# An Introduction to Relational Formal Modelling

***DI/UM, 2002***

J.N. Oliveira

# Functions are not enough

Partiality (in Haskell):

```
Mpi> (split head tail)(tail [1])
(
Program error: {head []}

Mpi> 2/0

Program error: {primDivDouble 2.0 0.0}
```

Functions such as `tail`, `/`, `head` (and many others!) are **partial**

# Functions are not enough

VDM-SL notation:

```
vdm> p tl [ ]
l. 1, c. 4:
  Run-Time Error 77: The sequence was empty
vdm>  p 2/0
l. 1, c. 3:
  Run-Time Error 76: Division with zero
vdm>
```

Functions such as `tl`, `/`, `hd` (and many others!) are **partial**

# Functions are not enough

```
gets : set of nat -> nat * set of nat
gets(s) == let a in set s
           in mk_(a,s \ {a}) ;
```

is not only partial

```
vdm> p gets({})
/home/jno/work/x.vdm, l. 4, c. 25:
  Run-Time Error 53: The binding environment was empty
vdm>
```

but also **non-deterministic**:

$$gets\{a,b\} = \langle a, \{b\}\rangle \vee gets\{a,b\} = \langle b, \{a\}\rangle$$

# Specifications as “properties”

- Specification of square root:

$$(sqrt\ x)^2 = x$$

that is

$$sq \cdot sqrt = id$$

(= $sqrt$ has left inverse $sq$)

- Specification of $sort$:

$$l' = sort\, l \ \Leftrightarrow\ (IsOrdered\, l') \wedge IsPermutation(l',$$

# Relational approach

Need to model

- total/**partial** functions
- **non-determinism**
- **properties**, datatype **invariants** and loop-invariants
- orders and **inductive** structures
- vagueness or **under-specification**. . .

# Relational approach

Need to model

- total/**partial** functions
- **non-determinism**
- **properties**, datatype **invariants** and loop-invariants
- orders and **inductive** structures
- vagueness or **under-specification**. . .

$\Rightarrow$ adoption of binary relations, which have a long tradition in the. . .

# Pre/post specification style

```
Sort(l: seq of int) r: seq of int
post IsPermutation(r,l) and IsOrdered(r);

IsPermutation: seq of int * seq of int -> bool
IsPermutation(l1,l2) == ....................

IsOrdered: seq of int -> bool
IsOrdered(l) == ............
```

```
gets(s: set of nat) r: nat * set of nat
pre  card s > 0
post r.#1 in set s and r.#2 = s \ {r.#1} ;
```

# Pre/post specification layout

```
Spec(a: A) r: B
pre Precond(a)
post Postcond(r,a);
```

where

$$Precond \quad : \quad A \longrightarrow 2$$

$$Postcond \quad : \quad B \times A \longrightarrow 2$$

leads to the **binary relation** approach:

$$Postcond \in 2^{B \times A} \quad \Leftrightarrow \quad Postcond \subseteq B \times A$$

# From predicates to relations

- **Predicate logic** connectives such as eg. $\wedge$ are "overloaded" operators
- They can be regarded as models of a more structured logic — that of **binary relations**
- **Functions** generalize to **binary relations** in a very natural way.
- Predicates, sets, functions and relations can **all** be combined in a **single relational calculus**
- Usual infix notation, *e.g.* $a < b$, can be generalized to any relation $R$, *e.g.* $aRb$

# Sets / functions made relational

Strategy: identify every

- **function** $f : A \longrightarrow B$ with the binary relation relating $a$ and $b$ iff $b = f\ a$. So, $b\,f\,a$ literally means $b = f\ a$.
- binary **predicate** $A \times B \xrightarrow{p} bool$ with binary relation $[\![p]\!]$ such that $a[\![p]\!]b \equiv p(a, b)$.
- unary **predicate** $A \xrightarrow{q} bool$ with binary relation $[\![q]\!]$ such that $a[\![q]\!]b \equiv a = b \wedge (q\ a)$.
- **set** $S \subseteq A$ with $[\![\lambda a.a \in S]\!]$. So,

$$a[\![S]\!]b \equiv a = b \wedge a \in S$$

# Arrows “are” binary relations

- **“Type”** relations in a way consistent with functions: $B \xleftarrow{R} A$ wherever $bRa$ involves $b \in B$ and $a \in A$.
- From now on, an arrow

$$B \xleftarrow{R} A$$

means a **binary relation** from $A$ (source) to $B$ (target) and write $bRa$ to denote that pair $\langle b, a \rangle$ is in $R$.

# Relations as Arrows

- **Ordering** on relations:

$$R \subseteq S \;\equiv\; bRa \Rightarrow bSa$$

  $R \subseteq S$ means that $R$ is either **less defined** or **more deterministic** than $S$.

- Extend **composition** $f \cdot g$ to $R \cdot S$ in the obvious way

$$b(R \cdot S)c \equiv \exists a \in A.bRa \wedge aSc$$

- Introduce **converse** $R^\circ$

$$a(R^\circ)b \equiv bRa$$

# Relational Equality

Pointwise equality:

$$R = S \quad \equiv \quad (bRa \equiv bSa)$$

Pointfree equality:

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

$$R = S \quad \equiv \quad R \subseteq S \wedge S \subseteq R$$

- **Indirect equality** rules:

$$\begin{aligned} R = S \quad &\equiv \quad \forall X.(X \subseteq R \equiv X \subseteq S) \\ &\equiv \quad \forall X.(R \subseteq X \equiv S \subseteq X) \end{aligned}$$

# Basic relational combinators

Given $C \xleftarrow{S} B$ and $B \xleftarrow{R} A$

- **Composition** $S \cdot R$ is s.t.

$$c(S \cdot R)a$$

holds wherever there exists some $b \in B$ such that $cSb \wedge bRa$.

- **Converse** $A \xleftarrow{R^\circ} B$ of $B \xleftarrow{R} A$

$$a(R^\circ)b \equiv bRa$$

- **Meet** $R \cap S$ — recall set-theoretical intersection

# Basic Relation Calculus (I)

Composition is associative:

$$R \cdot (S \cdot T) = (R \cdot S) \cdot T$$

Identity

$$R \cdot id = id \cdot R = R$$

Empty relation

$$R \cdot \bot = \bot \cdot R = \bot$$

where $B \xleftarrow{\bot} A$ is the smallest relation of its type.

# Basic Relation Calculus (II)

Composition is monotonic:

$$\frac{\begin{array}{c} R \subseteq S \\ T \subseteq U \end{array}}{(R \cdot T) \subseteq (S \cdot U)}$$

Bottom and top relations:

$$\bot \subseteq \ R \ \subseteq \top$$

where $B \xleftarrow{\bot} A$ is the largest relation of its type.
Pointwise descriptions:

$$b \top a \equiv true \ \ , \ \ b \bot a \equiv false$$

# Converse

$^\circ$-universal

$$X^\circ \subseteq Y \;\equiv\; X \subseteq Y^\circ$$

$^\circ$-monotonicity:

$$R \subseteq S \equiv R^\circ \subseteq S^\circ$$

Then:

Involution : $(R^\circ)^\circ = R$

Contravariance : $(R \cdot S)^\circ = S^\circ \cdot R^\circ$

These can be proved from $^\circ$**-universal** by (elegant) indirect proofs (example in next slide):

# Indirect proof of involution

$$
\begin{array}{ll}
 & (R^\circ)^\circ \subseteq Y \\
\equiv & \quad \{\ \circ\text{-universal}\ \ X^\circ \subseteq Y \ \equiv\ X \subseteq Y^\circ\ \ \text{for}\ X := \\
 & R^\circ \subseteq Y^\circ \\
\equiv & \quad \{\ \circ\text{-monotonicity}\} \\
 & R \subseteq Y \\
:: & \quad \{\ \text{indirection}\} \\
 & (R^\circ)^\circ = R
\end{array}
$$

# Meet and converse

$\cap$-universal

$$X \subseteq (R \cap S) \;\equiv\; (X \subseteq R) \wedge (X \subseteq S)$$

Converse distributes over $\cap$ (proof in next slide):

$$(R \cap S)^\circ = R^\circ \cap S^\circ$$

# Another indirect proof

$$
\begin{array}{ll}
 & X \subseteq R^\circ \cap S^\circ \\
\equiv & \quad \{ \cap\text{-universal} \} \\
 & (X \subseteq R^\circ) \wedge (X \subseteq S^\circ) \\
\equiv & \quad \{ \text{monotonicity and involution} \} \\
 & (X^\circ \subseteq R) \wedge (X^\circ \subseteq S) \\
\equiv & \quad \{ \cap\text{-universal} \} \\
 & X^\circ \subseteq (R \cap S) \\
\equiv & \quad \{ \text{monotonicity and involution} \} \\
 & X \subseteq (R \cap S)^\circ \\
:: & \quad \{ \text{indirection} \} \\
 & R^\circ \cap S^\circ = (R \cap S)^\circ
\end{array}
$$

# Converses of functions

Function converses $f^\circ, g^\circ$ etc. always exist (as **relations**) enjoying the following property:

$$(f\ b)R(g\ a) \equiv b(f^\circ \cdot R \cdot g)a$$

which unfolds to

$$bR(g\ a) \equiv b(R \cdot g)a \qquad \text{(f := id)}$$
$$(f\ b)Ra \equiv b(f^\circ \cdot R)a \qquad \text{(g := id)}$$

# Pointwise vs pointfree notation

## Function

$$\begin{aligned} fac\,0 &= 1 \\ fac(n+1) &= (n+1) * fac\,n \end{aligned}$$

in pointfree notation:

$$fac \cdot [\underline{0}, suc] = * \cdot [suc, fac]$$

## Property

$$fac\,n = fac\,m \;\Rightarrow\; n = m$$

($\equiv$ $fac$ is injective) in pointfree notation?

# Properties in pointfree style (I)

$fac$ is injective:

$$fac\, n = fac\, m \;\Rightarrow\; n = m$$

$\equiv$ { identity function / relation }

$$(fac\, n)\; id\; (fac\, m) \;\Rightarrow\; n\; id\; m$$

$\equiv$ { rule $(f\ b)R(g\ a) \equiv b(f^{\circ} \cdot R \cdot g)a$ }

$$n(fac^{\circ} \cdot id \cdot fac)m \;\Rightarrow\; n\; id\; m$$

$\equiv$ { dropping variables $n$ and $m$ ; natural-$id$ }

$$fac^{\circ} \cdot fac \;\subseteq\; id$$

# Properties in pointfree style (II)

Example property of integer arithmetics:

$$\begin{array}{r|l} n & d \\ \hline r & q \end{array} \qquad d \times q \leq n \equiv q \leq n/d$$

$\equiv$ { using "Haskell **section** notation" }

$$(d\times)q \leq n \;\equiv\; q \leq n(/d)$$

$\equiv$ { rule $(f\ b)Ra \equiv b(f^\circ \cdot R)a$ }

$$q((d\times)^\circ \cdot \leq)n \;\equiv\; q(\leq \cdot(/d))n$$

$\equiv$ { pointwise equality }

$$(d\times)^\circ \cdot \leq \;=\; \leq \cdot(/d)$$

# Orders and their taxonomy (A)

An order (or endo-relation) $A \xleftarrow{R} A$ is

| | |
|---|---|
| reflexive: | iff $id_A \subseteq R$ |
| coreflexive: | iff $R \subseteq id_A$ |
| 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 $R \cup R^\circ = \top$ |

where $A \xleftarrow{\top} A$ is the largest relation of its type.

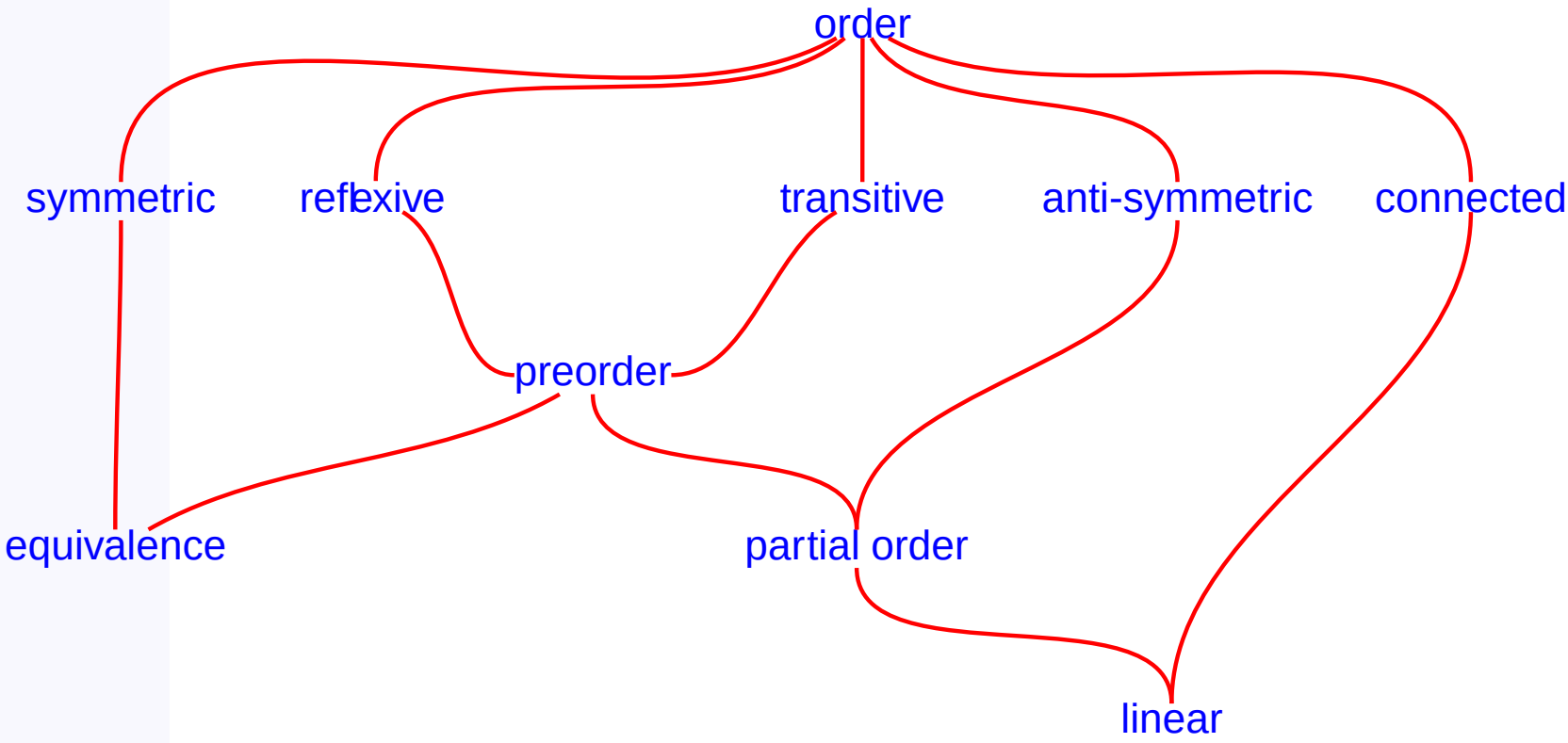
# Order taxonomy (B)

- **Preorders** are reflexive and transitive orders.
- **Partial** orders are anti-symmetric preorders
- **Linear** orders are connected partial orders
- **Equivalences** are symmetric preorders
- **Predicates** are coreflexive orders: the "meaning" of a predicate $Bool \xleftarrow{\phi} A$ is a coreflexive relation $[\![\phi]\!]$ such that

$$\phi\, a \;\equiv\; a[\![\phi]\!]a$$

mapping every $a$ which validates $\phi$ onto itself.

# Order taxonomy (C)



# Properties (A)

**Dedekind**'s rule (also known as the **modular law**):

$$(R \cdot S) \cap T \subseteq R \cdot (S \cap (R^\circ \cdot T))$$

Dually (apply converses and rename):

$$(R \cdot S) \cap T \subseteq (R \cap (T \cdot S^\circ)) \cdot S$$

Symmetrical equivalent statement:

$$(R \cdot S) \cap T \subseteq (R \cap (T \cdot S^\circ)) \cdot (S \cap (R^\circ \cdot T))$$

= "weak right-distribution of meet over composition".

# Derived combinators

- **Kernel** of $B \xleftarrow{R} A$ is $A \xleftarrow{\mathit{ker}\ R} A$ defined by

$$\mathit{ker}\ R \stackrel{\text{def}}{=} R^{\circ} \cdot R$$

- **Image** of $B \xleftarrow{R} A$ is $B \xleftarrow{\mathit{img}\ R} B$ defined by

$$\mathit{img}\ R \stackrel{\text{def}}{=} R \cdot R^{\circ}$$

- Duality:

$$\begin{aligned} \mathit{ker}\ (R^{\circ}) &= \mathit{img}\ R \\ \mathit{img}\ (R^{\circ}) &= \mathit{ker}\ R \end{aligned}$$

# Properties of kernel and image

Order-preservation:

$$R \subseteq S \;\Rightarrow\; \mathit{ker}\ R \subseteq \mathit{ker}\ S$$
$$R \subseteq S \;\Rightarrow\; \mathit{img}\ R \subseteq \mathit{img}\ S$$

Symmetry:

$$(\mathit{ker}\ R)^\circ \;=\; \mathit{ker}\ R$$
$$(\mathit{img}\ R)^\circ \;=\; \mathit{img}\ R$$

Also:

$$R \subseteq R \cdot \mathit{ker}\ R \qquad (= \mathit{img}\ R \cdot R)$$

# Entireness and simplicity

An **entire** (or total) relation is such that its kernel is reflexive:

$$R \text{ is entire } \equiv id \subseteq \mathit{ker}\ R$$

A **simple** (or functional) relation is such that its image is coreflexive:

$$R \text{ is simple } \equiv \mathit{img}\ R \subseteq id$$

Simplicity is the dual of entireness. Simple relations are also called **partial functions**.

# (Total) functions

**Functions** are both simple and entire relations, usually denoted by lowercase letters $f$:

$$\underbrace{id \subseteq f^\circ \cdot f}_{\text{entire}} \wedge \underbrace{f \cdot f^\circ \subseteq id}_{\text{simple}}$$

Thus:

$$f \subseteq R \Rightarrow R \text{ is entire}$$
$$R \subseteq f \Rightarrow R \text{ is simple}$$

In general, "larger than entire means entire" and "smaller than simple means simple"

# Surjectiveness and injectiveness

More taxonomy:

- $R$ is **surjective** iff $R^\circ$ is entire
- $R$ is **injective** iff $R^\circ$ is simple

Facts:

$$R \text{ is entire and injective} \equiv \mathit{ker}\ R = id$$
$$R \text{ is simple and surjective} \equiv \mathit{img}\ R = id$$

Summary:

| | **Reflexive** | **Coreflexive** |
|---|---|---|
| *ker* R | entire $R$ | injective $R$ |
| *img* R | surjective $R$ | simple $R$ |

# Bijections

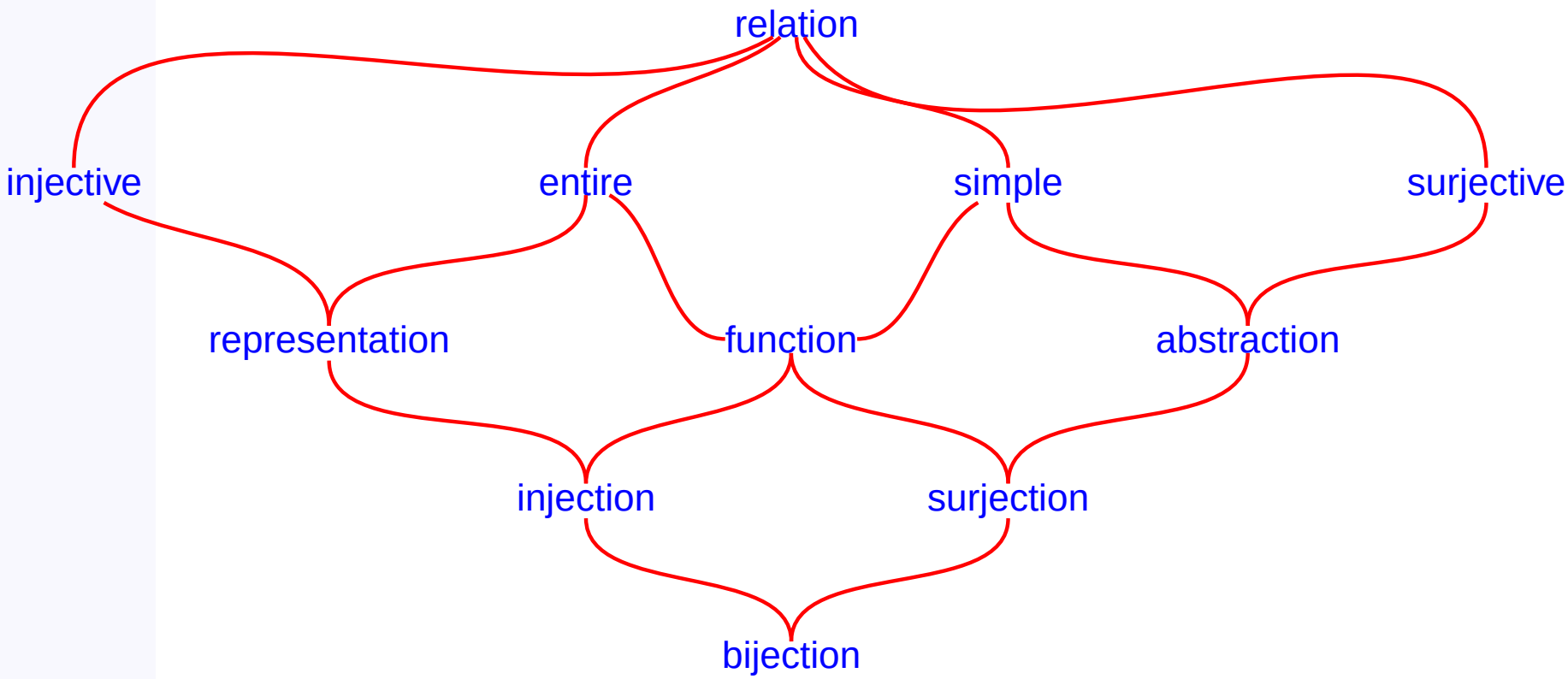
$f$ is **bijective** iff it is an injective and surjective function (thus simple and entire)

$$B \xleftarrow{f} A \text{ bijective} \equiv \mathit{ker}\ f = id \wedge \mathit{img}\ f = id$$

In this case

$$id = f^{\circ} \cdot f \ \wedge \ f \cdot f^{\circ} = id$$

# Binary relation taxonomy



# Reasoning about functions

## Shunting rules:

$$f \cdot R \subseteq S \equiv R \subseteq f^\circ \cdot S$$
$$R \cdot f^\circ \subseteq S \equiv R \subseteq S \cdot f$$

## Equality:

$$f \subseteq g \equiv f = g \equiv f \supseteq g$$

Ping-pong proof of the equality rule follows.

# Proof of functional equality

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

# Adding structure to the calculus

Note a recurrent **pattern** in several laws above:

$$\underbrace{X^\circ}_{f\ X} \subseteq Y \quad \equiv \quad X \subseteq \underbrace{Y^\circ}_{g\ Y}$$

$$\underbrace{(h\cdot)X}_{f\ X} \subseteq Y \quad \equiv \quad X \subseteq \underbrace{(h^\circ\cdot)Y}_{g\ Y}$$

$$\underbrace{X(\cdot h^\circ)}_{f\ X} \subseteq Y \quad \equiv \quad X \subseteq \underbrace{Y(\cdot h)}_{g\ Y}$$

as well as in

$$\underbrace{(d\times)q}_{f\ q} \leq n \quad \equiv \quad q \leq \underbrace{n(/d)}_{g\ n}$$

# Back to the primary school desk

The **integer division** algorithm

$$\begin{array}{r|l} 7 & 2 \\ \hline 1 & 3 \end{array} \qquad 2 \times 3 + 1 = 7 \quad \text{, "ie." } \quad 3 = 7/2$$

However

$$\begin{array}{r|l} 7 & 2 \\ \hline 3 & 2 \end{array} \qquad 2 \times 2 + 3 = 7 \quad \wedge \quad 2 \neq 7/2$$

$$\begin{array}{r|l} 7 & 2 \\ \hline 5 & 1 \end{array} \qquad 2 \times 1 + 5 = 7 \quad \wedge \quad 1 \neq 7/2$$

# Quotient is a supremum

$$\begin{array}{r|l} n & d \\ \hline r & q \end{array} \qquad d \times q + r = n \equiv q = n/d$$

provided $q$ is the largest such $q$ ($r$ is smallest)

$$\begin{aligned} n/d &= \bigvee \{q \mid \exists r \,.\, d \times q + r = n\} \\ &= \bigvee \{q \mid d \times q \leq n\} \end{aligned}$$

Maths teachers tell: it takes a while before children master the "$\bigvee$ semantics"!

What about you? Can you easily reason about $n/d$ in this format?

Try and prove $(n/m)/d = n/(m \times d)$.

# “Universal” property instead

Alternative:

$$\begin{array}{c|c} n & d \\ \hline r & q \end{array} \qquad d \times q \leq n \equiv q \leq n/d$$

“universal” property of integer division

Reasoning:

$$\begin{array}{ll} & q \leq (n/m)/d \\ \equiv & \quad \{ \text{ “universal” property } \} \\ & d \times q \leq n/m \end{array}$$

# Reasoning continued

$$
\begin{array}{ll}
\equiv & \{ \text{ “universal” property again } \} \\
& m \times (d \times q) \leq n \\
\equiv & \{ \text{ } \times \text{ is associative } \} \\
& (m \times d) \times q \leq n \\
\equiv & \{ \text{ “universal” property again } \} \\
& q \leq n/(m \times d)
\end{array}
$$

# Indirect equality

So we have

$$q \leq (n/m)/d \;\equiv\; q \leq n/(m \times d)$$

that is,

$$(n/m)/d \;=\; n/(m \times d)$$

by the **indirect equality** rule:

$$(q \leq x \equiv q \leq y) \equiv (x = y)$$

# Also easy to check

Cancellation law: $d \times (n/d) \leq n$

$\equiv$ { universal property }

$n/d \leq n/d$

$\equiv$ { reflexive $\leq$ }

$true$

"Reflection": $d \times 1 \leq n \equiv 1 \leq n/d$

$\equiv$ { $1$ is the unit of $\times$}

$d \leq n \equiv 1 \leq n/d$

# Galois connections

$n/d$ is a Galois connection:

$$\begin{array}{r|l} n & d \\ \hline r & q \end{array} \qquad \underbrace{d \times q}_{f\ q} \leq n \equiv q \leq \underbrace{n/d}_{g\ n}$$

In general, for **preorders** $(A, \leq)$ and $(B, \sqsubseteq)$ and

$$(A, \leq) \underset{f}{\overset{g}{\rightleftarrows}} (B, \sqsubseteq)$$

$(f, g)$ are Galois connected iff

# Galois adjoints

$$\underbrace{f}_{\text{lower adjoint}}\ b \leq a \ \equiv\ b \sqsubseteq \underbrace{g}_{\text{upper adjoint}}\ a$$

that is

$$f^{\circ} \cdot \leq \ = \ \sqsubseteq \cdot g$$

Remarks:

- Galois (connected) adjoints enjoy a number of interesting generic properties
- Very elegant —**calculational** —way of performing inequational reasoning (including logical deduction)

# Basic properties

Cancellation:

$$(f \cdot g)a \leq a \quad \text{and} \quad b \sqsubseteq (g \cdot f)b$$

Distribution (in case of lattice structures):

$$\begin{aligned} f(a \sqcup a') &= (f\ a) \vee (f\ a') \\ g(b \wedge b') &= (g\ b) \sqcap (g\ b') \end{aligned}$$

Conversely,

- If $f$ distributes over $\sqcup$ then it has an upper adjoint $g$ ($f^{\#}$)
- If $g$ distributes over $\wedge$ then it has a lower adjoint $f$ ($g^{\flat}$)

# Other properties

If $(f, g)$ are Galois connected,

- $f$ ($g$) **uniquely** determines $g$ ($f$) — thus the $\_^\flat$, $\_^\sharp$ notations
- $f$ and $g$ are **monotonic**
- $(g, f)$ are also Galois connected — **reverse** the orderings
- $f = f \cdot g \cdot f$ and $g = g \cdot f \cdot g$

etc

# Summary

| $(f\ b) \leq a \equiv b \sqsubseteq (g\ a)$ | | |
|---|---|---|
| **Description** | $f = g^{\flat}$ | $g = f^{\sharp}$ |
| Defi nition | $f\ b = \bigwedge\{a \mid b \sqsubseteq g\ a\}$ | $g\ a = \bigsqcup\{b \mid f\ b \leq a\}$ |
| Cancellation | $f(g\ a) \leq a$ | $b \sqsubseteq g(f\ a)$ |
| Distribution | $f(b \sqcup b') = (f\ b) \vee (f\ b')$ | $g(a' \sqcap a) = (g\ a') \sqcap (g\ a$ |
| Monotonicity | $b \sqsubseteq b' \Rightarrow f\ b \leq f\ b'$ | $a \leq a' \Rightarrow g\ a \sqsubseteq g\ a'$ |

# Converse

| $(f\ X) \subseteq Y \equiv X \subseteq (g\ Y)$ | | | |
|---|---|---|---|
| **Description** | $f = g^\flat$ | $g = f^\sharp$ | **Obs.** |
| converse | $(\_)^\circ$ | $(\_)^\circ$ | $bR^\circ a \equiv aRb$ |

Thus:

**Cancellation** $(R^\circ)^\circ = R$

**Monotonicity** $R \subseteq S \equiv R^\circ \subseteq S^\circ$

**Distributions** $(R \cap S)^\circ = R^\circ \cap S^\circ, (R \cup S)^\circ = R^\circ \cup$

# Functions

| $(f\ X) \subseteq Y \equiv X \subseteq (g\ Y)$ | | | |
|---|---|---|---|
| **Description** | $f = g^\flat$ | $g = f^\sharp$ | **Obs.** |
| shunting rule | $(h\cdot)$ | $(h^\circ\cdot)$ | NB: $h$ is a function |
| "converse" shunting rule | $(\cdot h^\circ)$ | $(\cdot h)$ | NB: $h$ is a function |

Consequences:

Functional equality: $h \subseteq g \equiv \quad h = k \quad \equiv h \supseteq k$

Functional division: $h^\circ \cdot R = h \setminus R$

**Question:** what does $h \setminus R$ mean?

# Relational division

| $(f\ X) \subseteq Y \equiv X \subseteq (g\ Y)$ | | | |
|---|---|---|---|
| **Description** | $f = g^\flat$ | $g = f^\sharp$ | **Obs.** |
| right-division | $(R\cdot)$ | $(R \setminus\ )$ | right-factor |
| left-division | $(\cdot R)$ | $(\ /\ R)$ | left-factor |

Immediate: $(R\cdot)$ and $(\cdot R)$ distribute over union:

$$\begin{aligned} R \cdot (S \cup T) &= (R \cdot S) \cup (R \cdot T) \\ (S \cup T) \cdot R &= (S \cdot R) \cup (T \cdot R) \end{aligned}$$

Some intuition about relational division operators follows.

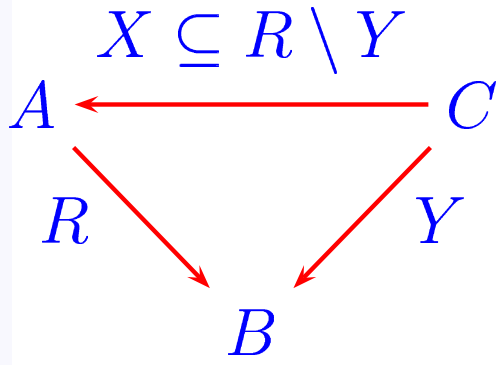
# Relational division

The **relational division** operators are upper-adjoints:

$$R \cdot X \subseteq Y \equiv X \subseteq R \setminus Y$$

$$X \cdot R \subseteq Y \equiv X \subseteq Y / X$$

**Right division** abstracts a (pointwise) universal quantifi cation

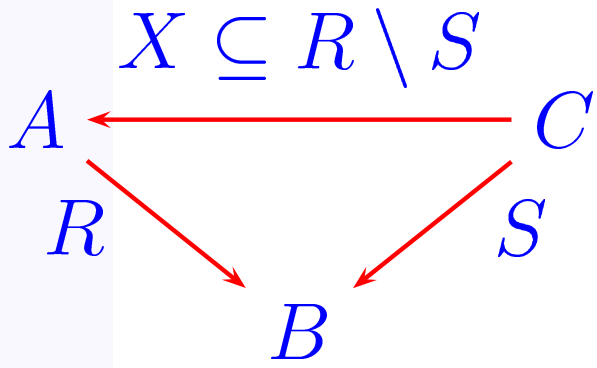



$$a(R \setminus Y)c \equiv (\forall b.bRa \Rightarrow bY\,c)$$

An example follows.

# Example

Recall data division in the **relational model**:




$$a(R \setminus S)c \equiv (\forall b.bRa \Rightarrow bSc)$$

$b\ R\ a$ = flight $b$ carries passenger $a$

$b\ S\ c$ = flight $b$ belongs to air-company $c$

$a\ (R \setminus S)\ c$ = passenger $a$ is faithful to company $c$, that is, (s)he only flies company $c$.

# Left division

By taking converses we arrive at $S \,/\, R = (R^\circ \setminus S^\circ)^\circ$:

$$
\begin{array}{lcl}
X \subseteq S \,/\, R & \equiv & X \subseteq (R^\circ \setminus S^\circ)^\circ \\
& \equiv & \quad \{ \text{converses and } (R^\circ\backslash)^\flat \} \\
& & R^\circ \cdot X^\circ \subseteq S^\circ \\
& \equiv & \quad \{ \text{converses} \} \\
& & X \cdot R \subseteq S
\end{array}
$$

ie. Galois connection

$$X \cdot R \subseteq S \;\equiv\; X \subseteq S \,/\, R$$

# Meet

## $\cap$-universal

$$X \subseteq (R \cap S) \;\equiv\; (X \subseteq R) \wedge (X \subseteq S)$$

is a Galois connection

$$(\Delta, \cap)$$

where $\Delta\ X = (X, X)$, cf.

$$(X, X)(\subseteq \times \subseteq)(R, S) \;\equiv\; X \subseteq \cap(R, S)$$

So $\cap = \Delta^\sharp$ distributes over itself, etc

# Properties of $\cap$

From $\cap$-universal infer:

- **$\cap$-cancellation** ($X := R \cap S$)

$$R \cap S \subseteq R \ \wedge \ R \cap S \subseteq S$$

- **$\cap$-abbreviation** ($X := R$)

$$R \subseteq S \ \equiv \ R = R \cap S$$

- **$\cap$-idempotency** ($S := R$)

$$R \cap R \ = \ R$$

# More properties of $\cap$

$\cap$ is **commutative**:

$$R \cap S \;=\; S \cap R$$

$\cap$ is **associative**:

$$R \cap (S \cap T) \;=\; (R \cap S) \cap T$$

**$\cap$-fusion**:

$$\begin{array}{rcl} T \cdot (R \cap S) & \subseteq & (T \cdot R) \cap (T \cdot S) \\ (R \cap S) \cdot T & \subseteq & (R \cdot T) \cap (S \cdot T) \end{array}$$

# Meet and join

| $(f\ X) \leq Y \equiv X \sqsubseteq (g\ Y)$ | | | |
|---|---|---|---|
| **Description** | $f = g^\flat$ | $g = f^\sharp$ | **Obs.** |
| meet | $\Delta$ | $\cap$ | $\leq$ is $(\subseteq \times \subseteq)$ |
| join | $\cup$ | $\Delta$ | $\sqsubseteq$ is $(\subseteq \times \subseteq)$ |

Join:

$$\cup(R, S) \subseteq Y \equiv (R, S)(\subseteq \times \subseteq)(Y, Y)$$

that is,

$$R \cup S \subseteq Y \equiv R \subseteq Y \wedge S \subseteq Y$$

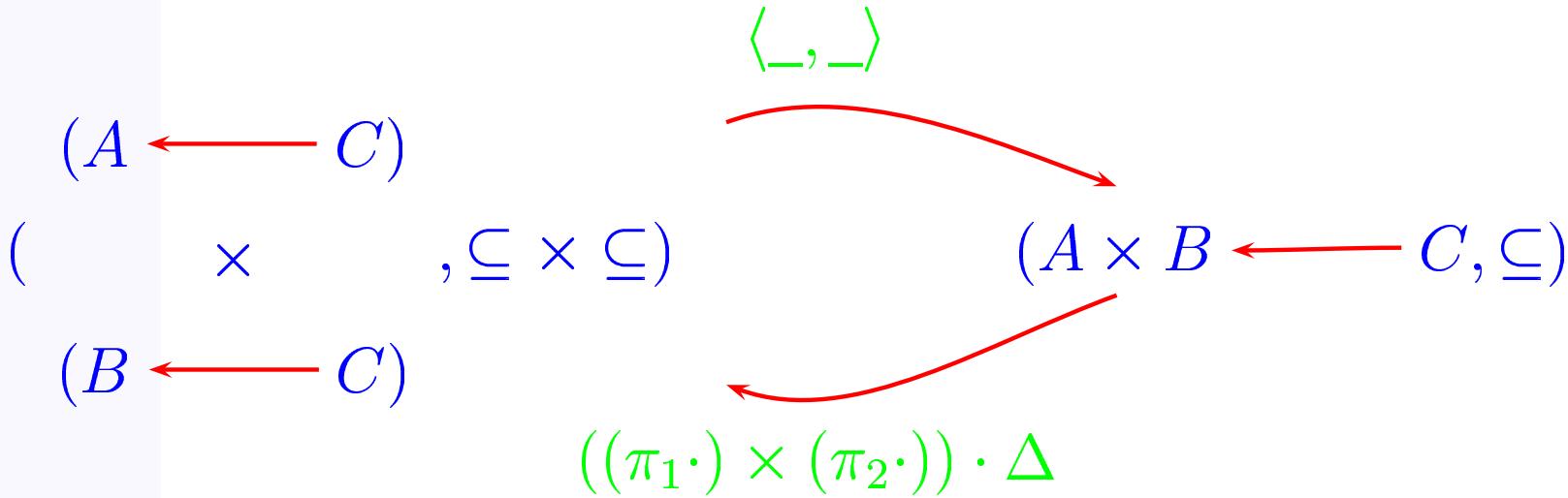
# Relational split

Functions:

$$x = \langle f, g \rangle \quad \equiv \quad \pi_1 \cdot x = f \quad \wedge \quad \pi_2 \cdot x = g$$

Relations:

$$X \subseteq \langle R, S \rangle \quad \equiv \quad \pi_1 \cdot X \subseteq R \quad \wedge \quad \pi_2 \cdot X \subseteq S$$



# Properties

$\langle \_, \_ \rangle$ is an **upper-adjoint**, so it distributes over meet

$$\begin{aligned} \langle R, S \cap T \rangle &= \langle R, S \rangle \cap \langle R, T \rangle \\ \langle S \cap T, R \rangle &= \langle S, R \rangle \cap \langle T, R \rangle \end{aligned}$$

etc. Moreover:

$$\langle R, S \rangle = (\pi_1^\circ \cdot R) \cap (\pi_2^\circ \cdot S) \qquad \text{(-98)}$$

Why? Again Galois at work:

# Calculation

$$
\begin{array}{lcl}
X \subseteq \langle R, S\rangle & \equiv & \pi_1 \cdot X \subseteq R \wedge \pi_2 \cdot X \subseteq S \\
& \equiv & \quad \{ \text{ Galois connected } ((f\cdot), (f^\circ\cdot))\} \\
& & X \subseteq \pi_1^\circ \cdot R \wedge X \subseteq \pi_2^\circ \cdot S \\
& \equiv & \quad \{ \text{ Galois connected } (\cap^\flat, \cap)\} \\
& & X \subseteq (\pi_1^\circ \cdot R \cap \pi_2^\circ \cdot S) \\
& :: & \quad \{ \text{ indirect equality}\} \\
& & \langle R, S\rangle = \pi_1^\circ \cdot R \cap \pi_2^\circ \cdot S
\end{array}
$$

# Pointwise $\langle R, S \rangle$

$$
\begin{array}{lcl}
(a,b)\langle R,S\rangle c & \equiv & (a,b)(\pi_1^\circ \cdot R \cap \pi_2^\circ \cdot S)c \\
& \equiv & \quad \{ \text{ pointwise } \cap \} \\
& & (a,b)(\pi_1^\circ \cdot R)c \wedge (a,b)(\pi_2^\circ \cdot S)c \\
& = & \quad \{ \text{ rule } (f\ b)Ra \equiv b(f^\circ \cdot R)a \ \} \\
& & \pi_1(a,b)Rc \wedge \pi_2(a,b)Sc \\
& = & \quad \{ \text{ projections} \} \\
& & aRc \wedge bSc
\end{array}
$$

# Relational either

Functions:

$$[f, g] = x \quad \equiv \quad f = x \cdot i_1 \ \wedge g = x \cdot i_2$$

Relations:

$$[R, S] \subseteq X \quad \equiv \quad R \subseteq X \cdot i_1 \ \wedge S \subseteq X \cdot i_2 \qquad \text{(-102)}$$

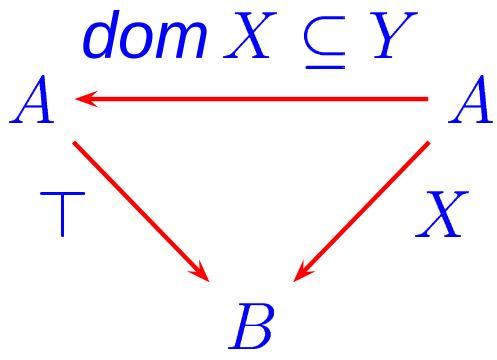
Thus $[\_, \_]$ is a **lower-adjoint**, it distributes over $\cup$, etc.
Moreover,

$$[R, S] \quad = \quad (R \cdot i_1^\circ) \cup (S \cdot i_2^\circ) \qquad \text{(-103)}$$

# Domain and range

| $(f\ X) \subseteq Y \equiv X \subseteq (g\ Y)$ | | | |
|---|---|---|---|
| **Description** | $f = g^\flat$ | $g = f^\sharp$ | **Obs.** |
| domain | *dom* | $(\top\cdot)$ | lower $\subseteq$ restricted to coreflexives |
| range | *rng* | $(\cdot\top)$ | lower $\subseteq$ restricted to coreflexives |

cf.




$$dom\ X \subseteq Y \equiv X \subseteq \top \cdot Y$$

# Domain and range

Dualization:

$$\mathit{dom}\, R = \mathit{rng}\, R^{\circ}$$

Explicit definitions:

$$\begin{aligned} \mathit{rng}\, R &= \mathit{img}\, R \cap id \\ \mathit{dom}\, R &= \mathit{img}\, R^{\circ} \cap id = \mathit{ker}\, R \cap id \end{aligned}$$

Facts:

$$\begin{aligned} R &= R \cdot (\mathit{dom}\, R) \\ R &= (\mathit{rng}\, R) \cdot R \end{aligned}$$

# Domain and split

The following fact holds:

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

Corollary:

$$\mathit{dom}\, R \;=\; \mathit{ker}\, \langle id, R\rangle$$

Another consequence of the fact above:

$$\mathit{ker}\, R \subseteq \mathit{ker}\, (S \cdot R) \;\Leftarrow\; S \text{ entire}$$

Corollary:

$$\mathit{ker}\, R \;\subseteq\; \mathit{ker}\, (f \cdot R)$$

# Comprehending relations

For each $B \xleftarrow{R} A$ define its **graph** or comprehension by

$$\mathcal{G}\ R = \{(b, a) \mid bRa\}$$

Clearly, $R = [\![\mathcal{G}\ R]\!]$ and so we often abbreviate $\mathcal{G}\ \mathcal{R}$ to $R$.
The graph of every **coreflexive** $S$ is made simpler for obvious reasons:

$$\mathcal{G}\ S = \{a \mid aSa\}$$

# Finite relations

$R$ is said to be finite wherever $\mathcal{G}\ \mathcal{R}$ is a finite set.

- Finite relations, which can be enumerated, browsed and stored in a computer, are the subject of **relational database** design.
- Every finite, **simple** relation expresses a **functional dependency**.
- The graphs of finite and simple relations are called **mappings** in VDM-SL terminology.
- We will use greek identifiers ($\sigma, \tau$ etc) to denote (finite) mappings

# VDM-SL mapping notation

- Datatype: `map` $A$ `to` $B$
- Pointwise VDM-SL concrete syntax

$$\{a \mapsto b \mid b\ \sigma\ a\}$$

  replaces $\{(b, a) \mid b\ \sigma\ a\}$.
- In VDM-SL notation, $b\ \sigma\ a$ is furthermore rephrased as $a \in \mathit{dom}\ \sigma \wedge b = \sigma(a)$ — cf. $\sigma = \sigma \cdot \mathit{dom}\ \sigma$ — that is, we have

$$\sigma \ = \ \{a \mapsto \sigma(a) \mid a \in \mathit{dom}\ \sigma\}$$

# Meaning of VDM-SL specs

```
Spec(a: A) r: B
pre precond(a)
post postcond(r,a);
```

where

$$bool \xleftarrow{precond} A \quad , \quad bool \xleftarrow{postcond} B \times A$$

means $B \xleftarrow{Spec} A$ where

$$Spec \stackrel{\text{def}}{=} [\![postcond]\!] \cdot [\![precond]\!]$$

# VDM-SL $Sqrt$ spec

```
Sqrt(x: real) r: real
pre true
post sq(r) = x ;
```

means

$$Sqrt = [\![\lambda(r,x).sq\ r = x]\!]$$

$\equiv$ { meaning of a binary predicate }

$$r\ Sqrt\ x \equiv (sq\ r)\ id\ x$$

$\equiv$ { converse of a function; natural-$id$ }

$$Sqrt = sq^{\circ}$$

# Turning implicit specifications...

Sorting in VDM-SL notation:

```
Sort(l: seq of int) r: seq of int
post IsOrdered(r) and IsPermutation(r,l);
```

where

```
IsPermutation: seq of int * seq of int -> bool
IsPermutation(l1,l2) ==
  forall e in set (elems l1 union elems l2) &
    card {i | i in set inds l1 & l1(i) = e} =
    card {i | i in set inds l2 & l2(i) = e};
```

# . . . into relational models

. . . abbreviates to

$$Sort \stackrel{\text{def}}{=} [\![IsOrdered]\!] \cdot IsPermutation$$

# . . . into relational models

. . . abbreviates to

$$Sort \stackrel{\text{def}}{=} [\![IsOrdered]\!] \cdot (\ker\ seq2bag)$$

assuming

```
seq2bag: seq of int -> map int to nat1
seq2bag(l) ==
  { e |-> card { i | i in set inds l & l(i) = e } |
      e in set elems l };
```

# . . . into relational models

. . . abbreviates to

$$Sort \stackrel{\text{def}}{=} [\![IsOrdered]\!] \cdot (\ker\ seq2bag)$$

assuming

```
seq2bag: seq of int -> map int to nat1
seq2bag(l) ==
  { e |-> card { i | i in set inds l & l(i) = e } |
      e in set elems l };
```

`IsPermutation` is an **equivalence** because *ker* $f$ always is reflexive, symmetric and transitive.

# Relational semantics of VDM-SL

From the VDM-SL on-line manual:

| Operator | Name | Semantics description |
|---|---|---|
| `s <: m` | Domain restrict to | Creates the map consisting of the elements in `m` whose key is in `s`. `s` need not be a subset of `dom m`. |

Formal **semantics**:

$$[\![s \texttt{<:}\ m]\!] \ = \ [\![m]\!] \cdot [\![s]\!]$$

where $[\![s]\!]$ is correflexive and $[\![m]\!]$ is simple.

# Relational semantics of VDM-SL

From the VDM-SL on-line manual:

| Operator | Name | Semantics description |
|---|---|---|
| `m1 ++ m2` | Override | overrides and merges `m1` with `m2`, i.e. it<br>like a merge except that `m1` and `m2` nee<br>not be compatible; any common el<br>ments are as by `m2` (so `m2` overrides `m1` |

Formal **semantics**:

$$[\![m_1 \texttt{++} m_2]\!] \;=\; [\![m_2]\!] \rightarrow [\![m_2]\!] \;,\; [\![m_1]\!]$$

cf. relational **McCarthy** conditional:

# Relational McCarthy conditional

It is defi ned by

$$R \to S \ , \ T \ \stackrel{\text{def}}{=} \ (S \cdot \mathsf{dom}\, R) \cup T \cdot (id - \mathsf{dom}\, R)$$

where

| $(f\ X) \subseteq Y \equiv X \subseteq (g\ Y)$ | | | |
|---|---|---|---|
| **Description** | $f = g^\flat$ | $g = f^\sharp$ | **Obs.** |
| difference | $(\_ - R)$ | $(R \cup\ )$ | |

that is,

$$\begin{aligned} X - R \subseteq Y \ &\equiv \ X \subseteq R \cup Y \\ X - R \ &= \ \bigcap \{Y \mid X \subseteq R \cup Y\} \end{aligned}$$

# Reasoning about VDM-SL

We want to prove VDM-SL properties such as

$$\begin{aligned} X \mathbin{<:} (Y \mathbin{<:} \sigma) &= (X \cap Y) \mathbin{<:} \sigma \\ \{\} \mathbin{<:} \sigma &= \{\mapsto\} \\ X \mathbin{<:} (\sigma_1 \mathbin{++} \sigma_2) &= (X \mathbin{<:} \sigma_1) \mathbin{++} (X \mathbin{<:} \sigma_2) \end{aligned}$$

First, some properties of coreflexives:

- Coreflexives are **symmetric** and **transitive**:

$$R = R^{\circ} = R \cdot R = R \cap id$$

- **Meet** of two coreflexives is composition:

$$R \cap S = R \cdot S$$

# Example of proof

$$
\begin{aligned}
& [\![X <: (Y <: \sigma)]\!] \\
= & \quad \{ \text{ relational meaning of } <: \} \\
& [\![Y <: \sigma]\!] \cdot [\![X]\!] \\
= & \quad \{ \text{ relational meaning of } <: \} \\
& ([\![\sigma]\!] \cdot [\![Y]\!]) \cdot [\![X]\!] \\
= & \quad \{ \text{ associativity of } \cdot \text{ and coreflexives} \} \\
& [\![\sigma]\!] \cdot ([\![X]\!] \cdot [\![Y]\!]) \\
= & \quad \{ \text{ meet of two coreflexives is composition } \} \\
& [\![\sigma]\!] \cdot ([\![X]\!] \cap [\![Y]\!])
\end{aligned}
$$

# Proof continued

$$
\begin{aligned}
& [\![\sigma]\!] \cdot ([\![X]\!] \cap [\![Y]\!]) \\
= & \quad \{ \text{ meaning of set intersection } \} \\
& [\![\sigma]\!] \cdot [\![X \cap Y]\!] \\
= & \quad \{ \text{ relational meaning of } \texttt{<:} \} \\
& [\![(X \cap Y) \texttt{<:} \sigma]\!]
\end{aligned}
$$

# Another proof

$$
\begin{array}{ll}
 & [\![X <: (\sigma_1 \mathbin{++} \sigma_2)]\!] \\
= & \quad \{ \text{ relational meaning of } <: \text{ and } ++ \} \\
 & (([\![\sigma_2]\!] \to [\![\sigma_2]\!]) , [\![\sigma_1]\!]) \cdot [\![X]\!] \\
= & \quad \{ \text{ McCarthy fusion law} \} \\
 & [\![\sigma_2]\!] \cdot [\![X]\!] \to [\![\sigma_2]\!] \cdot [\![X]\!] , [\![\sigma_1]\!] \cdot [\![X]\!] \\
= & \quad \{ \text{ relational meaning of } <: \} \\
 & [\![X <: \sigma_2]\!] \to [\![X <: \sigma_2]\!] , [\![X <: \sigma_1]\!] \\
= & \quad \{ \text{ relational meaning of } ++ \} \\
 & [\![(X <: \sigma_1) \mathbin{++} (X <: \sigma_2)]\!]
\end{array}
$$

# Etc.

Home work: define the relational semantics of *e.g.*.

| Operator | Name | Semantics description |
|---|---|---|
| `m <-: s` | Domain restricted by | Creates the map consisting of the elements in `m` whose key is not in `s`. `s` need not be a subset of `dom m`. |

and prove similar properties.

# Override pointwise

Since

$$dom\,(\sigma_1 \mathbin{++} \sigma_2) = dom\,\sigma_1 \cup dom\,\sigma_2$$

we have, after expansion of the relational definition:

```
s1 ++ s2 ==
   { k |-> if k in set dom s2
           then s2(k)
           else s1(k)
     | k in set dom s1 union dom s2 }
```

The above proof over this definition would have been far less compact.

# Inductive override

Another version of map override:

```
s1 ++ s2 ==
   if s1 = {|->}
   then s2
   else let k in set dom s1
        in  { k |-> if k in set dom s2
              then s2(k)
              else s1(k) } munion { k } <-: s1 ++
```

How do we arrive at this recursive scheme?

See next set of slides.