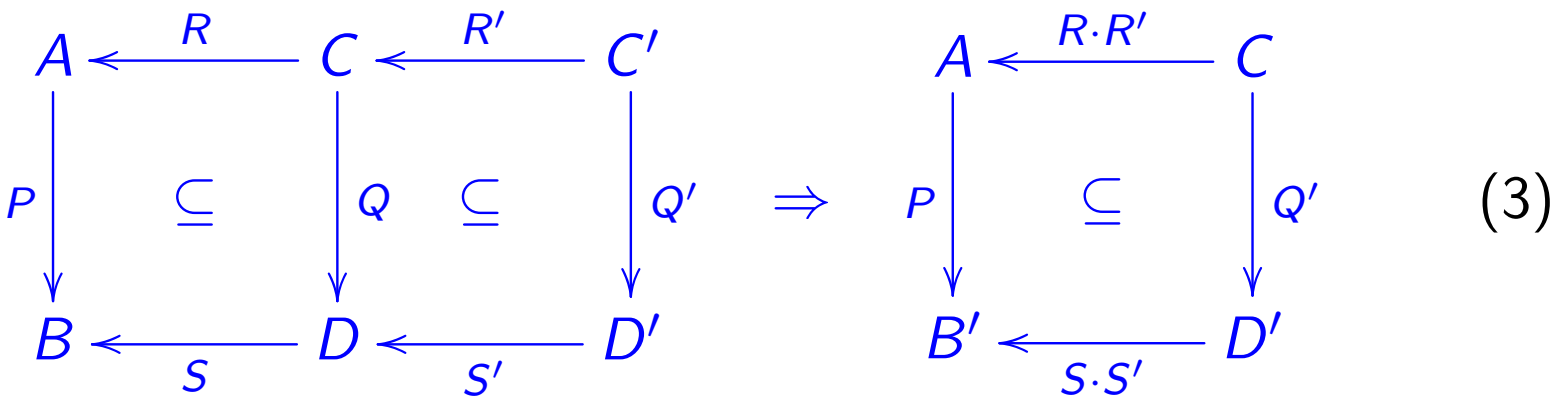


Vertical composition



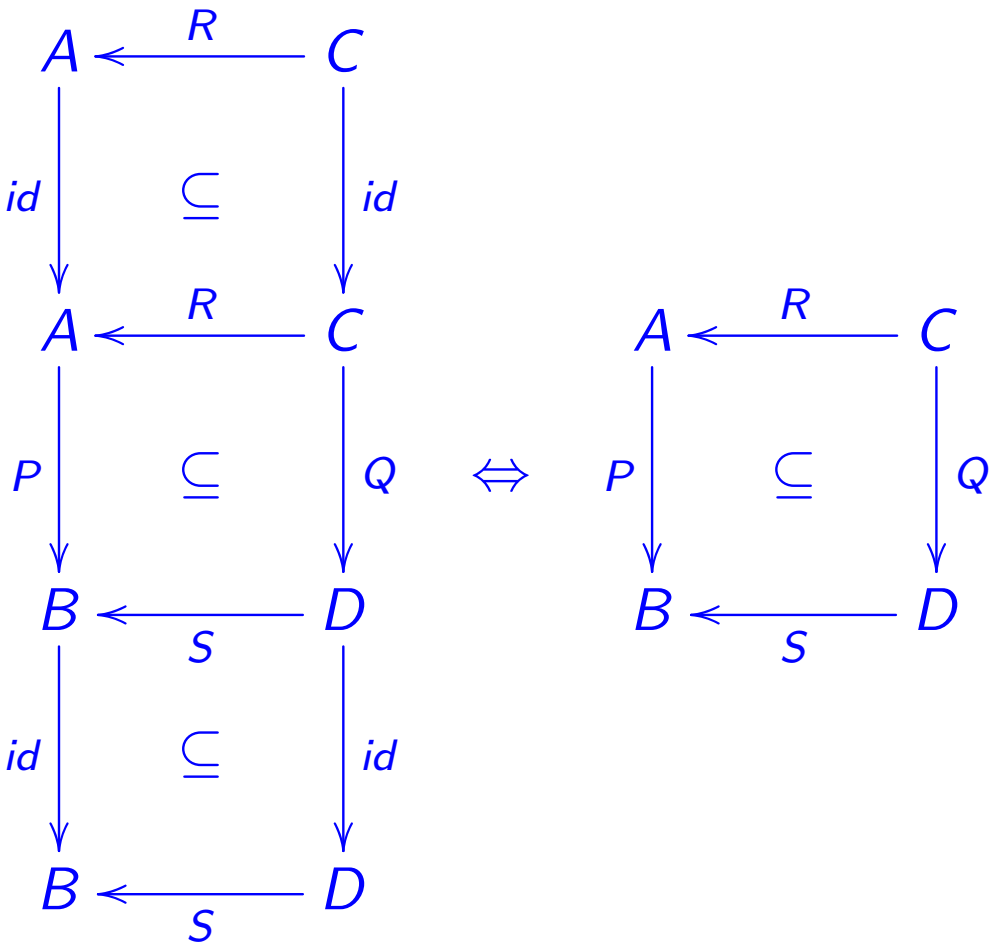
(2)

Horizontal composition



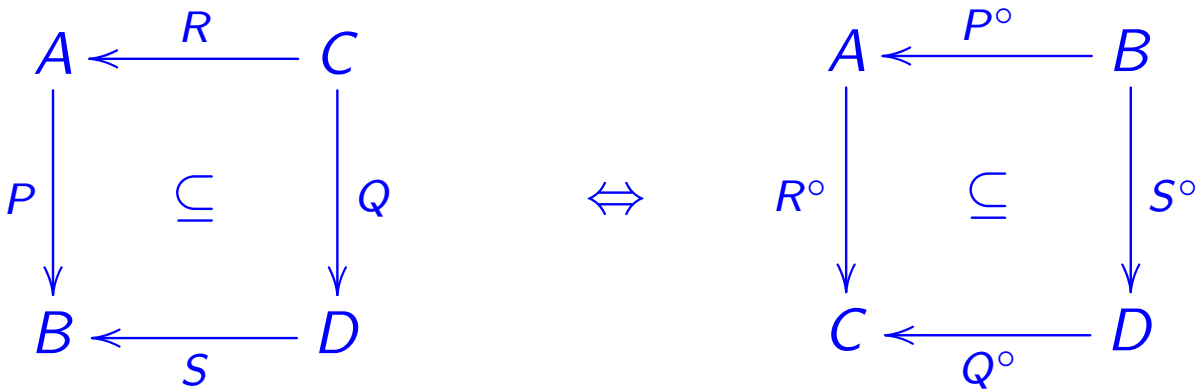
(3)

Identity



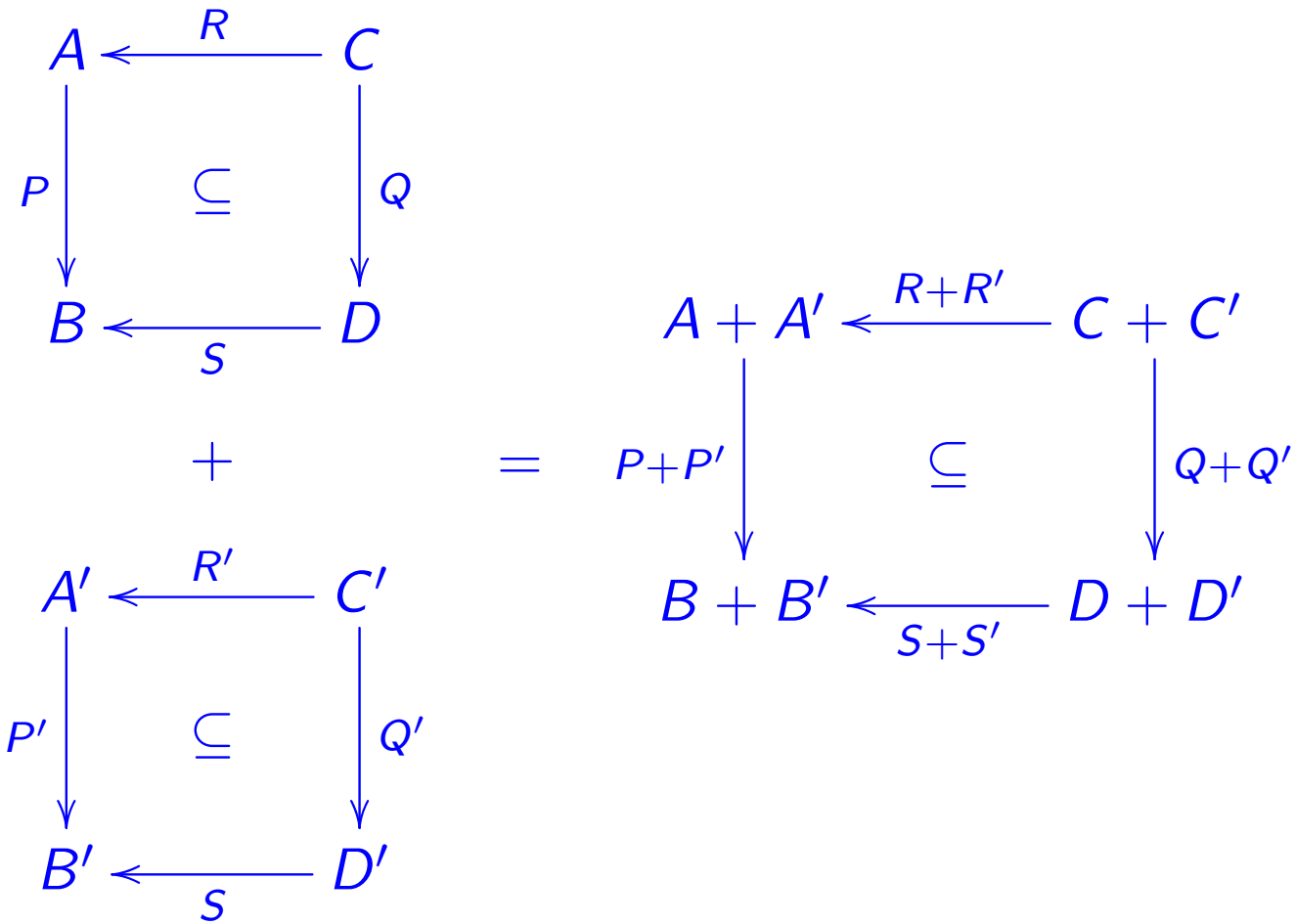
(Similarly for horizontal.)

Converse



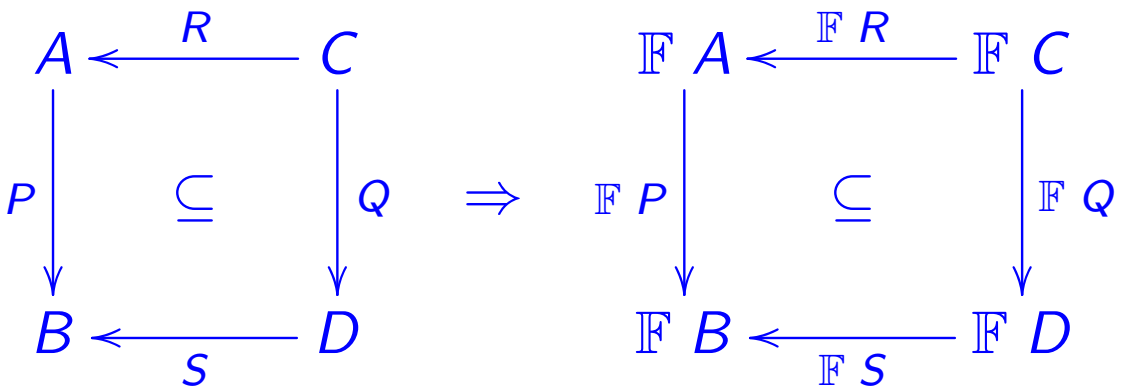
The converse of a square is its "*passive voice*"

Square (direct) sums



Functorial squares

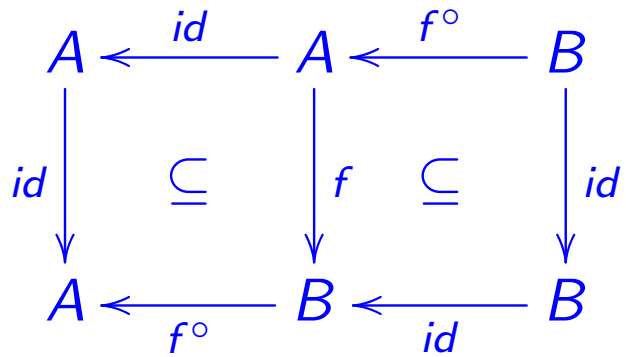
Functor $\mathbb{F}$:



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

Squares with functions

Some relations f fit into the following squares:



(4)

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

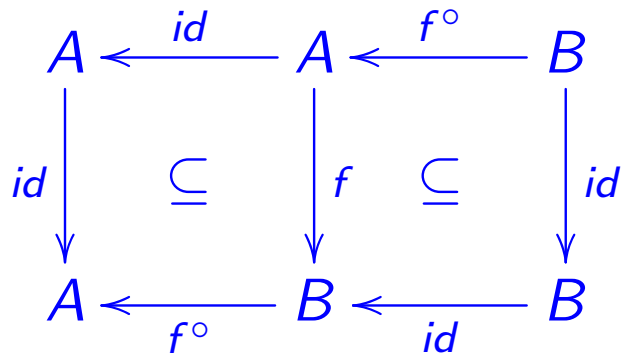
Right square: $\langle \forall\ b, b'\ ::\ \langle \exists\ a\ ::\ b\ f\ a \wedge b'\ f\ a \rangle \Rightarrow (b = b') \rangle$

f is **univocal**.

Such relations f are called **functions**.

Squares with functions

Some relations f fit into the following squares:



(4)

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

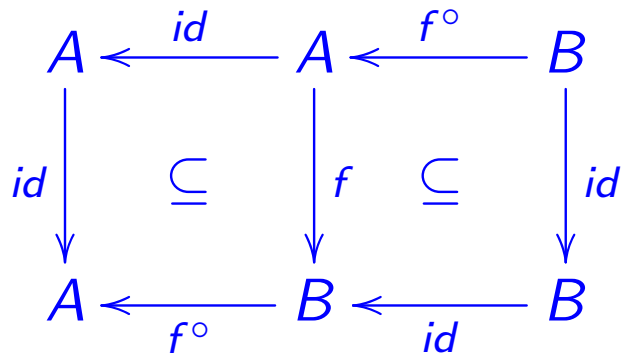
Right square: $\langle \forall\ b, b'\ ::\ \langle \exists\ a\ ::\ b\ f\ a \wedge b'\ f\ a \rangle \Rightarrow (b = b') \rangle$

f is **univocal**.

Such relations f are called **functions**.

Squares with functions

Some relations f fit into the following squares:



(4)

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

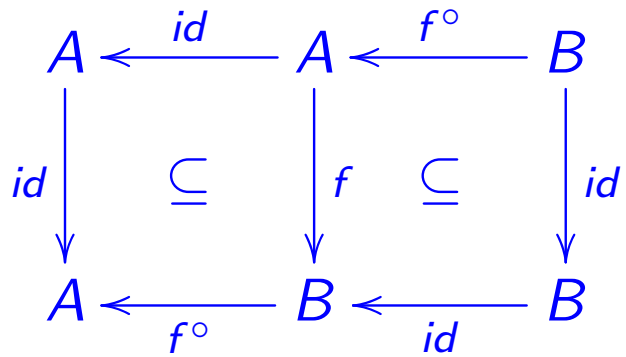
Right square: $\langle \forall\ b, b'\ ::\ \langle \exists\ a\ ::\ b\ f\ a \wedge b'\ f\ a \rangle \Rightarrow (b = b') \rangle$

f is **univocal**.

Such relations f are called **functions**.

Squares with functions

Some relations f fit into the following squares:



(4)

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

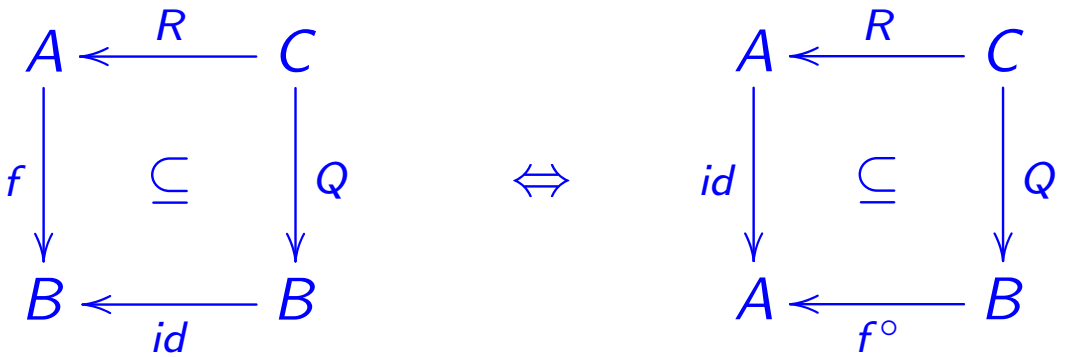
Right square: $\langle \forall\ b, b'\ ::\ \langle \exists\ a\ ::\ b\ f\ a \wedge b'\ f\ a \rangle \Rightarrow (b = b') \rangle$

f is **univocal**.

Such relations f are called **functions**.

Squares with functions

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



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**:

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

Existential quantifiers go away:

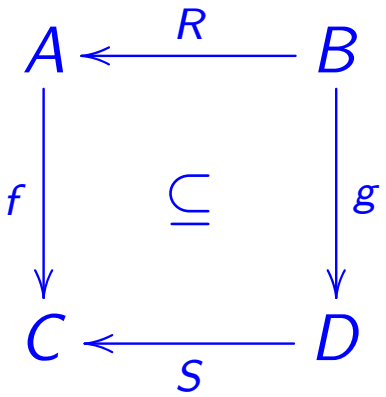
$$b\,(f^{\circ} \cdot R \cdot g)\,a \Leftrightarrow (f\ b)\,R\,(g\ a) \tag{8}$$

$$B \xrightarrow{f} C \xleftarrow{R} D \xleftarrow{g} A, \qquad B \xleftarrow{f^{\circ}\cdot R\cdot g} A$$

☞ Very useful in practice 

Squares with functions

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



$$f \cdot R \subseteq S \cdot g \tag{9}$$

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

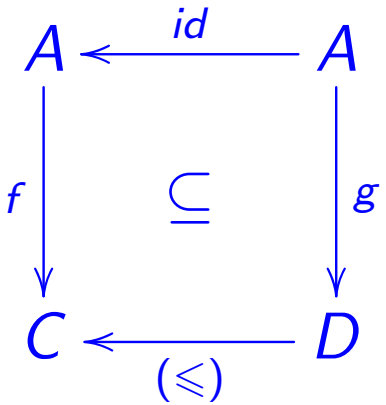
$$f \; S^R \; g \quad \Leftrightarrow \quad f \cdot R \subseteq S \cdot g \tag{10}$$

In words:

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

Squares with functions

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



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

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

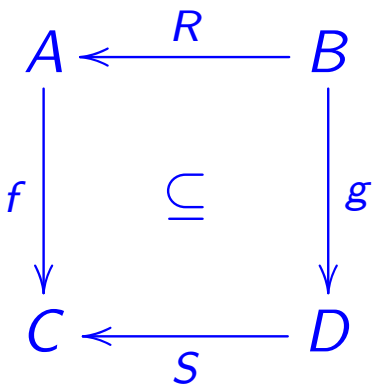
$$f\ (\leqslant)^{id}\ g \quad \Leftrightarrow \quad \langle \forall\ a\ ::\ f\ a \leqslant g\ a \rangle$$

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**:



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.

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

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

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

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

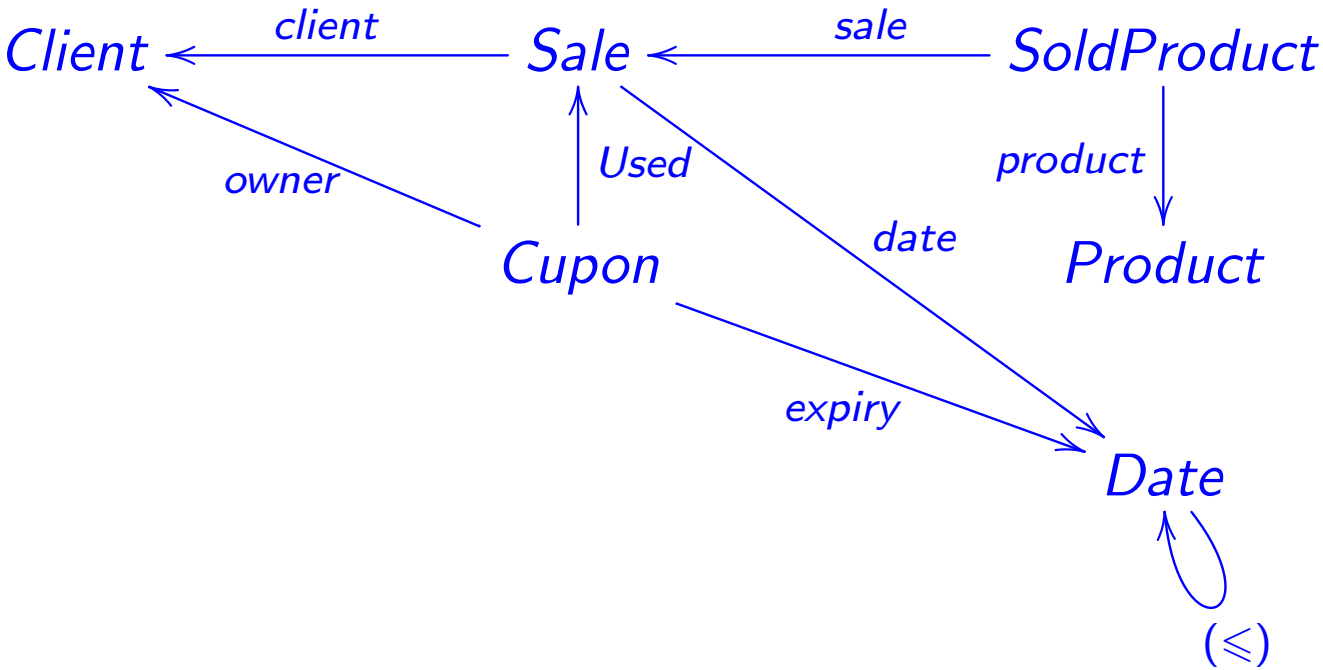
Then, by (11):

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

and so and so forth (very rich construction!).

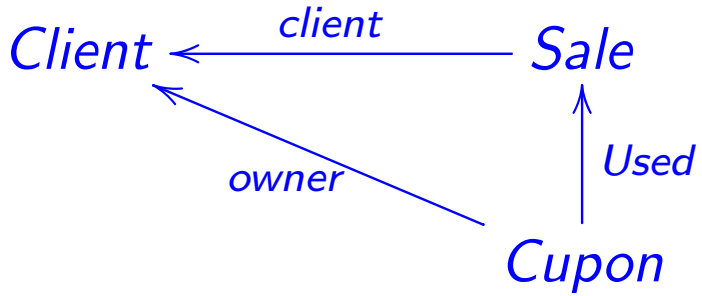
Prosaic applications

"Chase" the squares:

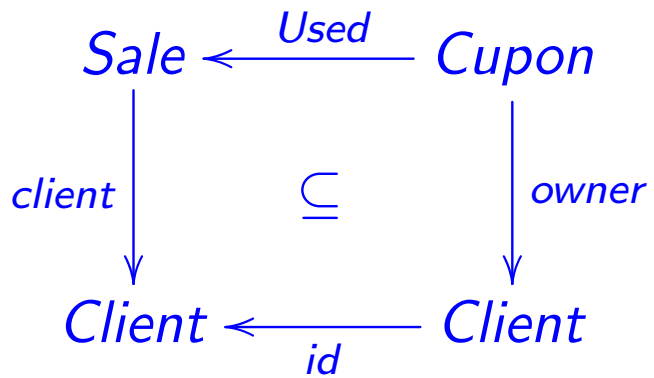


"Chase" the squares

Pick



and try orienting it:

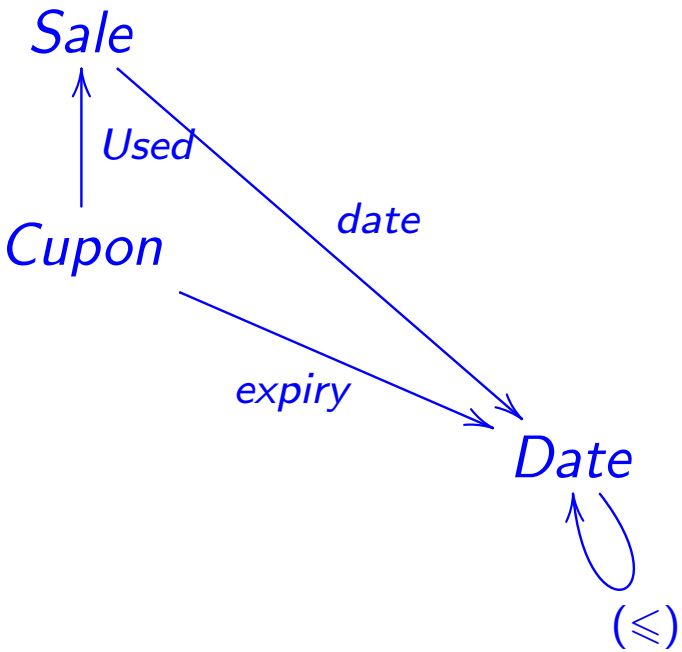


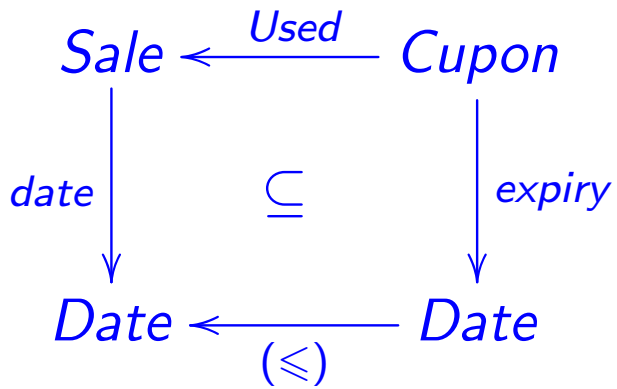
Indeed:

Coupons can only be used by clients who own them.

"Chase" the squares

Now this one:

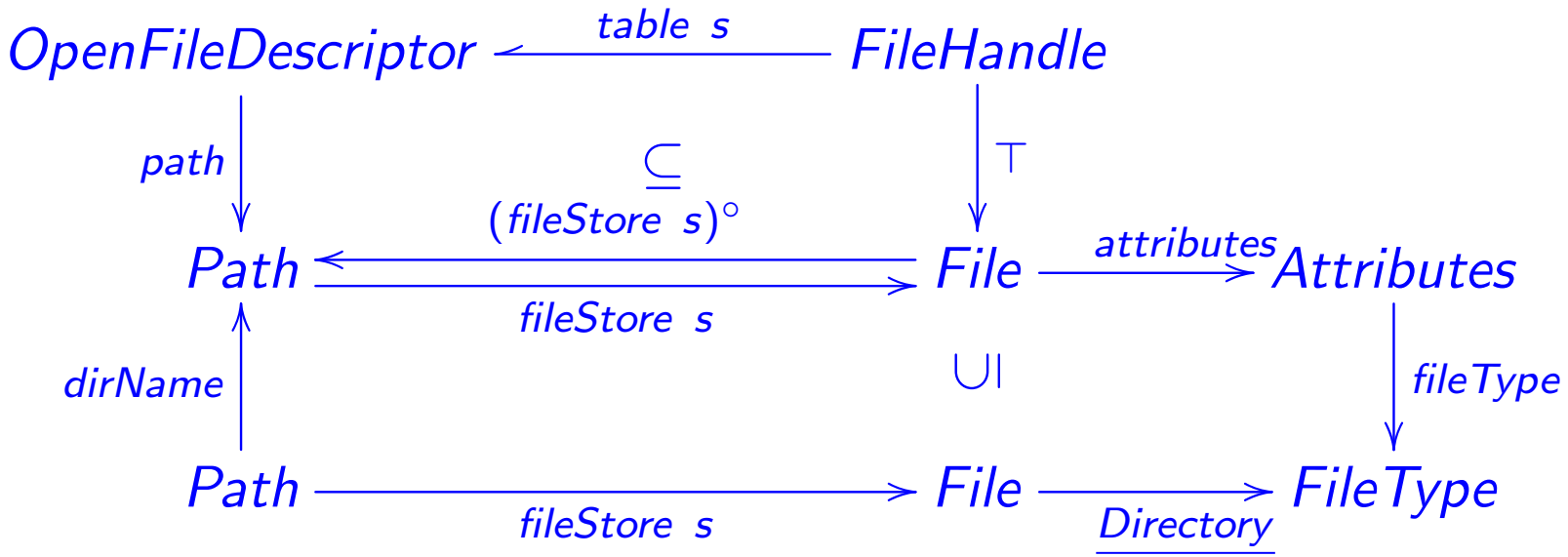




Coupons cannot be used beyond their expiry date.

Formal modelling

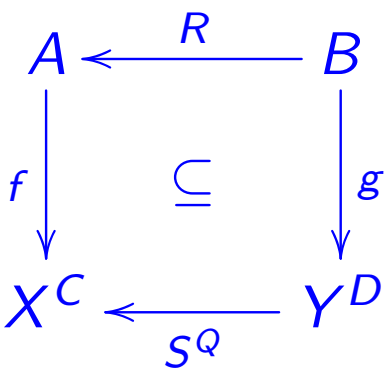
Model invariants as squares, cf. e.g.



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

Higher-order Reynolds squares

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



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

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 }

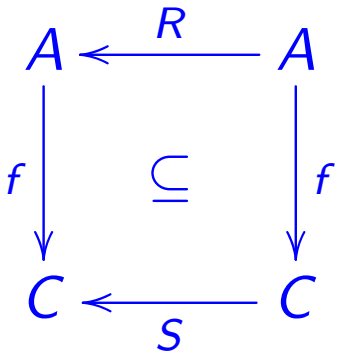
$$a\ R\ b \Rightarrow ((f\ a) \cdot Q \subseteq S \cdot (g\ b))$$

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



$$f \cdot R \subseteq S \cdot f \tag{21}$$

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).

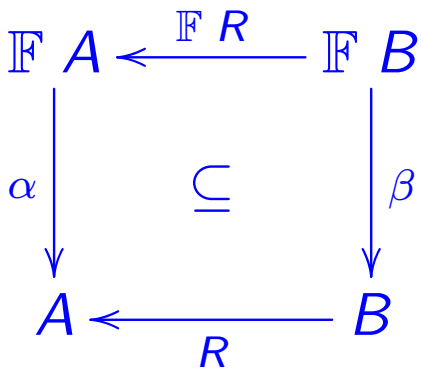
Relational type $R \to S$ corresponds to the homset $Rel_2\,(R, S)$.

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

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

Algebraic squares

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



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

Compare with:

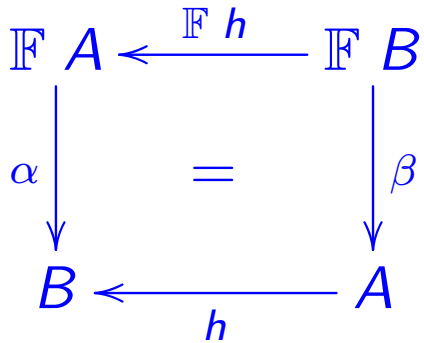
Definition 2.2. *Given a signature Σ and two models, M and N, of the language L generated by Σ, a* (binary) logical relation from M to N *consists of, for each type σ of L, a relation $R_\sigma \subseteq M_\sigma \times N_\sigma$ such that*

- *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 ($R := h$),



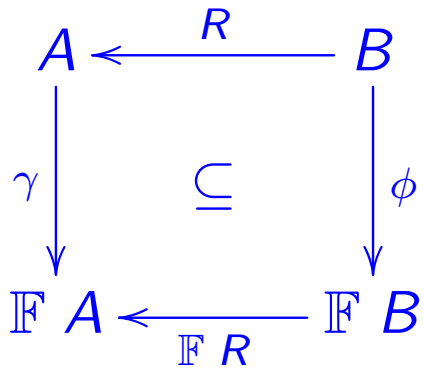
the square means

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

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**:

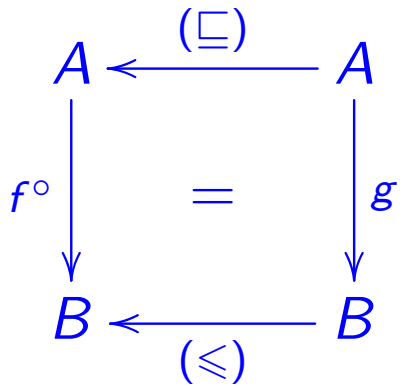


R is said to be a **bisimulation** between the two coalgebras, meaning:

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

Very special squares

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



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

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

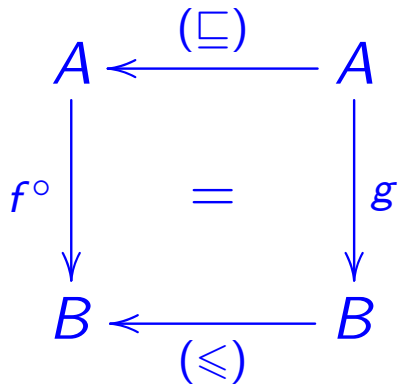
In this very special situation, f and g in

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

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

Very special squares

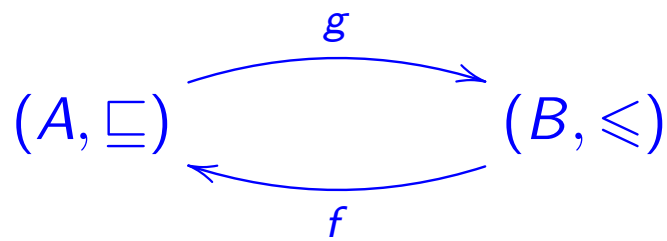
Assume two preorders $(\sqsubseteq)$ and $(\leqslant)$ in:



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

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

In this very special situation, f and g in



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

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

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:

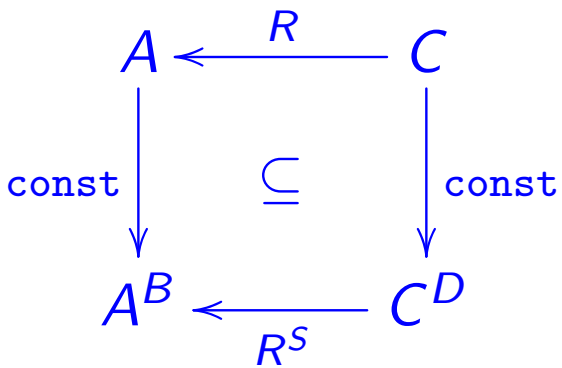
$$\begin{array}{ccc} \mathbb{F}\ A & \xleftarrow{\mathbb{F}\ R} & \mathbb{F}\ B \\ {\scriptstyle f}\downarrow & \subseteq & \downarrow{\scriptstyle f} \\ \mathbb{G}\ A & \xleftarrow[\mathbb{G}\ R]{} & \mathbb{G}\ B \end{array}$$

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

Free theorem squares

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

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



$$\texttt{const} \cdot R \subseteq R^S \cdot \texttt{const} \qquad (25)$$

Pointwise equivalent, recall (19,20):

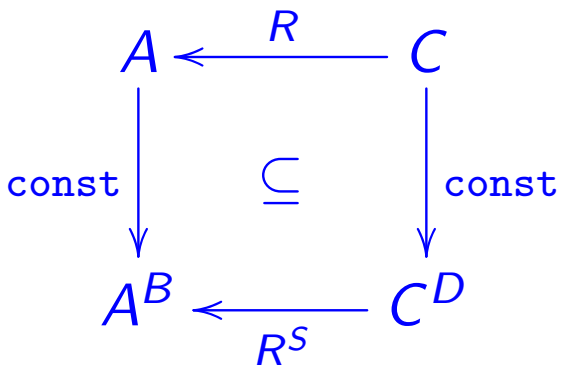
$$a\ R\ c \Rightarrow b\ S\ d \Rightarrow (\texttt{const}\ a\ b)\ R\ (\texttt{const}\ c\ d)$$

for all a, b, c, d:

Free theorem squares

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

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



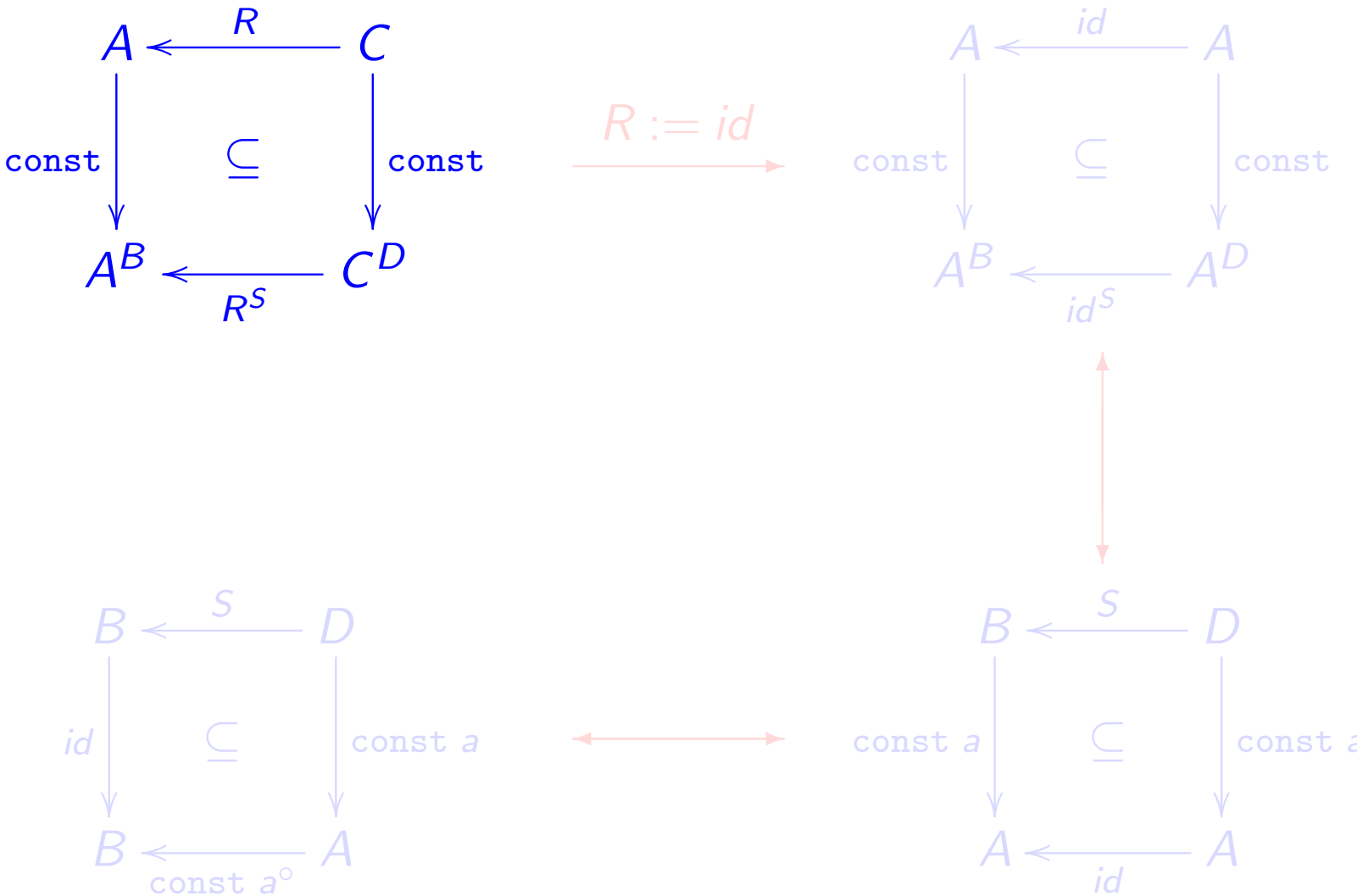
$$\texttt{const} \cdot R \subseteq R^S \cdot \texttt{const} \qquad (25)$$

Pointwise equivalent, recall (19,20):

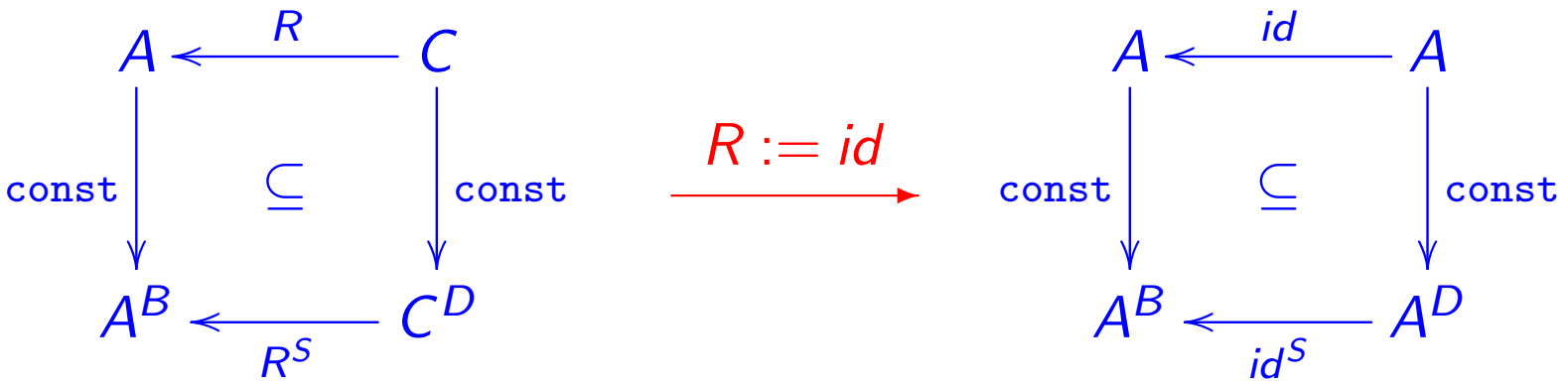
$$a\ R\ c \Rightarrow b\ S\ d \Rightarrow (\texttt{const}\ a\ b)\ R\ (\texttt{const}\ c\ d)$$

for all a, b, c, d:

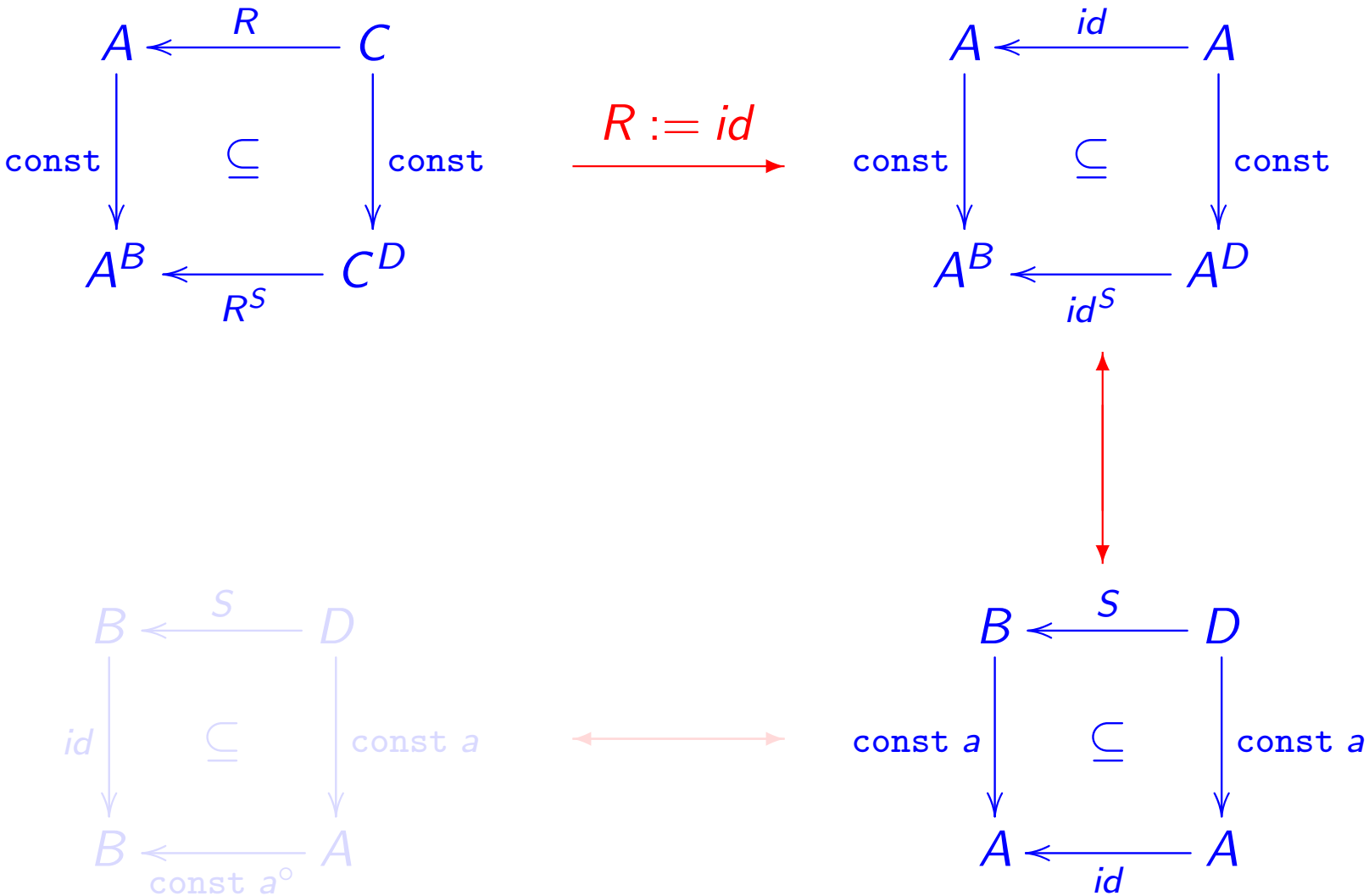
Free theorem squares



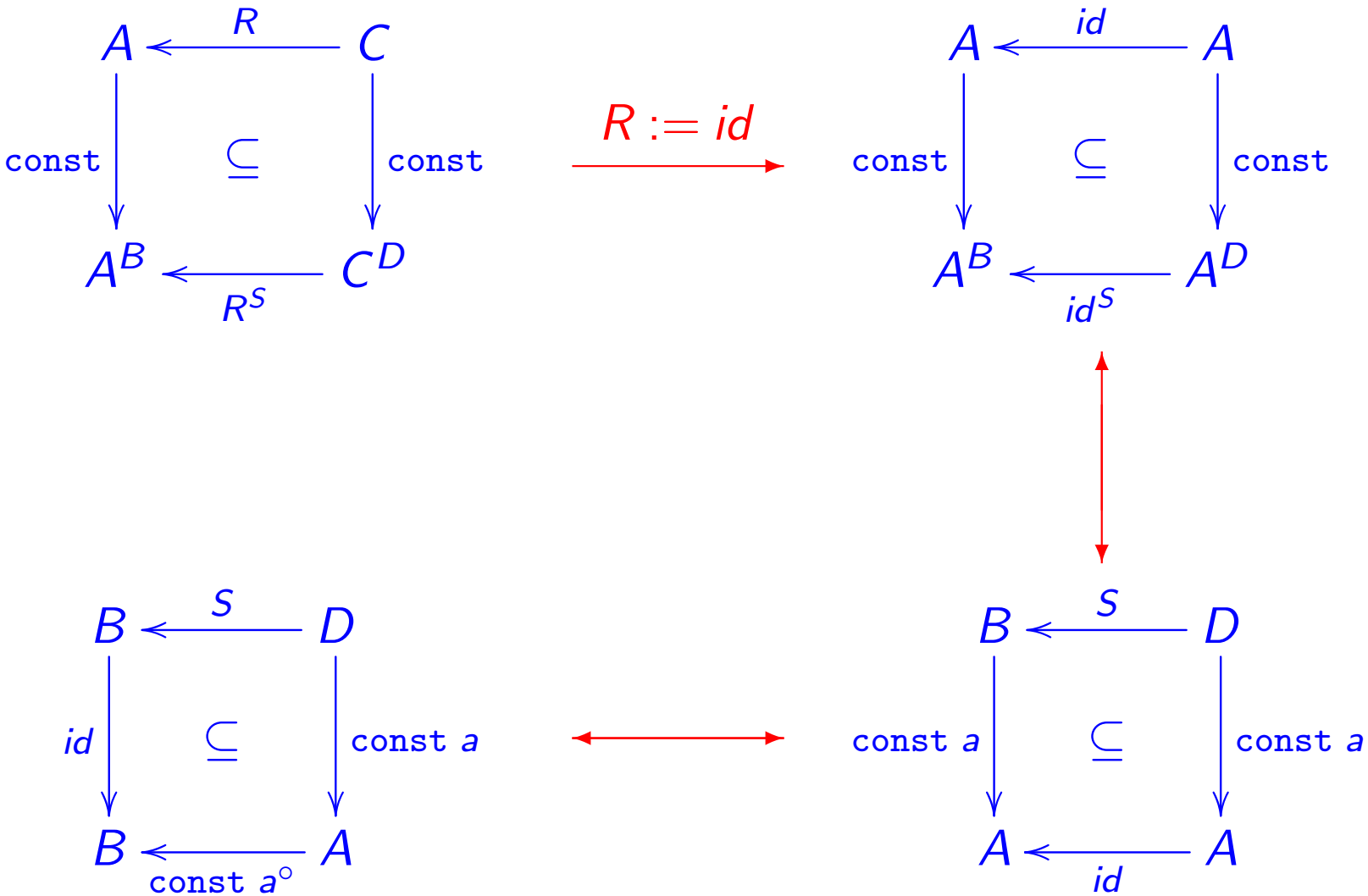
Free theorem squares



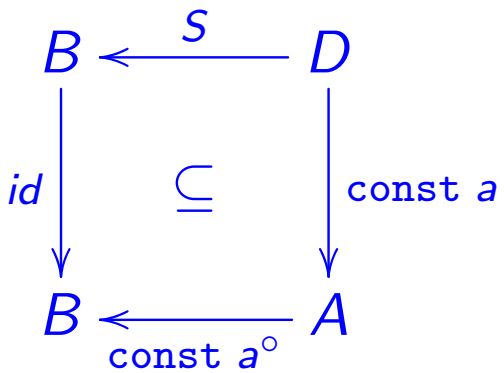
Free theorem squares



Free theorem squares



Free theorem squares



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

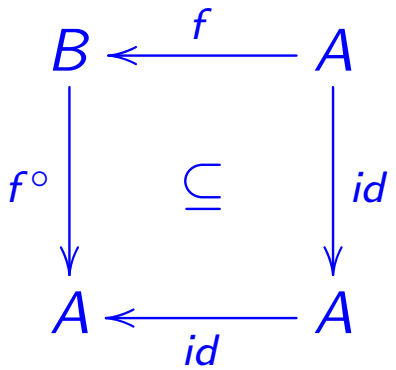
Thus $(\texttt{const}\ a)^\circ \cdot \texttt{const}\ a$ is the largest posible S, i.e. the **top relation** $\top$:

$$(\texttt{const}\ a)^\circ \cdot (\texttt{const}\ a) = \top \tag{26}$$

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

On Injectivity

Injective functions fit in the following square:

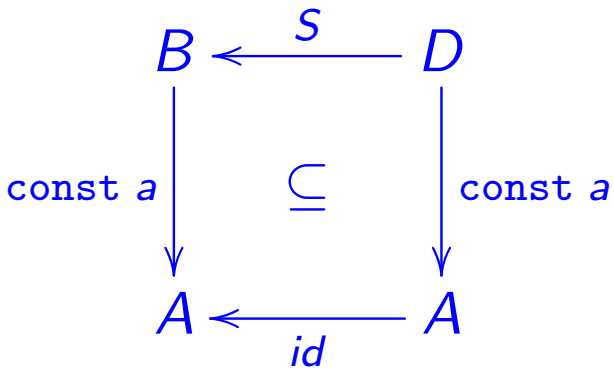


$f^\circ \cdot f$ is the **kernel** of f.

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

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

Free theorem squares



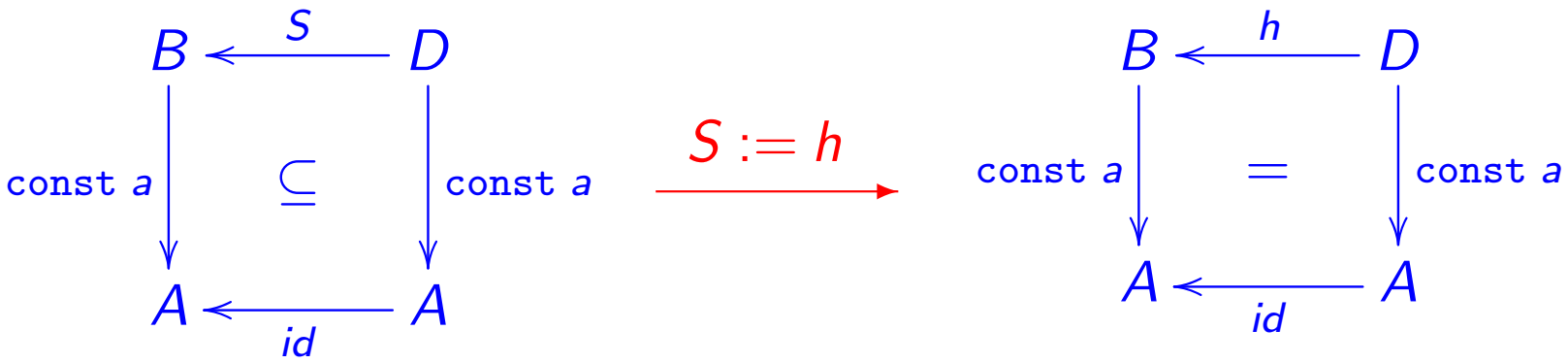
$S := h$

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

$R, S := h, id$

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

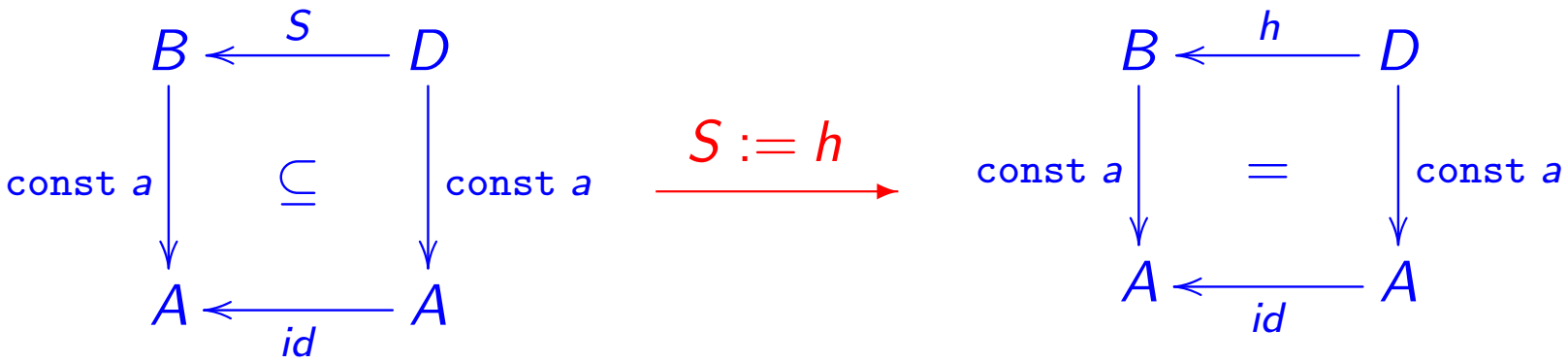
Free theorem squares



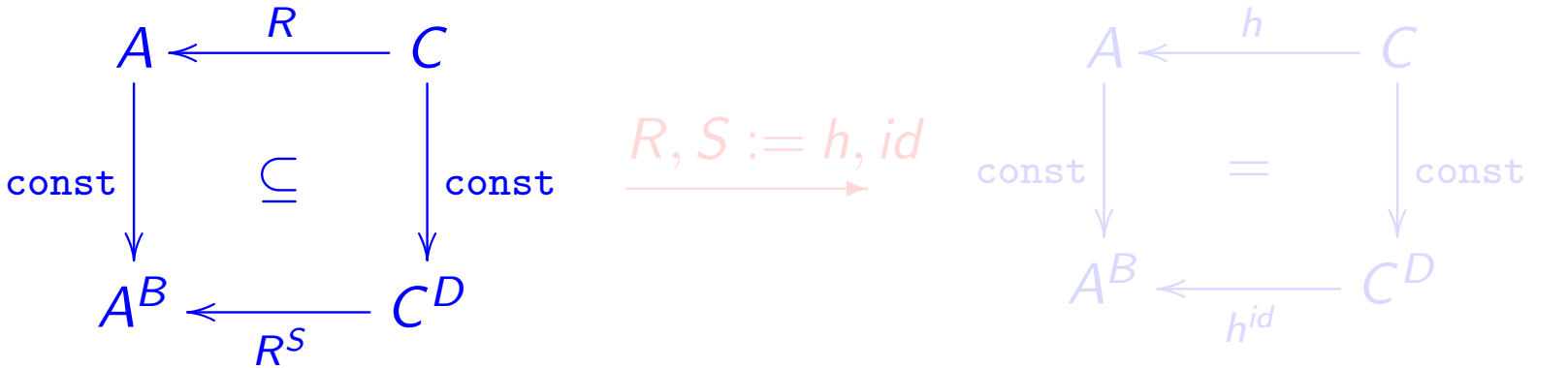
$$\texttt{const}\ a \cdot h = \texttt{const}\ a$$

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

Free theorem squares

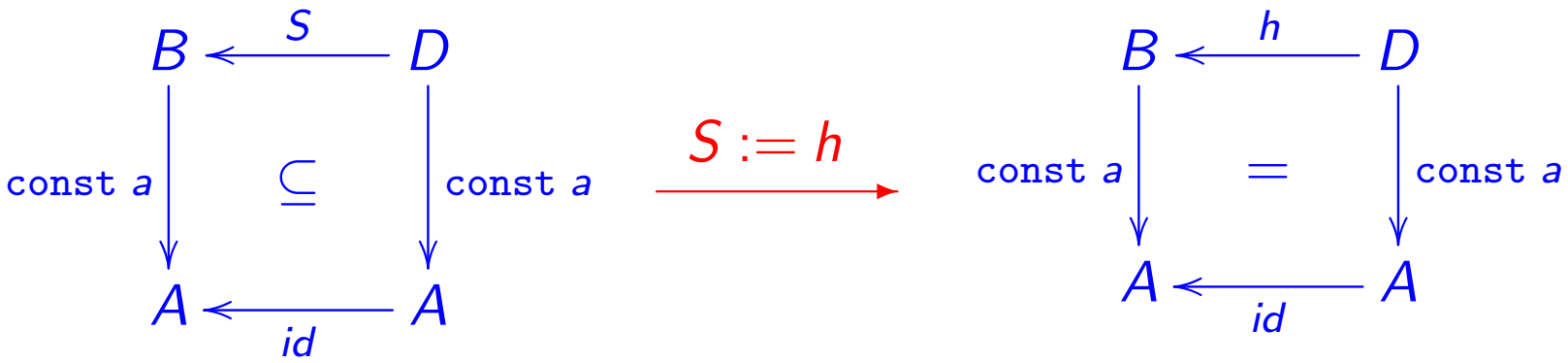


$$\texttt{const}\ a \cdot h = \texttt{const}\ a$$

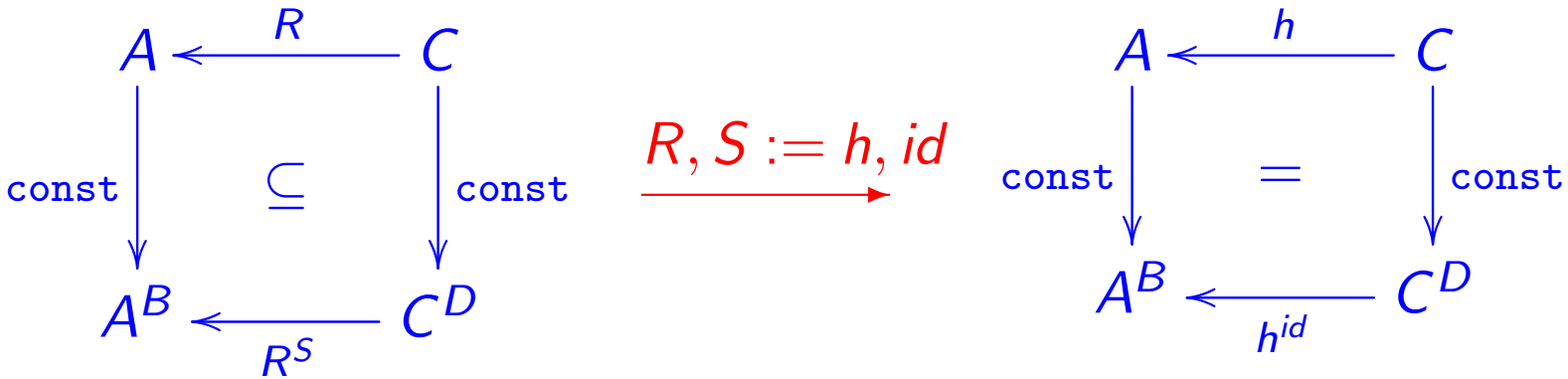


$$h \cdot (\texttt{const}\ a) = \texttt{const}\ (h\ a)$$

Free theorem squares



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



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

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:

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

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

Free theorem squares

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

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

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:

$$\begin{aligned} &\textbf{foldl} :: \mathit{Foldable}\ t \Rightarrow (b \to a \to b) \to b \to t\ a \to b \\ &\textbf{foldr} :: \mathit{Foldable}\ t \Rightarrow (a \to b \to b) \to b \to t\ a \to b \end{aligned} \tag{29}$$

(Hackage's Data.Foldable)

foldl and **foldr** squares

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

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

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

As seen above:

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

foldl squares

The squares of

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

are:

$$\begin{array}{ccc} B & \xleftarrow{S} & Y \\ {\scriptstyle g}\downarrow & \subseteq & \downarrow{\scriptstyle f} \\ B^A & \xleftarrow[S^R]{} & Y^X \end{array} \quad \Rightarrow \quad \begin{array}{ccc} B & \xleftarrow{S} & Y \\ {\scriptstyle \textbf{foldl}\ g}\downarrow & \subseteq & \downarrow{\scriptstyle \textbf{foldl}\ f} \\ B^{\mathbb{T}\,A} & \xleftarrow[S^{\mathbb{T}\,R}]{} & Y^{\mathbb{T}\,X} \end{array} \tag{32}$$

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:

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

Pointwise:

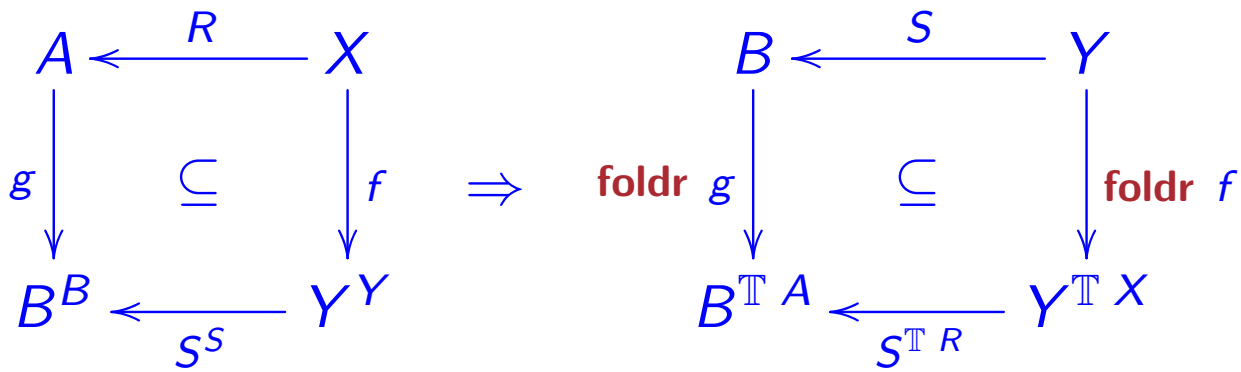
$$h\ (f\ y\ x) = g\ (h\ y)\ x \quad \Rightarrow \quad h\ (\mathbf{foldl}\ f\ y\ xs) = \mathbf{foldl}\ g\ (h\ y)\ xs$$

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

---

foldr

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



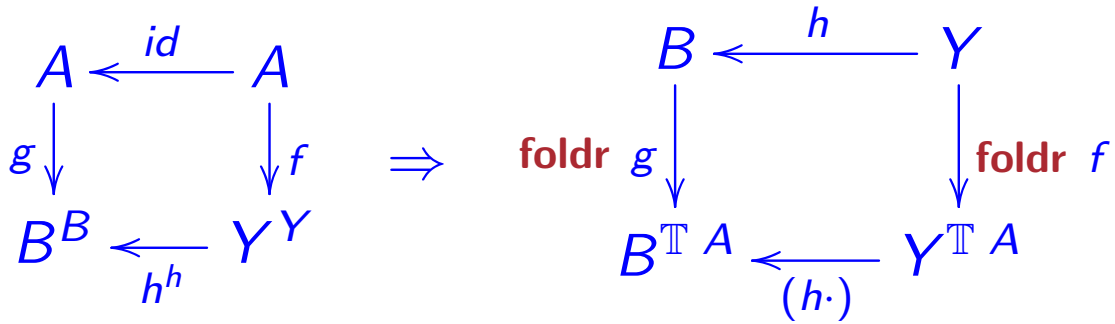
(33)

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

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

foldr squares

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

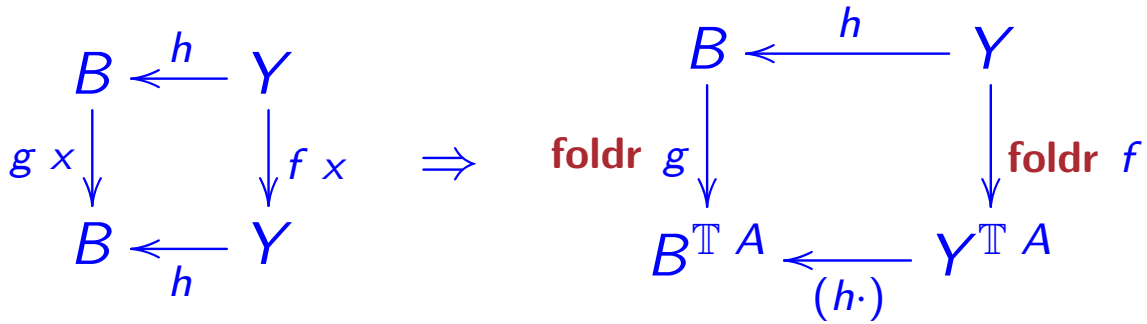


where the side-condition square unfolds to:

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

foldr squares

Altogether, one gets:



that is:

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

Going fully pointwise,

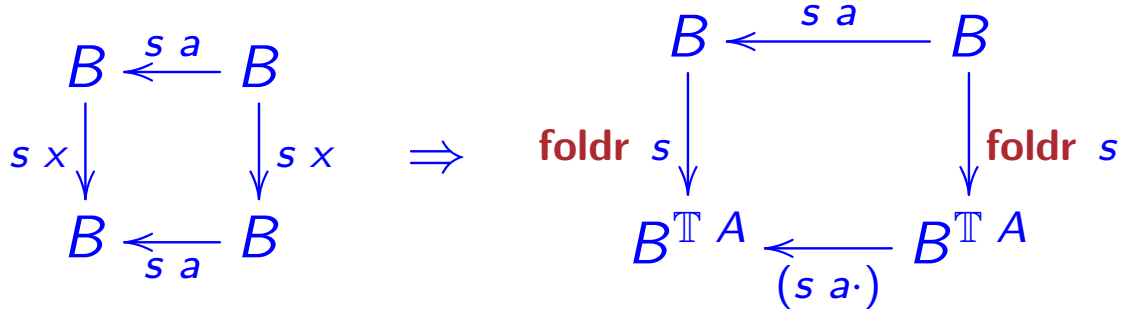
$$g\ x\ (h\ y) = h\ (f\ x\ y) \quad \Rightarrow \quad h\ (\textbf{foldr}\ f\ e\ xs) = \textbf{foldr}\ g\ (h\ e)\ xs \tag{35}$$

---

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 $h := s\ a$



Then (34) becomes:

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

Property

$$(s\ x) \cdot (s\ a) = (s\ a) \cdot (s\ x) \tag{37}$$

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 to the equations

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

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):

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

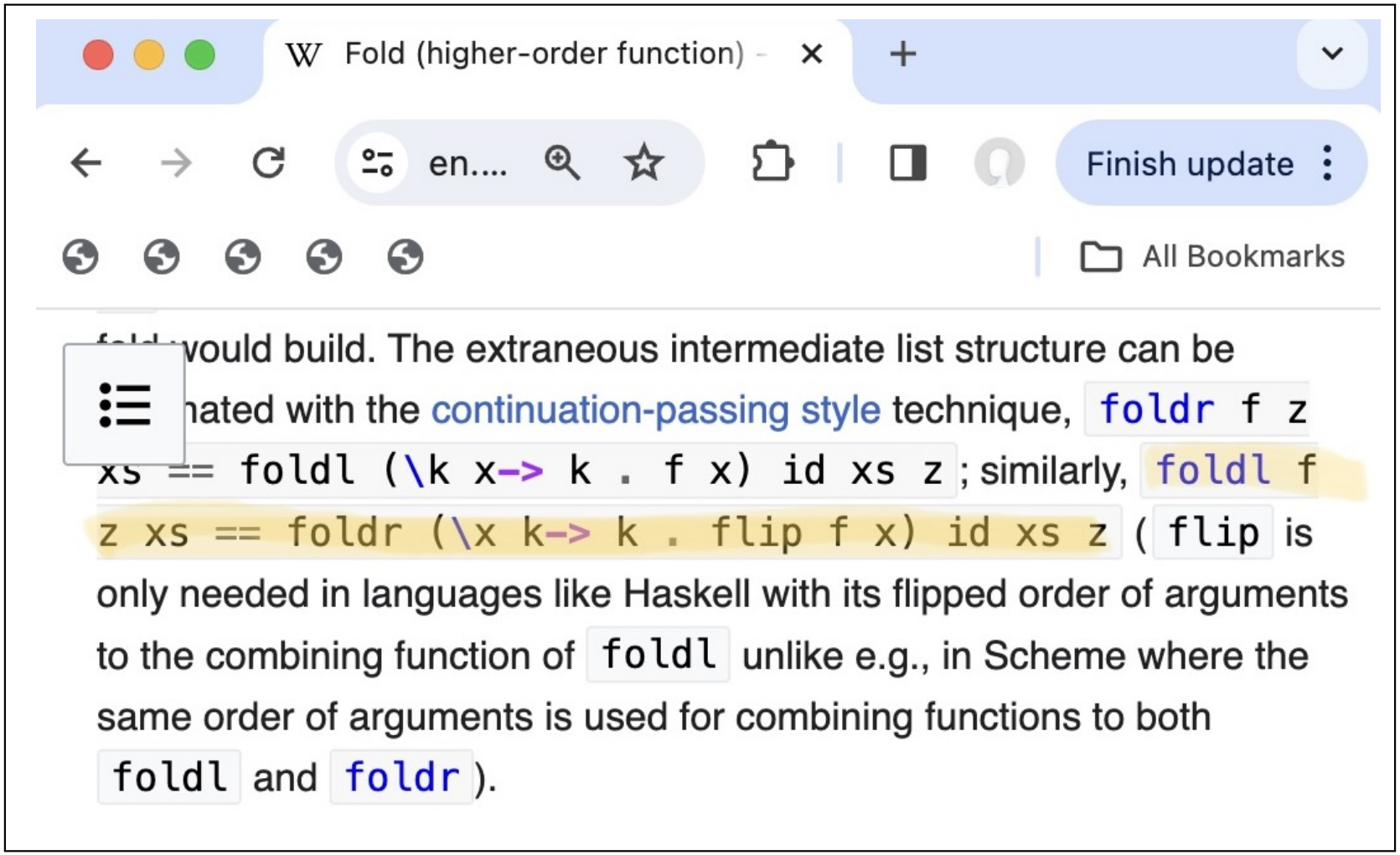
Altogether, (40) and (42) make up a universal property:

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

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

What about **foldl**?

Wikipedia



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

Wikipedia

That is,

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

or

$$\widetilde{\mathbf{foldl}\ f} = \mathbf{foldr}\ (\theta\ f)\ id \tag{45}$$

$$\mathbf{where}\ (\theta\ f)\ x\ k = k \cdot (\widetilde{f}\ x)$$

cf. the square

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

Ok?

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

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

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

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

$\Leftrightarrow$ { definition of θ (46) }

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

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

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

Universal-**foldl**

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

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

Universal-foldl

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

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

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

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

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

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

Universal-**foldl**

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

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

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

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

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

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

The 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?

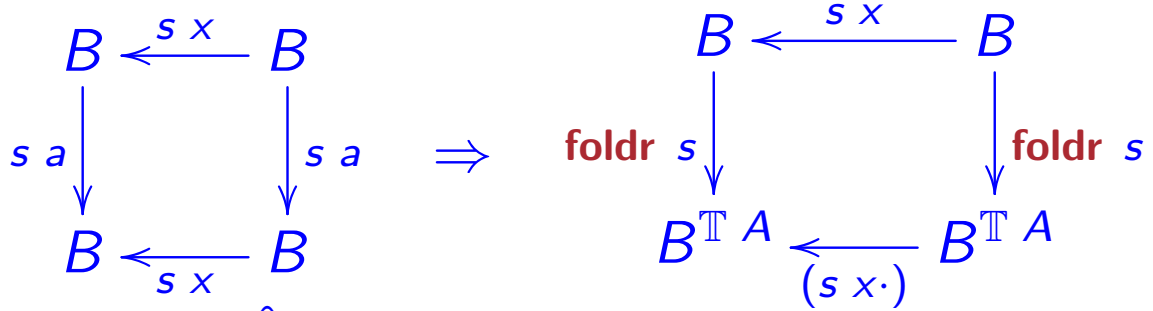
Equating **foldl** and **foldr**

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

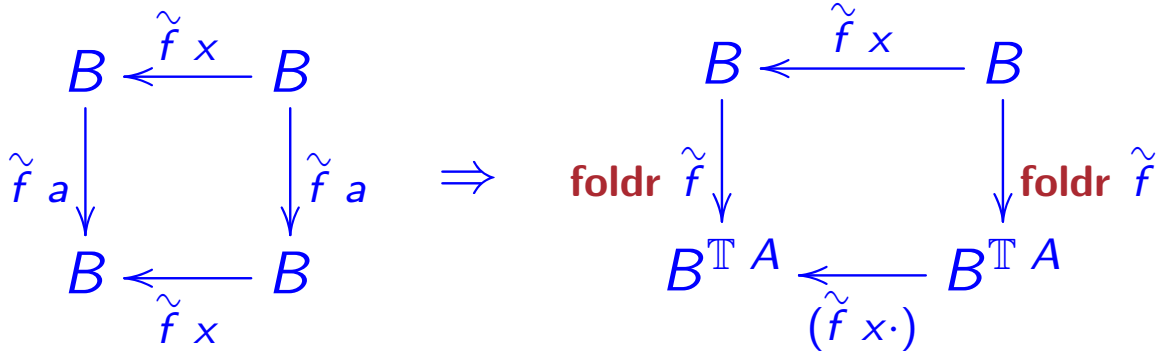
$$
\begin{array}{ll}
 & \mathbf{foldl}\ f = \mathbf{foldr}\ \widetilde{f} \\
\Leftrightarrow & \quad \{\ (47)\ \} \\
 & \left\{ \begin{array}{l} \mathbf{foldr}\ \widetilde{f}\ z\ \gamma = z \\ \mathbf{foldr}\ \widetilde{f}\ z\ (\alpha\ x\ xs) = \mathbf{foldr}\ \widetilde{f}\ (f\ z\ x)\ xs \end{array}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

Back to the permutativity squares

Recall (36)



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



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

Equating foldl and foldr

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

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

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

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

True

Conclusion

We conclude that **foldl** $f = \textbf{foldr}\ \widetilde{f}$ 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**:

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

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

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

that is:

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

Never underestimate indirect equality

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

Summary

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

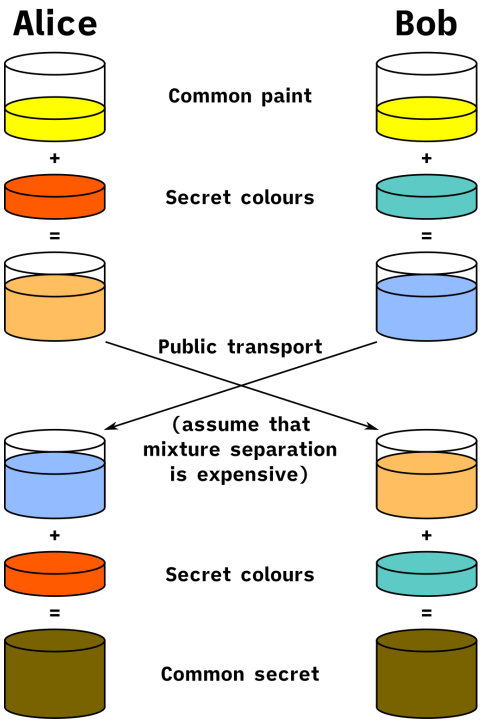
Danvy's reasoning is, however, quite different: he **postulates** permutativity as side condition and then **proves** it in Coq by **list induction**.

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

Moreover, we've shown that a commutative $+$ associative **lower adjoint** f in $f \dashv g$ ensures 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:

$$\mathit{insert} :: \mathit{Ord}\ a \Rightarrow a \to [a] \to [a]$$

Thus **insertion sort**

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

computes the same as

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

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

Annex

On relational exponentials S^R

By vertical composition (2) one immediately infers:

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

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

By horizontal composition (3) we get

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

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

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

Data.Foldable

instance *Foldable* $\mathbb{M}$ **where**

$$\begin{aligned}
&\mathit{foldMap} = \mathit{maybe}\ \mathit{mempty}\\
&\mathbf{foldr}\ _\ z\ \mathit{Nothing} = z\\
&\mathbf{foldr}\ f\ z\ (\mathit{Just}\ x) = f\ x\ z\\
&\mathbf{foldl}\ _\ z\ \mathit{Nothing} = z\\
&\mathbf{foldl}\ f\ z\ (\mathit{Just}\ x) = f\ z\ x
\end{aligned}$$

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

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

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.