PF transform: where everything becomes a relation

J.N. Oliveira

Dept. Informática,
Universidade do Minho
Braga, Portugal

DI/UM, 2007 (last update: Oct-2013)

Motivation

So far, we have been using **Predicate Logic** in formalizing subtleties and complex aspects of real-life problems.

Question: Is this formalism the best for **formal modelling**?

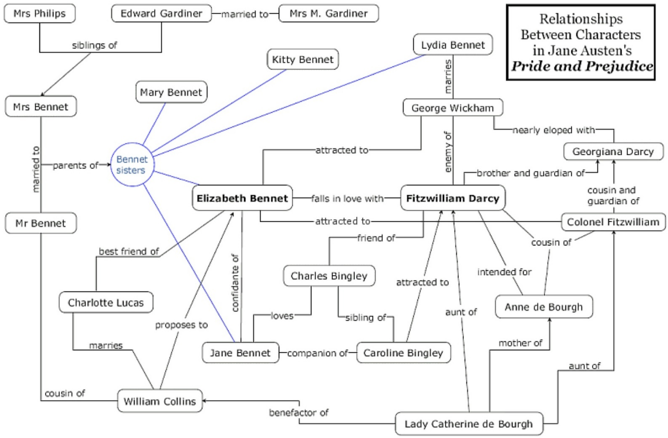
Historically, it was not the first to be proposed:

- Augustus de Morgan (1806-71) — whom you've met already, recall *de Morgan* laws — proposed a **Logic of Relations** as early as 1867.
- Predicate Logic appeared later.

Perhaps de Morgan was right in the first place: in real life, "everything is a **relation**"...

Everything is a relation

... as diagram

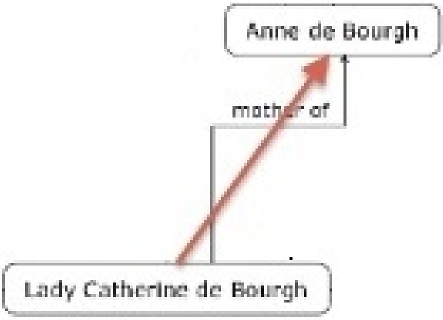


shows. (Wikipedia: **Pride and Prejudice**, by Jane Austin, 1813.)

Arrow notation for relations

The picture is a collection of **relations** — vulg. a **semantic network** — elsewhere known as a (binary) **relational system**.

However, in spite of the use of **arrows** in the picture (aside) not many people would write



$$mother\_of : People \to People$$

as the **type** of **relation** $mother\_of$.

Pairs

Consider assertions

$$\begin{array}{ccc} 0 & \leq & \pi \\ \texttt{Catherine} & \mathit{isMotherOf} & \texttt{Anne} \\ 3 & =(1+) & 2 \end{array}$$

They are statements of fact concerning various kinds of object — real numbers, people, natural numbers, etc

They involve *two* such objects, that is, **pairs**

$$(0, \pi)$$
$$(\texttt{Catherine}, \texttt{Anne})$$
$$(3, 2)$$

respectively.

Sets of pairs

So, we might have written

$$\begin{array}{rcl} (0,\pi) & \in & \leq \\ (\texttt{Catherine},\texttt{Anne}) & \in & \mathit{isMotherOf} \\ (3,2) & \in & (1+) \end{array}$$

What are $(\leq)$, $\mathit{isMotherOf}$, $(1+)$?

- they can be regarded as **sets of pairs**
- better, they should be regarded as **binary relations**.

Therefore,

- **orders** — eg. $(\leq)$ — are special cases of relations
- **functions** — eg. $\mathit{succ} \triangleq (1+)$ — are special cases of relations.

Binary Relations

Binary relations are typed:

Arrow notation

Arrow $A \xrightarrow{R} B$ denotes a binary relation from A (source) to B (target).

A, B are types. Writing $B \xleftarrow{R} A$ means the same as $A \xrightarrow{R} B$.

Infix notation

The usual infix notation used in natural language — eg. `Catherine` *isMotherOf* `Anne` — and in maths — eg. $0 \leq \pi$ — extends to arbitrary $B \xleftarrow{R} A$: we write

$$b \; R \; a$$

to denote that $(b, a) \in R$.

Binary Relations

Binary relations are typed:

Arrow notation

Arrow $A \xrightarrow{R} B$ denotes a binary relation from A (source) to B (target).

A, B are types. Writing $B \xleftarrow{R} A$ means the same as $A \xrightarrow{R} B$.

Infix notation

The usual infix notation used in natural language — eg. $\texttt{Catherine}\ \mathit{isMotherOf}\ \texttt{Anne}$ — and in maths — eg. $0 \leq \pi$ — extends to arbitrary $B \xleftarrow{R} A$: we write

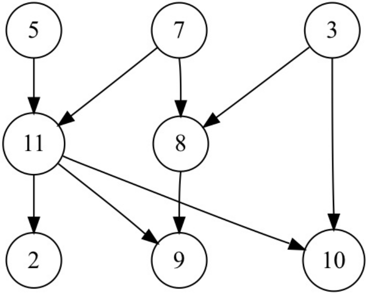
$$b\ R\ a$$

to denote that $(b, a) \in R$.

Binary relations are matrices

Binary relations can be regarded as Boolean **matrices**, eg.

Relation R:



Matrix M:

	1	2	3	4	5	6	7	8	9	10	11
1	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	1
3	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0
8	0	0	1	0	0	0	1	0	0	0	0
9	0	0	0	0	0	0	0	1	0	0	1
10	0	0	1	0	0	0	0	0	0	0	1
11	0	0	0	0	1	0	1	0	0	0	0

In this case $A = B = \{1..11\}$. Relations $A \xleftarrow{R} A$ over a single type are also referred to as (directed) **graphs**.

Functions are relations

- Lowercase letters (or identifiers starting by one such letter) will denote special relations known as **functions**, eg. f, g, $succ$, etc.
- We regard function $f : A \longrightarrow B$ as the binary relation which relates b to a iff $b = f\ a$. So,

$$b\ f\ a \text{ literally means } b = f\ a \tag{36}$$

- Therefore, we generalize

$$\boxed{\begin{array}{c} B \xleftarrow{f} A \\ b = f\ a \end{array}} \quad \text{to} \quad \boxed{\begin{array}{c} B \xleftarrow{R} A \\ b\ R\ a \end{array}}$$

Exercise

Taken from PROPOSITIONES AD ACUENDOS IUUENES ("Problems to Sharpen the Young"), by abbot Alcuin of York († 804):

> XVIII. PROPOSITIO DE HOMINE ET CAPRA ET LVPO.
> *Homo quidam debebat ultra fluuium transferre lupum, capram, et fasciculum cauli. Et non potuit aliam nauem inuenire, nisi quae duos tantum ex ipsis ferre ualebat. Praeceptum itaque ei fuerat, ut omnia haec ultra illaesa omnino transferret. Dicat, qui potest, quomodo eis illaesis transire potuit?*

Exercise

XVIII. FOX, GOOSE AND BAG OF BEANS PUZZLE. *A farmer goes to market and purchases a fox, a goose, and a bag of beans. On his way home, the farmer comes to a river bank and hires a boat. But in crossing the river by boat, the farmer could carry only himself and a single one of his purchases - the fox, the goose, or the bag of the beans. (If left alone, the fox would eat the goose, and the goose would eat the beans.) Can the farmer carry himself and his purchases to the far bank of the river, leaving each purchase intact?*

Identify the main **types** and **relations** involved in the puzzle and draw them in a diagram.

Propositio de homine et capra et lvpo

Data types:

$$\begin{aligned} Being &= \{Farmer, Fox, Goose, Beans\} & (37) \\ Bank &= \{Left, Right\} & (38) \end{aligned}$$

Relations:

$$\begin{tikzcd} Being \arrow[r, "Eats"] & Being \arrow[d, "where"'] & \\ & Bank \arrow[r, "cross"] & Bank \end{tikzcd} \quad (39)$$

Composition

Recall **function composition**

$$\begin{tikzcd} B & A \arrow[l, "f"'] & C \arrow[l, "g"'] \arrow[ll, bend left, "f \cdot g"] \end{tikzcd} \quad (40)$$

$$b = f(g\ c)$$

We extend $f \cdot g$ to $R \cdot S$ in the obvious way:

$$b(R \cdot S)c \quad \equiv \quad \langle \exists\ a\ ::\ b\ R\ a\ \wedge\ a\ S\ c \rangle \quad (41)$$

Note how this rule of the **PF**[1]-transform *removes* $\exists$ when applied from right to left

[1] PF stands for "point-free".

Check generalization

Back to functions, (41) becomes

$$
\begin{array}{rcl}
b(f \cdot g)c & \equiv & \langle \exists\, a \;::\; b\ f\ a \wedge a\ g\ c \rangle \\
& \equiv & \qquad \{\ a\ g\ c \text{ means } a = g\ c \text{ (36)}\ \} \\
& & \langle \exists\, a \;::\; b\ f\ a \wedge a = g\ c \rangle \\
& \equiv & \qquad \{\ \exists\text{-trading ; } b\ f\ a \text{ means } b = f\ a \text{ (36)}\ \} \\
& & \langle \exists\, a \;:\; a = g\ c :\; b = f\ a \rangle \\
& \equiv & \qquad \{\ \text{one-point rule } (\exists)\ \} \\
& & b = f(g\ c)
\end{array}
$$

So, we easily recover what we had before (40).

Inclusion generalizes equality

- **Equality** on functions

$$f = g \quad \equiv \quad \langle \forall\ a\ :\ a \in A :\ f\ a\ =_B\ g\ a \rangle \tag{42}$$

generalizes to **inclusion** on relations:

$$R \subseteq S \quad \equiv \quad \langle \forall\ b, a\ :\ b\ R\ a :\ b\ S\ a \rangle \tag{43}$$

(read $R \subseteq S$ as "R is at most S")

- For $R \subseteq S$ to hold both need to be of the same type, say $B \xleftarrow{R,S} A$
- $R \subseteq S$ is a partial order (reflexive, transitive and anti-symmetric).

Special relations

Every type $B \longleftarrow A$ has its

- *bottom* relation $B \xleftarrow{\perp} A$, which is such that, for all b, a, $b \perp a \equiv \textsc{False}$
- *topmost* relation $B \xleftarrow{\top} A$, which is such that, for all b, a, $b \top a \equiv \textsc{True}$

Type $A \longleftarrow A$ has the

- *identity* relation $A \xleftarrow{id} A$ which is nothing but the function

$$id\ a \triangleq a$$

Clearly, for every R,

$$\perp \subseteq R \subseteq \top \tag{44}$$

Exercises

Exercise 18: Resort to (42), (43) and to the Eindhoven quantifier calculus to show that

$$f \subseteq g \quad \equiv \quad f = g$$

holds (moral: for functions, inclusion and equality coincide).
□

Exercise 19: Resort to PF-transform rule (41) and to the Eindhoven quantifier calculus to show that

$$R \cdot id \;=\; R \;=\; id \cdot R \tag{45}$$

$$R \cdot \bot \;=\; \bot \;=\; \bot \cdot R \tag{46}$$

hold and that composition is associative:

$$R \cdot (S \cdot T) \;=\; (R \cdot S) \cdot T \tag{47}$$

□

Converses

Every relation $B \xleftarrow{R} A$ has a **converse** $B \xrightarrow{R^\circ} A$ which is such that, for all a, b,

$$a(R^\circ)b \equiv b\ R\ a \tag{48}$$

Note that converse commutes with composition

$$(R \cdot S)^\circ = S^\circ \cdot R^\circ \tag{49}$$

and with itself:

$$(R^\circ)^\circ = R \tag{50}$$

Converse captures the **passive voice**: *Catherine eats the apple* — $R = (eats)$ — the same as *the apple is eaten by Catherine* — $R^\circ = (is\ eaten\ by)$.

Function converses

Function converses f°, g° etc. always exist (as **relations**) and enjoy the following (very useful) PF-transform property:

$$(f\ b)R(g\ a) \quad \equiv \quad b(f^\circ \cdot R \cdot g)a \tag{51}$$

cf. diagram:

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

Let us see an example of its use.

PF-transform at work

Transforming a well-known PW-formula:

$$
\begin{array}{ll}
 & f \text{ is injective} \\
\equiv & \quad \{ \text{ recall definition from discrete maths } \} \\
 & \langle \forall\ y, x\ :\ (f\ y) = (f\ x) :\ y = x \rangle \\
\equiv & \quad \{ \text{ introduce } id \text{ (twice) } \} \\
 & \langle \forall\ y, x\ :\ (f\ y) id (f\ x) :\ y (id) x \rangle \\
\equiv & \quad \{ \text{ rule } (f\ b) R (g\ a) \equiv b (f^{\circ} \cdot R \cdot g) a \text{ (51) } \} \\
 & \langle \forall\ y, x\ :\ y (f^{\circ} \cdot id \cdot f) x :\ y (id) x \rangle \\
\equiv & \quad \{ \text{ (45) ; then go pointfree via (43) } \} \\
 & f^{\circ} \cdot f \subseteq id
\end{array}
$$

The other way round

Let us now see what $id \subseteq f \cdot f^\circ$ means:

$$
\begin{array}{ll}
& id \subseteq f \cdot f^\circ \\
\equiv & \qquad \{ \text{ relational inclusion (43) } \} \\
& \langle \forall\, y, x \;:\; y(id)x :\; y(f \cdot f^\circ)x \rangle \\
\equiv & \qquad \{ \text{ identity relation ; composition (41) } \} \\
& \langle \forall\, y, x \;:\; y = x :\; \langle \exists\, z \;::\; y\ f\ z \wedge z\ f^\circ x \rangle \rangle \\
\equiv & \qquad \{ \text{ converse (48) } \} \\
& \langle \forall\, y, x \;:\; y = x :\; \langle \exists\, z \;::\; y\ f\ z \wedge x\ f\ z \rangle \rangle \\
\equiv & \qquad \{ \ \forall\text{-one point ; trivia ; function } f \ \} \\
& \langle \forall\, x \;::\; \langle \exists\, z \;::\; x = f\ z \rangle \rangle \\
\equiv & \qquad \{ \text{ recalling definition from maths } \} \\
& f \text{ is surjective}
\end{array}
$$

Why *id* (really) matters

Terminology:

- Say R *is reflexive* iff $id \subseteq R$
  pointwise: $\langle \forall\ a\ ::\ a\ R\ a \rangle$ (check as homework);
- Say R *is coreflexive* (or *diagonal*) iff $R \subseteq id$
  pointwise: $\langle \forall\ b, a\ :\ b\ R\ a\ :\ b = a \rangle$ (check as homework).

Define, for $B \xleftarrow{R} A$:

Kernel of R	**Image** of R
$A \xleftarrow{\ker R} A$	$B \xleftarrow{\mathrm{img}\ R} B$
$\ker R \stackrel{\mathrm{def}}{=} R^\circ \cdot R$	$\mathrm{img}\ R \stackrel{\mathrm{def}}{=} R \cdot R^\circ$

Example: kernels of functions

$$
\begin{array}{ll}
& a'(\ker f)a \\
\equiv & \{ \text{ substitution } \} \\
& a'(f^\circ \cdot f)a \\
\equiv & \{ \text{ PF-transform rule (51) } \} \\
& (f\ a') = (f\ a)
\end{array}
$$

In words: $a'(\ker f)a$ means a' *and* a *"have the same* f*-image"*

Exercise 20: Let K be a nonempty data domain, $k \in K$ and $\underline{k}$ be the *"everywhere* k*"* function:

$$
\begin{array}{rcl}
\underline{k} & : & A \longrightarrow K \\
\underline{k}\,a & \triangleq & k
\end{array}
\tag{52}
$$

Compute which relations are defined by the following PF-expressions:

$$
\ker \underline{k} \quad , \quad \underline{b} \cdot \underline{c}^\circ \quad , \quad \mathrm{img}\,\underline{k}
\tag{53}
$$

Binary relation taxonomy

Topmost criteria:

binary relation

injective entire simple surjective

Definitions:

	Reflexive	*Coreflexive*
$\ker R$	entire R	injective R
$\operatorname{img} R$	surjective R	simple R

(54)

Facts:

$$\ker(R^\circ) = \operatorname{img} R \quad (55)$$

$$\operatorname{img}(R^\circ) = \ker R \quad (56)$$

Binary relation taxonomy

The whole picture:

$$
\begin{tikzcd}[column sep=tiny]
 & & & \text{binary relation} \arrow[dlll, dash] \arrow[dl, dash] \arrow[dr, dash] \arrow[drrr, dash] & & & \\
\text{injective} \arrow[dr, dash] & & \text{entire} \arrow[dl, dash] \arrow[dr, dash] & & \text{simple} \arrow[dl, dash] \arrow[dr, dash] & & \text{surjective} \arrow[dl, dash] \\
 & \text{representation} \arrow[dr, dash] & & \text{function} \arrow[dl, dash] \arrow[dr, dash] & & \text{abstraction} \arrow[dl, dash] & \\
 & & \text{injection} \arrow[dr, dash] & & \text{surjection} \arrow[dl, dash] & & \\
 & & & \text{bijection} & & &
\end{tikzcd}
\tag{57}
$$

Exercise 21: Resort to (55,56) and (54) to prove the following rules of thumb:

- converse of **injective** is **simple** (and vice-versa)
- converse of **entire** is **surjective** (and vice-versa)

□

Exercise

Exercise 22: Prove the following fact

$$\textit{A function } f \textit{ is a bijection } \textbf{iff} \textit{ its converse } f^\circ \textit{ is a function} \tag{58}$$

by completing:

$$
\begin{array}{ll}
 & f \text{ and } f^\circ \text{ are functions} \\
\equiv & \quad \{ \ \ldots \ \} \\
 & (id \subseteq \ker f \wedge \operatorname{img} f \subseteq id) \wedge (id \subseteq \ker (f^\circ) \wedge \operatorname{img} (f^\circ) \subseteq id) \\
\equiv & \quad \{ \ \ldots \ \} \\
 & \vdots \\
\equiv & \quad \{ \ \ldots \ \} \\
 & f \text{ is a bijection}
\end{array}
$$

□

PROPOSITIO DE HOMINE ET CAPRA ET LVPO

Exercise 23: Check which of the following properties,

simple , *entire* , *injective* , *surjective* , *transitive* , *(co)reflexive* , *(anti)symmetric* , *connected*

hold for relation $Eats$ (39), which is the food chain $Fox > Goose > Beans$.

□

Exercise 24: Relation $cross$ (39) is defined by:

$$\begin{array}{rcl} cross\ Left & = & Right \\ cross\ Right & = & Left \end{array}$$

It therefore is a bijection. Why?

□

PROPOSITIO DE HOMINE ET CAPRA ET LVPO

Exercise 25: Relation $where : Being \to Bank$ should obey the following constraints:

- *everyone is somewhere in a bank*
- *no one can be in both banks at the same time.*

Encode such constraints in relational terms. Conclude that $where$ should be a function.

□

Exercise 26: There are only two constant functions in the type $Being \longrightarrow Bank$. Identify them and explain the role they play in the puzzle.

□

Functions in one slide

Recall that a function f is a binary relation such that

Pointwise	Pointfree	
"Left" Uniqueness		
$b\ f\ a \wedge b'\ f\ a \quad \Rightarrow \quad b = b'$	$\mathrm{img}\, f \quad \subseteq \quad id$	(f is simple)
Leibniz principle		
$a = a' \quad \Rightarrow \quad f\ a = f\ a'$	$id \quad \subseteq \quad \ker f$	(f is entire)

NB: Following a widespread convention, functions will be denoted by lowercase characters (eg. f, g, ϕ) or identifiers starting with lowercase characters, and function application will be denoted by juxtaposition, eg. $f\ a$ instead of $f(a)$.

Functions, relationally

Shunting rules:

$$f \cdot R \subseteq S \quad \equiv \quad R \subseteq f^{\circ} \cdot S \tag{59}$$

$$R \cdot f^{\circ} \subseteq S \quad \equiv \quad R \subseteq S \cdot f \tag{60}$$

Equality:

$$f \subseteq g \equiv f = g \equiv f \supseteq g \tag{61}$$

"Cyclic inclusion" calculation of the equality rule (61) follows (recall exercise 18):

Proof of functional equality

$$
\begin{array}{ll}
 & f \subseteq g \\
\equiv & \quad \{ \text{ identity } \} \\
 & f \cdot id \subseteq g \\
\equiv & \quad \{ \text{ shunting on } f \ \} \\
 & id \subseteq f^{\circ} \cdot g \\
\equiv & \quad \{ \text{ shunting on } g \ \} \\
 & id \cdot g^{\circ} \subseteq f^{\circ} \\
\equiv & \quad \{ \text{ converses; identity } \} \\
 & g \subseteq f
\end{array}
$$

Exercises

Exercise 27: Infer $id \subseteq \ker f$ (f is total) and $\operatorname{img} f \subseteq id$ (f is simple) from any of shunting rules (59) or (60).

□

Exercise 28: Check the meaning of shunting rules (59) and (60) by converting them to pointwise (Eindhoven) notation.

Show that they indeed hold by resorting to the rules of the Eindhoven calculus.

□

Exercise 29: Let $s\ S\ n$ mean: *"student s is assigned number n"*. Check the meaning of assertion: $S \cdot \leq^{\circ} \subseteq \top \cdot S$.

□

Exercises

Exercise 30: As generalization of exercise 29, draw the most general type diagram which accommodates relational assertion:

$$M \cdot R^\circ \quad \subseteq \quad \top \cdot M \tag{62}$$

□

Exercise 31: Type the following relational assertions

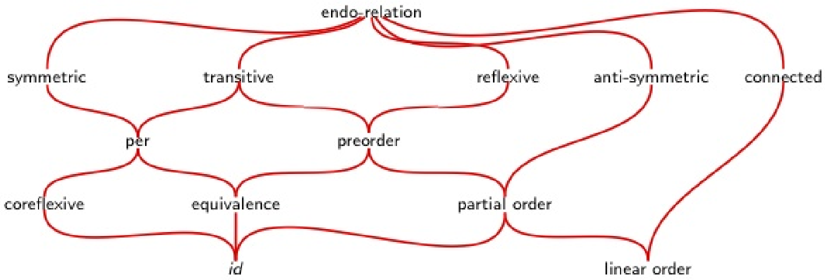
$$M \cdot N^\circ \quad \subseteq \quad \bot \tag{63}$$
$$M \cdot N^\circ \quad \subseteq \quad id \tag{64}$$
$$M^\circ \cdot \top \cdot N \quad \subseteq \quad > \tag{65}$$

and check their pointwise meaning.

□

Relation taxonomy — orders

Orders are endo-relations $A \xleftarrow{R} A$ classified as



(Criteria definitions: next slide)

Orders and their taxonomy

Besides

reflexive:	iff $id_A \subseteq R$
coreflexive:	iff $R \subseteq id_A$

an order (or endo-relation) $A \xleftarrow{R} A$ can be

transitive:	iff $R \cdot R \subseteq R$
anti-symmetric:	iff $R \cap R^\circ \subseteq id_A$
symmetric:	iff $R \subseteq R^\circ (\equiv R = R^\circ)$
connected:	iff $R \cup R^\circ = \top$

Orders and their taxonomy

Therefore:

- **Preorders** are reflexive and transitive orders.
  Example: $y \ \mathit{IsAtMostAsOldAs} \ x$
- **Partial** orders are anti-symmetric preorders
  Example: $y \subseteq x$
- **Linear** orders are connected partial orders
  Example: $y \leq x$
- **Equivalences** are symmetric preorders
  Example: $y \ \mathit{Permutes} \ x$
- **Pers** are partial equivalences
  Example: $y \ \mathit{IsBrotherOf} \ x$

Exercises

Exercise 32: Expand all criteria in the previous slides to pointwise notation.

□

Exercise 33: A relation R is said to be *co-transitive* iff the following holds:

$$\langle \forall\ b, a\ :\ b\ R\ a :\ \langle \exists\ c\ :\ b\ R\ c :\ c\ R\ a \rangle \rangle \tag{66}$$

Compute the PF-transform of the formula above. Find a relation (eg. over numbers) which is co-transitive and another which is not.

□

Meet and join

Meet (intersection) and join (union) lift pointwise conjunction and disjunction, respectively,

$$b\,(R \cap S)\,a \;\equiv\; b\,R\,a \wedge b\,S\,a \tag{67}$$

$$b\,(R \cup S)\,a \;\equiv\; b\,R\,a \vee b\,S\,a \tag{68}$$

for R, S of the same type. Their meaning is captured by the following **universal** properties:

$$X \subseteq R \cap S \;\equiv\; X \subseteq R \;\wedge\; X \subseteq S \tag{69}$$

$$R \cup S \subseteq X \;\equiv\; R \subseteq X \;\wedge\; S \subseteq X \tag{70}$$

NB: recall the notions of **greatest lower bound** and **least upper bound**, respectively.

In summary

Type $B \longleftarrow A$ forms a lattice:

$$
\begin{tikzcd}
 & \top \arrow[d, dotted, no head] & & \text{“top”} \\
 & R \cup S \arrow[dl, dotted, no head] \arrow[dr, dotted, no head] & & \text{join, lub (“least upper bound”)} \\
R \arrow[dr, no head] & & S \arrow[dl, no head] & \\
 & R \cap S \arrow[d, dotted, no head] & & \text{meet, glb (“greatest lower bound”)} \\
 & \bot & & \text{“bottom”}
\end{tikzcd}
$$

Propositio de homine et capra et lvpo

Being at the same bank:

$$SameBank = \ker where$$

Risk of somebody eating somebody else:

$$CanEat = SameBank \cap Eats$$

Starving ensured by *Farmer*'s presence at the same bank,

$$CanEat \subseteq SameBank \cdot \underline{Farmer} \qquad (71)$$

unfolds into:

$$where \cdot \underbrace{(where^\circ \cdot where \cap Eats)}_{CanEat} \subseteq where \cdot \underline{Farmer}$$

Propositio de homine et capra et lvpo

In this version, the *starving* invariant (71) is depictable as a diagram:

$$\begin{tikzcd}
Being \arrow[d, "where"'] \arrow[dr, phantom, "\subseteq"] & Being \arrow[l, "CanEat"'] \arrow[d, "\underline{Farmer}"] \\
Bank & Being \arrow[l, "where"]
\end{tikzcd} \tag{72}$$

which "reads" in a nice way:

> *where* (somebody) *CanEat* (somebody else) (that's)
> *where* (the) *Farmer* (is).

Distributivity

As we will prove later, **composition** distributes over **union**

$$R \cdot (S \cup T) = (R \cdot S) \cup (R \cdot T) \quad (73)$$
$$(S \cup T) \cdot R = (S \cdot R) \cup (T \cdot R) \quad (74)$$

while distributivity over **intersection** is side-conditioned:

$$(S \cap Q) \cdot R = (S \cdot R) \cap (Q \cdot R) \Leftarrow \left\{ \begin{array}{c} Q \cdot \mathrm{img}\, R \subseteq Q \\ \vee \\ S \cdot \mathrm{img}\, R \subseteq S \end{array} \right. \quad (75)$$

$$R \cdot (Q \cap S) = (R \cdot Q) \cap (R \cdot S) \Leftarrow \left\{ \begin{array}{c} (\ker R) \cdot Q \subseteq Q \\ \vee \\ (\ker R) \cdot S \subseteq S \end{array} \right. \quad (76)$$

On relational equality

Although **relational equality** can be established in a pointwise fashion,

$$R = S \quad \equiv \quad \langle \forall\ b, a \ :: \ b\ R\ a \equiv b\ S\ a \rangle$$

we will prefer the **pointfree** style, in basically two variants:

- **Cyclic inclusion** ("ping-pong") rule:

$$R = S \quad \equiv \quad R \subseteq S \wedge S \subseteq R \tag{77}$$

- **Indirect equality** rules:

$$R = S \quad \equiv \quad \langle \forall\ X \ :: \ (X \subseteq R \equiv X \subseteq S) \rangle \tag{78}$$

$$\equiv \quad \langle \forall\ X \ :: \ (R \subseteq X \equiv S \subseteq X) \rangle \tag{79}$$

Example of indirect proof

$$
\begin{array}{ll}
& X \subseteq (R \cap S) \cap T \\
\equiv & \qquad \{ \cap\text{-universal (69)} \} \\
& X \subseteq (R \cap S) \wedge X \subseteq T \\
\equiv & \qquad \{ \cap\text{-universal (69)} \} \\
& (X \subseteq R \wedge X \subseteq S) \wedge X \subseteq T \\
\equiv & \qquad \{ \wedge \text{ is associative} \} \\
& X \subseteq R \wedge (X \subseteq S \wedge X \subseteq T) \\
\equiv & \qquad \{ \cap\text{-universal (69) twice} \} \\
& X \subseteq R \cap (S \cap T) \\
:: & \qquad \{ \text{indirection (78)} \} \\
& (R \cap S) \cap T = R \cap (S \cap T)
\end{array}
\tag{80}
$$

All (data structures) **in one** (PF notation)

Products

$$\begin{tikzcd} A & A \times B \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & B \\ & C \arrow[ul, "R"] \arrow[u, "\langle R,S \rangle"'] \arrow[ur, "S"'] & \end{tikzcd} \tag{81}$$

where $\pi_1(a,b) = a$, $\pi_2(a,b) = b$ and

ψ	$PF\ \psi$
$a\ R\ c \wedge b\ S\ c$	$(a,b)\langle R,S \rangle c$
$b\ R\ a \wedge d\ S\ c$	$(b,d)(R \times S)(a,c)$

(82)

Clearly: $R \times S = \langle R \cdot \pi_1, S \cdot \pi_2 \rangle$

Exercises

Exercise 34: Show that

$$(b,c)\langle R,S\rangle a \quad \equiv \quad b\ R\ a \wedge c\ S\ a$$

PF-transforms to

$$\langle R,S\rangle \quad = \quad \pi_1^\circ \cdot R \cap \pi_2^\circ \cdot S \tag{83}$$

□

Exercise 35: Infer universal property

$$\pi_1 \cdot X \subseteq R \ \wedge \ \pi_2 \cdot X \subseteq S \quad \equiv \quad X \subseteq \langle R,S\rangle \tag{84}$$

from (83) via indirect equality (78).

□

Exercises

Exercise 36: Unconditional distribution laws

$$
\begin{aligned}
(P \cap Q) \cdot S &= (P \cdot S) \cap (Q \cdot S) \\
R \cdot (P \cap Q) &= (R \cdot P) \cap (R \cdot Q)
\end{aligned}
$$

will hold provide one of R or S is simple and the other injective. Tell which (justifying).

$\square$

Exercise 37: Derive from

$$
\langle R, S \rangle^{\circ} \cdot \langle X, Y \rangle = (R^{\circ} \cdot X) \cap (S^{\circ} \cdot Y) \tag{85}
$$

the following properties:

$$
\ker \langle R, S \rangle = \ker R \cap \ker S \tag{86}
$$

$\ker \langle R, id \rangle$ is always injective, for whatever R

$\square$

Sums

Example (Haskell):

```
data X = Boo Bool | Err String
```

PF-transforms to

$$\begin{tikzcd} Bool \arrow[r, "i_1"] \arrow[dr, "Boo"'] & Bool + String \arrow[d, "{[Boo\ ,Err]}"] & String \arrow[l, "i_2"'] \arrow[dl, "Err"] \\ & X & \end{tikzcd} \tag{87}$$

where

$$[R\ ,S] \;=\; (R \cdot i_1^\circ) \cup (S \cdot i_2^\circ) \qquad \text{cf.} \qquad \begin{tikzcd} A \arrow[r, "i_1"] \arrow[dr, "R"'] & A + B \arrow[d, "{[R\ ,S]}"] & B \arrow[l, "i_2"'] \arrow[dl, "S"] \\ & C & \end{tikzcd}$$

Dually: $R + S = [i_1 \cdot R\ , i_2 \cdot S]$

Sums

From $[R\ ,S] \;=\; (R \cdot i_1^\circ) \cup (S \cdot i_2^\circ)$ above one easily infers, by indirect equality,

$$[R\ ,S] \subseteq X \quad\equiv\quad R \subseteq X \cdot i_1 \;\wedge\; S \subseteq X \cdot i_2$$

(check this).

It turns out that inclusion can be strengthened to equality, and therefore **relational coproducts** have exactly the same properties as functional ones, stemming from the universal property:

$$[R\ ,S] = X \quad\equiv\quad R = X \cdot i_1 \;\wedge\; S = X \cdot i_2 \tag{88}$$

Thus $[i_1\ ,i_2] = id$ — solve (88) for R and S when $X = id$, etc etc.

Sums

The property for sums (coproducts) corresponding to (85) for products is:

$$[R\ ,S] \cdot [T\ ,U]^\circ \quad = \quad (R \cdot T^\circ) \cup (S \cdot U^\circ) \tag{89}$$

Exercise 38: Show that:

$$\mathrm{img}\,[R\ ,S] \quad = \quad \mathrm{img}\,R \cup \mathrm{img}\,S \tag{90}$$

$$\mathrm{img}\,i_1 \cup \mathrm{img}\,i_2 \quad = \quad id \tag{91}$$

□

$+$ meets $\times$

Exercise 39: Start by proving the fusion law

$$\langle R, S\rangle \cdot f \;=\; \langle R \cdot f, S \cdot f\rangle \tag{92}$$

where f is a function. Then, relying on both (88) and (92) infer the **exchange law**,

$$[\langle R, S\rangle \;, \langle T, V\rangle] \;=\; \langle [R \;, T], [S \;, V]\rangle \tag{93}$$

holding for all relations as in diagram

$$\begin{tikzcd}
A \arrow[r, "i_1"] \arrow[d, "R"'] \arrow[drr, "S"'] & A + B & B \arrow[l, "i_2"'] \arrow[d, "V"] \arrow[dll, "T"] \\
C & C \times D \arrow[l, "\pi_1"] \arrow[r, "\pi_2"'] & D
\end{tikzcd}$$

$\square$

Last but not least: monotonicity

All relational combinators seen so far are $\subseteq$-monotonic, for instance:

$$
\begin{array}{rcl}
R \subseteq S & \Rightarrow & R^\circ \subseteq S^\circ \\
R \subseteq S \wedge U \subseteq V & \Rightarrow & R \cdot U \subseteq S \cdot V \\
R \subseteq S \wedge U \subseteq V & \Rightarrow & R \cap U \subseteq S \cap V \\
R \subseteq S \wedge U \subseteq V & \Rightarrow & R \cup U \subseteq S \cup V
\end{array}
$$

etc

Exercise 40: Prove the following rules of thumb:

- smaller than injective (simple) is injective (simple)
- larger than entire (surjective) is entire (surjective)

□

Exercises

Exercise 41: Check which of the following hold:

- If relations R and S are simple, then so is $R \cap S$
- If relations R and S are injective, then so is $R \cup S$
- If relations R and S are entire, then so is $R \cap S$

□

Exercise 42: Prove that relational composition preserves *all* relational classes in the taxonomy of (57).

□

Exercise 43: Show that the following condional fusion law holds:

$$\langle R, S \rangle \cdot T = \langle R \cdot T, S \cdot T \rangle \quad \Leftarrow \quad R \cdot (\mathrm{img}\ T) \subseteq R \vee S \cdot (\mathrm{img}\ T) \subseteq S$$

□

Exercises

Exercise 44: Recalling (58), prove that

$$swap \triangleq \langle \pi_2, \pi_1 \rangle \tag{94}$$

is a bijection. (Assume property $R \cap S^\circ = R^\circ \cap S^\circ$.)

□

Exercise 45: Let $\leq$ be a preorder and f be a function taking values on the carrier set of $\leq$.

1. Define the pointwise version of relation $\sqsubseteq \;\triangleq\; f^\circ \cdot \leq \cdot f$
2. Show that $\sqsubseteq$ is a preorder.
3. Show that $\sqsubseteq$ is not (in general) a total order even in the case $\leq$ is so.

□

Lexicographic orderings

Let $R \Rightarrow S$ be the relational operator

$$b(R \Rightarrow S)a \quad \equiv \quad (b\ R\ a) \Rightarrow (b\ S\ a) \tag{95}$$

It can be shown that universal property

$$R \cap X \subseteq Y \quad \equiv \quad X \subseteq (R \Rightarrow Y) \tag{96}$$

holds.

We define the **lexicographic chaining** of two relations R and S as follows:

$$R\,;S \quad \triangleq \quad R \cap (R^\circ \Rightarrow S) \tag{97}$$

Exercise

Exercise 46: Let students in a course have two numeric marks,

$$\mathbb{N} \xleftarrow{mark1} Student \xrightarrow{mark2} \mathbb{N}$$

and define the preorders:

$$\begin{array}{lcl} \leq_{mark1} & \triangleq & mark1^\circ \cdot \leq \cdot\, mark1 \\ \leq_{mark2} & \triangleq & mark2^\circ \cdot \leq \cdot\, mark2 \end{array}$$

Spell out in pointwise notation the meaning of lexicographic ordering

$$\leq_{mark1} ; \leq_{mark2}$$

□

Exercises

Exercise 47: From (96) infer:

$$\bot \Rightarrow R \;=\; \top \tag{98}$$

$$R \Rightarrow \top \;=\; \top \tag{99}$$

□

Exercise 48: Via indirect equality over (97) show that

$$\top \,;\, S \;=\; S \tag{100}$$

holds for any S and that, for R symmetric, we have:

$$R \,;\, R \;=\; R \tag{101}$$

□

Modeling exercise

Consider diagram

$$\begin{tikzcd}
ISBN \arrow[d, "M"'] \arrow[dr, phantom, "\supseteq"] & ISBN \times UID \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] \arrow[d, "R"'] \arrow[dr, phantom, "\subseteq"] & UID \arrow[d, "N"] \\
Title \times Publisher \arrow[r, "\top"'] & Date & Name \times Address \times Phone \arrow[l, "\top"']
\end{tikzcd} \tag{102}$$

where

- M — records books in a loan library, identified by $ISBN$;
- N — records library users (identified by user id's in UID);

(both simple) and

- R — records loan dates.

Modeling exercise

The two squares impose bounds on R:

- Non-existing books cannot be loaned (left square);
- Only known users can take books home (right square).

Exercise 49: Add variables to both squares in (102) so that the same conditions are expressed pointwise. Then show that the conjunction of the two squares means the same as assertion

$$R^{\circ} \quad \subseteq \quad \langle M^{\circ} \cdot \top, N^{\circ} \cdot \top \rangle \tag{103}$$

and draw this in a diagram.

□

Modeling exercise

Exercise 50: Consider implementing M, R and N as files in a relational database. Before that, think of operations on the database such as, for example, that which records new loans (K):

$$borrow(K,(M,R,N)) \quad \triangleq \quad (M, R \cup K, N) \tag{104}$$

It is easy to see that the pre-condition

$$\text{pre-}borrow(K,(M,R,N)) \quad \triangleq \quad R \cdot K^{\circ} \subseteq id$$

captures a necessary condition for maintaining (102) (why?) but not it is not enough. Calculate for a rectangle in (102) at your choice the corresponding clause to add to pre-*borrow*.

□

Annex

Rules of the PF-transform seen so far:

ϕ	$PF\ \phi$
$\langle \exists\ a\ ::\ b\ R\ a \wedge a\ S\ c \rangle$	$b(R \cdot S)c$
$\langle \forall\ a, b\ ::\ b\ R\ a \Rightarrow b\ S\ a \rangle$	$R \subseteq S$
$\langle \forall\ a\ ::\ a\ R\ a \rangle$	$id \subseteq R$
$(f\ b)\ R\ (g\ a)$	$b(f^\circ \cdot R \cdot g)a$
$b\ R\ a \wedge b\ S\ a$	$b\ (R \cap S)\ a$
$b\ R\ a \vee b\ S\ a$	$b\ (R \cup S)\ a$
$b\ R\ a \wedge c\ S\ a$	$(b, c)\langle R, S \rangle a$
$x = i_1\ a \wedge c\ R\ a\ \vee\ x = i_2\ b \wedge c\ S\ b$	$c[R\ , S]x$
$b\ R\ a \wedge d\ S\ c$	$(b, d)(R \times S)(a, c)$
TRUE	$b\ \top\ a$
FALSE	$b\ \bot\ a$

Annex

Conversion of *simplicity* ('left uniqueness') of functions,

$$\mathrm{img}\, f \quad \subseteq \quad id \tag{105}$$

— recall slide 30 — into pointwise notation (Eindhoven quantifier notation). We calculate:

$$\begin{array}{ll} & \mathrm{img}\, f \quad \subseteq \quad id \\ \equiv & \qquad \{ \ (43)\ \text{etc} \ \} \\ & \langle \forall\ b, a\ :\ b(f \cdot f^{\circ})a :\ b\ id\ a \rangle \\ \equiv & \qquad \{ \ \text{composition (41) ; converse (48) ; } id\ a = a \ \} \\ & \langle \forall\ b, a\ :\ \langle \exists\ c\ :\ b\ f\ c :\ a\ f\ c \rangle :\ b = a \rangle \end{array}$$

Annex

$\equiv$ { prepare for splitting (7) via nesting (11) }

$\langle \forall\, b, a \;:\; \text{TRUE} \wedge \langle \exists\, c \;:\; b\ f\ c :\; a\ f\ c \rangle :\; b = a \rangle$

$\equiv$ { nesting (11) }

$\langle \forall\, b \;:\; \text{TRUE} :\; \langle \forall\, a \;:\; \langle \exists\, c \;:\; b\ f\ c :\; a\ f\ c \rangle :\; b = a \rangle \rangle$

$\equiv$ { splitting (7) }

$\langle \forall\, b \;:\; \text{TRUE} :\; \langle \forall\, c \;:\; b\ f\ c :\; \langle \forall\, a \;:\; a\ f\ c :\; b = a \rangle \rangle \rangle$

$\equiv$ { (un)nesting (11) }

$\langle \forall\, b, c \;:\; b\ f\ c :\; \langle \forall\, a \;:\; a\ f\ c :\; b = a \rangle \rangle$

$\equiv$ { (un)nesting (11) }

$\langle \forall\, b, c, a \;:\; b\ f\ c \wedge a\ f\ c :\; b = a \rangle$

Annex — proofs by $\subseteq$-transitivity

Wanting to prove $R \subseteq S$, the following rules may help in doing so by relying on a "mid-point" M (analogy with interval arithmetics):

- Rule A: **lowering the upper side**

$$
\begin{array}{ll}
 & R \subseteq S \\
\Leftarrow & \quad \{ \ M \subseteq S \text{ is known ; transitivity of } \subseteq \ \} \\
 & R \subseteq M
\end{array}
$$

  and then proceed with $R \subseteq M$.

- Rule B: **raising the lower side**

$$
\begin{array}{ll}
 & R \subseteq S \\
\Leftarrow & \quad \{ \ R \subseteq M \text{ is known; transitivity of } \subseteq \ \} \\
 & M \subseteq S
\end{array}
$$

  and then proceed with $M \subseteq S$.

Example

Proof of shunting rule (59):

$$
\begin{array}{cl}
 & R \subseteq f^\circ \cdot S \\
\Leftarrow & \qquad \{ \;\; id \subseteq f^\circ \cdot f \;\; ; \text{ raising the lower-side } \} \\
 & f^\circ \cdot f \cdot R \subseteq f^\circ \cdot S \\
\Leftarrow & \qquad \{ \;\; \text{monotonicity of } (f^\circ \cdot) \;\; \} \\
 & f \cdot R \subseteq S \\
\Leftarrow & \qquad \{ \;\; f \cdot f^\circ \subseteq id \;\; ; \text{ lowering the upper-side } \} \\
 & f \cdot R \subseteq f \cdot f^\circ \cdot S \\
\Leftarrow & \qquad \{ \;\; \text{monotonicity of } (f \cdot) \;\; \} \\
 & R \subseteq f^\circ \cdot S
\end{array}
$$

Thus the equivalence in (59) is established by circular implication.