Pointfree Factorization of Operation Refinement

J.N. Oliveira & C.J. Rodrigues

Dept. Informática,
Universidade do Minho
Braga, Portugal

FM'06 — 14th Int. Symp. on Formal Methods
McMaster University, Hamilton, Ontario Canada
August 21 - 27, 2006

Context

PURe

- Program **understanding** by **reverse** engineering
- Software architecture "fission"
- Systems and components as **coalgebras**

Understanding

Analysing, **factoring** (splitting, slicing) , (converse of) **refining**

Are we ready for this

- Are our maths up-to-date for all this?
- Go back to basics?

About the title – refinement

Refinement

- {S,SP,SC}-refinement
- $T_\bullet$-refinement
- $W_\bullet$-refinement
- *downward*, *upward* refinement
- *forwards*, *backwards* refinement
- ...

Wikipedia

Operation refinement — *converts a specification of an operation on a system into an implementable program (e.g., a procedure). The* **postcondition** *can be strengthened and/or the* **precondition** *weakened in this process.*

About the title — factorization

Expressing (numbers, expressions, etc) as products of factors

School example

Rather than "brute force" arithmetic calculations, eg.

$$\frac{756}{792} = 0.9545454\ldots$$

use prime factorization

$$\begin{aligned} \frac{756}{792} &= \frac{2^2 \times 3^3 \times 7}{2^3 \times 3^2 \times 11} \\ &= 2^{-1} \times 3 \times 7 \times 11^{-1} \\ &= \frac{21}{22} \end{aligned}$$

In general, **factorization** identifies "basic building blocks" so that facts about the whole can be inferred from facts about its blocks.

About the title

Pointfree

What ??

Operation refinement

Suppose s and r are software components described by (set-valued) state transition functions

Total correctness

Component $\mathcal{P}A \xleftarrow{r} A$ **refines** (implements, reifies) component $\mathcal{P}A \xleftarrow{s} A$ — written $s \vdash r$ — iff

$$s \vdash r \stackrel{\mathrm{def}}{=} \langle \forall\ a\ :\ \emptyset \subset (s\ a) :\ \emptyset \subset (r\ a) \subseteq (s\ a) \rangle \qquad (1)$$

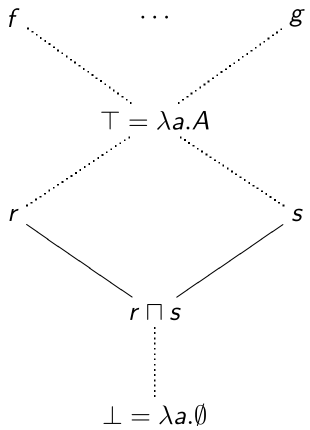
where $s\ a$ means the set of states reachable (in machine s) from state a.

Comments:

- Consensual and conceptually simple
- Copes with model undefinedness and vagueness

However...

Funny shaped semi-lattice:



$\langle \forall\ a\ ::\ \#(f\ a) = \#(g\ a) = 1 \rangle$

"chaos"

glb= "largest common spec"

"sink"

$$r \sqcap s \quad = \quad \mathit{lambdaa.(if}\ \ (r\ \ a) = \emptyset \vee (s\ \ a) = \emptyset\ \ \mathit{then}\ \emptyset$$
$$\mathit{else}\ (r\ \ a) \cup (s\ \ a))$$

Can we break the complexity of $\vdash$?

Yes, following a plan in two steps:

Change of "math space"

Express and reason about $\vdash$ with "less symbols" and "more agile" rules — thus the *pointfree transform*.

Factorization

Factor $\vdash$ in simpler "building blocks" — eg. by dissociating decrease of *nondeterminism* from increase of *definition*.

Can $\vdash$ be factored in (simpler) "building blocks"?

Yes:

Groves factorization

It has been noted by Lindsay Groves (and others) that

$$\begin{array}{rcl} s \vdash r & \equiv & \langle \exists\ t\ ::\ s \vdash_{pre} t \wedge t \vdash_{post} r \rangle \\ & \equiv & \langle \exists\ t'\ ::\ s \vdash_{post} t' \wedge t' \vdash_{pre} r \rangle \end{array}$$

where

- $s \vdash_{pre} t$ — t only weakens the precondition of s
- $t \vdash_{post} r$ — r only strengthens the post-condition of t

Question:

- In what sense are $\vdash_{pre} / \vdash_{post}$ factors of $\vdash$?
- What can we expect from such factoring?

Need for something else...

About changing "math space"

Another school maths example:

The problem

Find three consecutive integers which together add up 120

The model

$$x + (x + 1) + (x + 2) = 120$$

The calculation

$$
\begin{array}{ll}
 & 3x + 3 = 120 \\
\equiv & \qquad \{ \text{ "al-djabr" rule } \} \\
 & 3x = 120 - 3 \\
\equiv & \qquad \{ \text{ "al-hatt" rule } \} \\
 & x = 40 - 1
\end{array}
$$

School maths example

The solution

$$\begin{array}{rcl} x & = & 39 \\ x+1 & = & 40 \\ x+2 & = & 41 \end{array}$$

The calculus

"al-djabr" rule:

$$x - z \leq y \quad \equiv \quad x \leq y + z$$

"al-hatt" rule:

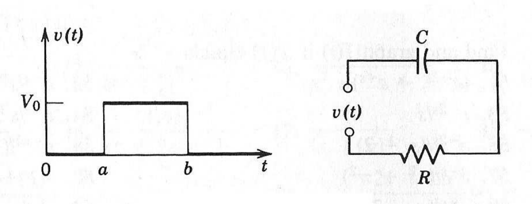
$$x * z \leq y \quad \equiv \quad x \leq y * z^{-1} \qquad (z > 0)$$

High-school example

Handling more demanding problems, eg. electrical circuits:

The problem

Predict $i(t)$ for RC-circuit

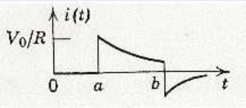


The model

$$
\begin{array}{rcl}
v(t) & = & Ri(t) + \frac{1}{C}\int_0^t i(\tau)d\tau \\
v(t) & = & V_0(u(t-a) - u(t-b)) \qquad (b > a)
\end{array}
$$

High-school example

The solution



Calculation?

Physicists and engineers overcome difficult calculations involving integral/differential equations by changing the “mathematical space”, for instance by moving (temporarily) from the time-space to the s-space in the *Laplace transformation*.

Laplace transform

$f(t)$ is transformed into $(\mathcal{L}\ f)s = \int_0^\infty e^{-st} f(t)dt$

High-school example

Laplace-transformed RC-circuit model

$$RI(s) + \frac{I(s)}{sC} = \frac{V_0}{s}(e^{-as} - e^{-bs})$$

Algebraic solution for $I(s)$

$$I(s) = \frac{\frac{V_0}{R}}{s + \frac{1}{RC}}(e^{-as} - e^{-bs})$$

Back to the t-space

$$i(t) = \begin{cases} 0 & \mathit{if} \quad t < a \\ (\frac{V_0 e^{-\frac{a}{RC}}}{R})e^{-\frac{t}{RC}} & \mathit{if} \quad a < t < b \\ (\frac{V_0 e^{-\frac{a}{RC}}}{R} - \frac{V_0 e^{-\frac{b}{RC}}}{R})e^{-\frac{t}{RC}} & \mathit{if} \quad t > b \end{cases}$$

(after some algebraic manipulation)

Lesson

Laplace transform softens the "notation conflict" involved in

$e = m + c$

$$engineering = \underline{model}\ first,\ then\ \underline{calculate} \ldots$$

arising from a

notation conflict

- descriptiveness (useful in modelling)
- compactness (for agile calculation)

Is there a "Laplace transform" applicable to software calculation?

Perhaps there is, cf. ...

$$\langle \int x \ : \ 0 < x < 10 : \ x^2 - x \rangle$$

$$\langle \forall\, x \ : \ 0 < x < 10 : \ x^2 \geq x \rangle$$

An integral transform

$(\mathcal{L}\ f)s = \int_0^\infty e^{-st} f(t)dt$, eg.

$f(t)$	$\mathcal{L}(f)$
1	$\frac{1}{s}$
t	$\frac{1}{s^2}$
t^n	$\frac{n!}{s^{n+1}}$
e^{at}	$\frac{1}{s-a}$
etc	

Pierre Laplace (1749-1827)

An "s-space equivalent" for logical quantification

The pointfree ($\mathcal{PF}$) transform

ϕ	$\mathcal{PF}\ \phi$
$\langle \exists\ a\ ::\ b\ R\ a \wedge a\ S\ c \rangle$	$b(R \cdot S)c$
$\langle \forall\ a, b\ :\ b\ R\ a :\ b\ S\ a \rangle$	$R \subseteq S$
$\langle \forall\ a\ ::\ a\ R\ a \rangle$	$id \subseteq R$
$\langle \forall\ x\ :\ x\ R\ b :\ x\ S\ a \rangle$	$b(R \setminus S)a$
$\langle \forall\ c\ :\ b\ R\ c :\ a\ S\ c \rangle$	$a(S\ /\ R)b$
$b\ R\ a \wedge c\ S\ a$	$(b, c)\langle R, S \rangle a$
$b\ R\ a \wedge d\ S\ c$	$(b, d)(R \times S)(a, c)$
$b\ R\ a \wedge b\ S\ a$	$b\ (R \cap S)\ a$
$b\ R\ a \vee b\ S\ a$	$b\ (R \cup S)\ a$
$(f\ \ b)\ R\ (g\ \ a)$	$b(f^\circ \cdot R \cdot g)a$
TRUE	$b\ \top\ a$
FALSE	$b\ \bot\ a$

What are R, S, id ?

A transform for logic and set-theory

An old idea

$$\mathcal{PF}(\text{sets, predicates}) \quad = \quad \text{pointfree binary relations}$$

Calculus of binary relations

- 1860 - introduced by De Morgan, embryonic
- 1870 - Peirce finds interesting equational laws
- 1941 - Tarski's school, cf. *A Formalization of Set Theory without Variables*
- 1980's - coreflexive models of sets (Freyd and Scedrov, Eindhoven MPC group and others)

Unifying approach

Everything is a (binary) relation

Binary Relations

Arrow notation

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

Identity of composition

id such that $R \cdot id = id \cdot R = R$

Converse

Converse of R — R° such that $a(R^{\circ})b$ iff $b\ R\ a$.

Ordering

"$R \subseteq S$ — the "R is at most S" — the obvious $R \subseteq S$ **ordering**.

Binary Relations

Pointwise meaning

$b \ R \ a$ means that pair $\langle b, a \rangle$ is in R, eg.

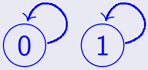
$$\begin{array}{ccc} 1 & \leq & 2 \\ \texttt{John} & \mathit{IsFatherOf} & \texttt{Mary} \\ 3 & = (1+) & 2 \end{array}$$

Reflexive and coreflexive relations

- Reflexive relation: $id \subseteq R$
- Coreflexive relation: $R \subseteq id$

Sets

Are represented by coreflexives, eg. set $\{0, 1\}$ is



PF-transform example: "indirect at-most" rule

For $\vdash$ a preorder,

Pointwise

$$X \vdash Y \equiv \langle \forall\, Z : Z \vdash X : Z \vdash Y \rangle \tag{2}$$

Pointfree

$$\vdash \; = \; \vdash \backslash \vdash \tag{3}$$

Comments

- Variables (points) X, Y, Z disappear (PF = "point+free")
- $\forall$ is gone

Calculation

One-slide long calculation of (2) — via (3) — follows shortly

Galois connections

GCs provide uniform structure to any kind of (in)equational reasoning, for example:

GC

The ("al-djabr") **rule**

$$T \cdot R \subseteq S \quad \equiv \quad R \subseteq T \setminus S$$

The **calculus** (tiny fragment!):

- Monotonicity: $(T\setminus)$ is monotonic and $(\setminus S)$ is antimonotonic
- Identity: $id \setminus S = S$
- Distributions, eg: $(R \cup T) \setminus S \;=\; (R \setminus S) \cap (T \setminus S)$
- Coreflexive Φ: $(R \cdot \Phi \setminus S) \cap \Phi \;=\; (R \setminus S) \cap \Phi$

etc

One-slide-long calculation style

$$
\begin{array}{ll}
& \vdash = \vdash \setminus \vdash \\
\equiv & \qquad \{ \text{ antisymmetry } \} \\
& \vdash \setminus \vdash \subseteq \vdash \quad \wedge \quad \vdash \subseteq \vdash \setminus \vdash \\
\equiv & \qquad \{ \text{ identity } (R = id \setminus R) \text{ and the GC itself } \} \\
& \vdash \setminus \vdash \subseteq id \setminus \vdash \quad \wedge \quad \vdash \cdot \vdash \subseteq \vdash \\
\Leftarrow & \qquad \{ \text{ antimonotonicity } \} \\
& id \subseteq \vdash \quad \wedge \quad \vdash \cdot \vdash \subseteq \vdash \\
\equiv & \qquad \{ \text{ PF-definitions of reflexive and transitive relation } \} \\
& \vdash \text{ is a preorder}
\end{array}
$$

Final touch: indirect *equality*

Thanks to the former result, we carry on:

$$
\begin{array}{c}
\vdash \cap \vdash^\circ = (\vdash \setminus \vdash) \cap (\vdash \setminus \vdash)^\circ \\
\equiv \qquad \{ \text{ in case } \vdash \text{ is antisymmetric } \} \\
id = (\vdash \setminus \vdash) \cap (\vdash \setminus \vdash)^\circ
\end{array}
$$

which — back to points — yields

Indirect equality rule

For $\vdash$ a partial order,

$$X = Y \quad \equiv \quad \langle \forall\, Z \ :: \ Z \vdash X \equiv Z \vdash Y \rangle \tag{4}$$

holds

which will be essential to PF-reasoning about the given $\vdash$ relation

PF-transform of $\vdash$

Recall

Pointwise

$$s \vdash r \stackrel{\text{def}}{=} \langle \forall\, a \;:\; \emptyset \subset (s\ a) :\; \emptyset \subset (r\ a) \subseteq (s\ a) \rangle$$

Define $R = \in \cdot r$ and $S = \in \cdot s$ and apply PF-transformation rules to obtain

Pointfree

$$S \vdash R \quad \equiv \quad \delta\, S \subseteq (R \setminus S) \cap \delta\, R \tag{5}$$

where $\delta\, S = S \cdot S^\circ \cap id$ is the coreflexive relation which denotes the **domain** of S

Goal

Calculate $s \sqcap r$ via PF-transform (5)

From *"invent & verify"* to calculation

Classical way

- invent $R \sqcap S$
- verify that $R \sqcap S$ is a common lowerbound of R and S
- verify that it is the *greatest* of all such lowerbounds

Calculational way

Calculate $\sqcap$ as the (unique) solution to universal property, for all X

$$X \vdash R \sqcap S \quad \equiv \quad X \vdash R \wedge X \vdash S \tag{6}$$

Let us solve this equation for unknown $\sqcap$:

$$\begin{array}{cl} & X \vdash R \sqcap S \\ \equiv & \quad \{ \ (6) \ \} \end{array}$$

Calculation of $\sqcap$

$$X \vdash R \wedge X \vdash S$$

$\equiv$ { (5) twice; composition of coreflexives is meet }

$$\delta X \subseteq (R \setminus X) \cap (S \setminus X) \cap \delta R \cdot \delta S$$

$\equiv$ { distribution }

$$\delta X \subseteq (R \cup S) \setminus X \cap \delta R \cdot \delta S$$

$\equiv$ { recall $(Y \cdot \Phi \setminus X) \cap \Phi = (Y \setminus X) \cap \Phi$, for $\Phi = \delta R \cdot \delta S$ }

$$\delta X \subseteq (((R \cup S) \cdot \delta R \cdot \delta S) \setminus X) \cap (\delta R \cdot \delta S)$$

$\equiv$ { $\delta ((R \cup S) \cdot \delta R \cdot \delta S) = \delta R \cdot \delta S$ (coreflexives) ; (5) }

$$X \vdash ((R \cup S) \cdot \delta R \cdot \delta S)$$

$::$ { indirect equality }

$$R \sqcap S = (R \cup S) \cdot \delta R \cdot \delta S$$

Back to points

PF-calculation has thus led to

$$R \sqcap S \;=\; (R \cup S) \cdot \delta\, R \cdot \delta\, S \tag{7}$$

which — back to points — is nothing but what was anticipated earlier on:

$$(r \sqcap s)a \stackrel{\text{def}}{=} \textit{if } (r\ a) = \emptyset \vee (s\ a) = \emptyset \textit{ then } \emptyset \textit{ else } (r\ a) \cup (s\ a)$$

Challenge (for the ones who haven't tried it yet)

Calculate the above directly from the pointwise definition of $\vdash$

Summary

- No invent & verify
- Elegance of reasoning
- Economy of thinking

However

(Lack of) monotonicity

- $R \cap S$ is **not** monotonic with respect to $\vdash$
- $R \cup S$ is **not** monotonic with respect to $\vdash$
- $R \cdot S$ is **not** monotonic with respect to $\vdash$

although

- $R \times S$ (special case of $R \cap S$) is $\vdash$-monotonic
- $R + S$ (special case of $R \cup S$) is $\vdash$-monotonic

Questions

- Why?
- Is (McCarthy) conditional

$$P \rightarrow S \ , \ T \quad \stackrel{\mathrm{def}}{=} \quad (S \cdot \delta \, P) \cup T \cdot (id - \delta \, P)$$

$\vdash$-monotonic?

Two sub-relations of $\vdash$

Do not weaken the precondition

$$S \vdash_{post} R \quad \equiv \quad S \vdash R \wedge \delta R \subseteq \delta S \tag{8}$$

Do not strengthen the postcondition

$$S \vdash_{pre} R \equiv S \vdash R \wedge S \subseteq R \cdot \delta S \tag{9}$$

By definition

$$\vdash_{pre} \;\subseteq\; \vdash \quad , \quad \vdash_{post} \;\subseteq\; \vdash \tag{10}$$

and therefore,

$$\vdash_{pre} \cdot \vdash_{post} \;\subseteq\; \vdash \tag{11}$$

$$\vdash_{post} \cdot \vdash_{pre} \;\subseteq\; \vdash \tag{12}$$

Simple PF-calculations lead to

$$S \vdash_{pre} R \quad \equiv \quad R \cdot \delta\, S = S \tag{13}$$

$$S \vdash_{post} R \quad \equiv \quad R \subseteq S \wedge \delta\, R = \delta\, S \tag{14}$$

where (14) can be written in less symbols as

$$\vdash_{post} \quad = \quad \subseteq^{\circ} \cap\, \delta^{\circ} \cdot \delta \tag{15}$$

Also easy to show

$$S \vdash_{pre} S \cup R \quad \equiv \quad R \cdot \delta\, S \subseteq S \tag{16}$$

$$S \cup R \vdash_{post} R \quad \equiv \quad \delta\,(S \cup R) = \delta\, R \tag{17}$$

$$S \vdash_{post} S \cap R \quad \equiv \quad \delta\, S = \delta\,(R \cap S) \tag{18}$$

$$S \cap R \vdash_{pre} R \quad \equiv \quad R \cdot \delta\,(S \cap R) \subseteq S \cap R \tag{19}$$

(eg. (19) is (16) for $S := S \cap R$)

$\vdash \subseteq \vdash_{pre} \cdot \vdash_{post}$

$$
\begin{array}{ll}
S \vdash R & \\
\equiv & \{ \text{ expand } \delta\, S \subseteq (R \setminus S) \cap \delta\, R \text{ (Galois) } \} \\
& R \cdot \delta\, S \subseteq S \wedge \delta\, S \subseteq \delta\, R \\
\equiv & \{ A \subseteq B \equiv A \cup B \} \\
& R \cdot \delta\, S \subseteq S \wedge (\delta\, S) \cup (\delta\, R) = \delta\, R \\
\equiv & \{ \ \delta \text{ distributes over } \cup \text{ (Galois) } \} \\
& R \cdot \delta\, S \subseteq S \wedge \delta\, (S \cup R) = \delta\, R \\
\equiv & \{ \text{ previous slide } (16\ ,\ 17) \ \} \\
& (S \vdash_{pre} S \cup R) \wedge (S \cup R) \vdash_{post} R \\
\Rightarrow & \{ \text{ composition } \} \\
& S(\vdash_{pre} \cdot \vdash_{post})R
\end{array}
$$

$\vdash \subseteq \vdash_{post} \cdot \vdash_{pre}$

$$
\begin{array}{ll}
S \vdash R & \\
\equiv & \{\ \text{since}\ \ S \vdash R \ \Rightarrow\ \delta\, S = \delta\,(S \cap R)\ \} \\
& \delta\, S = \delta\,(S \cap R) \wedge R \cdot \delta\, S \subseteq S \\
\equiv & \{\ R \cdot \delta\, S \text{ is at most } R\ ;\ \cap\text{-universal}\ \} \\
& \delta\, S = \delta\,(S \cap R) \wedge R \cdot \delta\, S \subseteq S \cap R \\
\equiv & \{\ \text{substitution}\ \} \\
& \delta\, S = \delta\,(S \cap R) \wedge R \cdot \delta\,(S \cap R) \subseteq S \cap R \\
\equiv & \{\ (18) \text{ and } (19)\ \} \\
& (S \vdash_{post} S \cap R) \wedge (S \cap R) \vdash_{pre} R \\
\Rightarrow & \{\ \text{composition}\ \} \\
& S(\vdash_{post} \cdot \vdash_{pre})R
\end{array}
$$

$\vdash$-Factorization

Diagram

$$
\begin{tikzcd}
 & S \cup R \arrow[dr, dash, "\vdash_{post}"] & \\
S \arrow[ur, dash, "\vdash_{pre}"] \arrow[rr, dash, "\vdash"'] \arrow[dr, dash, "\vdash_{post}"'] & & R \\
 & S \cap R \arrow[ur, dash, "\vdash_{pre}"'] &
\end{tikzcd}
$$

Summary

$$\vdash_{pre} \cdot \vdash_{post} \;=\; \vdash \;=\; \vdash_{post} \cdot \vdash_{pre} \tag{20}$$

Comments

PF-calculation style free of extra ingredients such as **negation** and **consistency** (special case where δ distributes over $\cap$).

Example of reasoning by factorization — conditional

$$
\begin{tikzcd}
 & P \to S \cup S' \ , \ T \arrow[dl, dash, "\vdash_{pre}"'] \arrow[dr, dash, "\vdash_{post}"] & \\
P \to S \ , \ T \arrow[rr, dash, "\vdash"'] & & P \to S' \ , \ T
\end{tikzcd}
$$

$(\cdot T)$ and $(\cup T)$ are $\vdash_{post}$-monotonic

$$
S \vdash_{post} R \quad \Rightarrow \quad S \cdot T \vdash_{post} R \cdot T \tag{21}
$$

$$
S \vdash_{post} R \quad \Rightarrow \quad S \cup T \vdash_{post} R \cup T \tag{22}
$$

Therefore

(McCarthy) conditional is $\vdash_{post}$-monotonic

$$
\left\{ \begin{array}{l} P \vdash_{post} P' \\ S \vdash_{post} S' \\ T \vdash_{post} T' \end{array} \right. \quad \Rightarrow \quad P \to S \ , \ T \ \vdash_{post} \ P' \to S' \ , \ T' \tag{23}
$$

Example of reasoning by factorization — conditional

$(\cdot\Phi)$ is $\vdash_{pre}$-monotonic (Φ coreflexive)

$$S \vdash_{pre} R \quad \Rightarrow \quad S \cdot \Phi \vdash_{pre} R \cdot \Phi \tag{24}$$

(Constrained) $(\cup T)$ $\vdash_{pre}$-monotonicity

$$S \vdash_{pre} R \quad \Rightarrow \quad S \cup T \vdash_{pre} R \cup T \equiv R \cdot \delta\, T \subseteq S \cup T \tag{25}$$

entailing

$$S \vdash_{pre} R \ \wedge\ R \cdot \delta\, T \subseteq S \cup T \quad \Rightarrow \quad S \cup T \vdash_{pre} R \cup T \tag{26}$$

Example of reasoning by factorization — conditional

Monotonicity of *then*-branch of conditional:

$$\begin{tikzcd}
 & P \rightarrow S \cup S' \,,\; R \arrow[dl, dash, "\vdash_{pre}"'] \arrow[dr, dash, "\vdash_{post}"] & \\
P \rightarrow S \,,\; R \arrow[rr, dash, "\vdash"'] & & P \rightarrow S' \,,\; R
\end{tikzcd}$$

Only $\vdash_{pre}$-factor matters:

$$\begin{array}{cl}
 & P \rightarrow S \,,\; R \quad \vdash_{pre} \quad P \rightarrow S \cup S' \,,\; R \\
\equiv & \quad \{ \text{ definition ; abbreviate } T := R \cdot (id - \delta\, P) \ \} \\
 & (S \cdot \delta\, P) \cup T \quad \vdash_{pre} \quad ((S \cup S') \cdot \delta\, P) \cup T \\
\Leftarrow & \quad \{ \ (24) \text{ and } (26) \ \} \\
 & S \vdash_{pre} S \cup S' \ \wedge \ (S \cup S') \cdot (\delta\, P) \cdot (\delta\, T) \subseteq (S \cdot \delta\, P) \cup T
\end{array}$$

Example of reasoning by factorization — conditional

$$\Leftarrow \quad \{ \text{ factorization of } S \vdash S' \text{ and domain of } T \; \}$$

$$S \vdash S' \;\wedge\; (S \cup S') \cdot (\delta\, P) \cdot (\delta\, R) \cdot (id - \delta\, P) \subseteq (S \cdot \delta\, P) \cup T$$

$$\equiv \quad \{ \; \delta\, P \cdot (id - \delta\, P) = \perp \; \}$$

$$S \vdash S' \;\wedge\; \textsc{True}$$

Since monotonicity of *else*-branch is analogous, we get

McCarthy conditional monotonicity

$$\left\{ \begin{array}{l} P \vdash_{post} P' \\ S \vdash S' \\ T \vdash T' \end{array} \right. \quad \Rightarrow \quad P \rightarrow S \;,\; T \;\vdash\; P' \rightarrow S' \;,\; T' \qquad (27)$$

Refinement across relational taxonomy

Binary relation sub-class	$\vdash_{post}$	$\vdash_{pre}$	$\vdash$
Entire relations	$\subseteq^\circ$	id	$\subseteq^\circ$
Simple relations	id	$\subseteq$	$\subseteq$
Functions	id	id	id

where

- **Entire** R — $id \subseteq R^\circ \cdot R$ (vulg. total)
- **Simple** R — $R \cdot R^\circ \subseteq id$ (vulg. univocal)
- **Function** f — both entire and simple

(Polytypic) structural refinement

$\vdash$-monotonicity of an arbitrary parametric type F

$$S \vdash R \quad \Rightarrow \quad \mathsf{F}\, S \vdash \mathsf{F}\, R \tag{28}$$

Technically, parametricity is captured by regarding F as a *relator*:

Relators

A **relator** *F* is a functor on relations, for instance

$$\begin{array}{ccc} A & & \mathsf{F}\,A \\ {\scriptstyle R}\big\downarrow & & \big\downarrow{\scriptstyle \mathsf{F}\,R} \\ B & & \mathsf{F}\,B \end{array}$$

which is $\subseteq$-monotonic and commutes with composition, converse and the identity.

Structural refinement example

Sequence relator ("map")

$$
\begin{array}{ccc}
A & & A^\star \\
R \big\downarrow & & \big\downarrow R^\star \\
B & & B^\star
\end{array}
$$

where

$$l(R^\star)l' \;\equiv\; \mathsf{len}\, l = \mathsf{len}\, l' \wedge \langle \forall\, i \;:\; i \in \mathsf{inds}\, l :\; (l\; i) R (l'\; i) \rangle$$

Sequence ("map") refinement example

$$> \vdash \mathit{succ} \;\Rightarrow\; (>)^\star \vdash (\mathit{succ})^\star$$

that is

$$\langle \forall\, x \;::\; x + 1 > x \rangle \;\Rightarrow\; \langle \forall\, l \;::\; [\, x + 1 \mid x \leftarrow l \,] >^\star l \rangle$$

Calculation of (28)

Since

Every relator F is both $\vdash_{pre}/\vdash_{post}$-monotonic

$$F \cdot \vdash_{post} \;\subseteq\; \vdash_{post} \cdot F \tag{29}$$

$$F \cdot \vdash_{pre} \;\subseteq\; \vdash_{pre} \cdot F \tag{30}$$

the structural refinement law (28) is easy to calculate :

$$\begin{array}{ll}
 & \textsc{True} \\
\equiv & \quad \{ \; (29) \; \} \\
 & F \cdot \vdash_{post} \;\subseteq\; \vdash_{post} \cdot F \\
\Rightarrow & \quad \{ \; \text{monotonicity of composition} \; \} \\
 & F \cdot \vdash_{post} \cdot \vdash_{pre} \;\subseteq\; \vdash_{post} \cdot F \cdot \vdash_{pre}
\end{array}$$

Calculation of (28)

$$
\begin{array}{ll}
\Rightarrow & \{ \text{ (30) and } \subseteq\text{-transitivity } \} \\
& \mathsf{F} \cdot \vdash_{post} \cdot \vdash_{pre} \;\subseteq\; \vdash_{post} \cdot \vdash_{pre} \cdot \mathsf{F} \\
\equiv & \{ \text{ (20) } \} \\
& \mathsf{F} \cdot \vdash \;\subseteq\; \vdash \cdot \mathsf{F} \\
\equiv & \{ \text{ go pointwise on } S \text{ and } R \text{ } \} \\
& R \vdash S \;\Rightarrow\; \mathsf{F}\, R \vdash \mathsf{F}\, S
\end{array}
$$

Related work

- Boudriga et al (FAC, 1992) formulate $S \vdash R$ relationally but reason at pointwise level
- Lindsay Groves (BCS FACS Refinement Workshop, 2002) postulates and then proves $\vdash$ factorization in the context of the Z schema calculus, requiring the extra notion of *compatible* relations, which complicates proofs unnecessarily.
- Wolfram Kahl (ENTCS, 2003) presents the factorization at PF-level with no further use of it.

Current work on PF-transform

Coalgebraic refinement

Our main goal is to apply this factorization to the refinement of "components as coalgebras" — eg. (monadic) machines (=objects) of type

$$B \times (M\ A)^I \longleftarrow A$$

where M is a behaviour monad — cf. Barbosa and Meng (AMAST'04).

Databases

PF-refactoring of **database theory** — functional and multivalued dependencies made simpler and more general

Current work on PF-transform

Data refinement

Data-refinement calculus based on (in)equations of shape

$$
\begin{tikzcd}
A \arrow[r, bend left, "R"] \arrow[r, phantom, "\leq"] & B \arrow[l, bend left, "F"]
\end{tikzcd}
\qquad \text{such that } F^{\circ} \vdash R
$$

where F is the *abstraction* relation and R the *representation*.

Other topics on the PF-transform

PF-transform automation

- RelView (Berghammer et al)
- UMinho Haskell Libraries (Cunha, Visser et al)

Multirelations

Cf. angelic / demonic nondeterminism, etc

Alloy

PF-transform applied to `Alloy` model reasoning (where "everything is a relation")

Summary

- Invest in **perennial** reasoning strategies
- Shift from "implication first" to "let the symbols do the work"

  "Chase" equivalence : bad use of implication-first logic may lead to "50% loss in theory"

- Rôle of **transforms**, **abstract** notation and abstract patterns (easier to spot *al-djabr* rules)
- Stimulate **elegance** in mathematics (it is effective!)
- Learn with the other engineering disciplines