Revisiting Invariants

Luís Barbosa[1] J.N. Oliveira[1] Alexandra Silva[2]

[1]DI - CCTC, Univ. Minho, Braga
[2]CWI, Amsterdam

IFIP WG1.3
March, 23 - 24, 2007
Braga, Portugal

Motivation

Previous work on software components

- as persistent (state-based) and interacting entities
- leads to the development of coalgebraic models as generalised (= parametrized by a strong monad) Mealy machines [Bar00]
- and calculi to compose components and reason compositionally about them [BO02,BO03].

... but somehow neglected the ubiquity of "business rules" in systems design.

Motivation

Previous work on software components

- as persistent (state-based) and interacting entities
- leads to the development of coalgebraic models as generalised (= parametrized by a strong monad) Mealy machines [Bar00]
- and calculi to compose components and reason compositionally about them [BO02,BO03].

... but somehow neglected the ubiquity of "business rules" in systems design.

Clearly, most "business rules" are invariants. But

- how can we calculate with invariants, in a generic way?
- and preserve them along the component assembly process?

Invariants

Definition (by Bart Jacobs)

An invariant for a coalgebra $c : X \to F(X)$ is a predicate $P \subseteq X$ which is "closed under c":

$$x \in P \quad \Rightarrow \quad c(x) \in Pred(F)(P)$$

for all $x \in X$.

Question

Is such a definition amenable to formal calculation?
(formal $\equiv$ in a *let-the-symbols-do-the-work* style)

Modelling vs Calculating

The use of formal modelling methods often raises a kind of

Notation conflict

between

- *descriptiveness* — ie., adequacy to describe domain-specific objects and properties and build suitable models, and
- *compactness* — as required by algebraic reasoning and solution calculation.

Modelling vs Calculating

The use of formal modelling methods often raises a kind of

Notation conflict

between

- *descriptiveness* — ie., adequacy to describe domain-specific objects and properties and build suitable models, and
- *compactness* — as required by algebraic reasoning and solution calculation.

More demanding problems entails the need for a temporary change of the working "mathematical space", e.g.

Laplace transform

From the time-space to the s-space:
$f(t)$ is transformed into $(\mathcal{L}\ f)s = \int_0^\infty e^{-st} f(t) dt$

Quoting Kreyszig's book, p.242

"(...) The Laplace transformation is a method for solving differential equations (...) [which] consists of three main steps:

1st step. *The given "hard" problem is transformed into a "simple" equation (subsidiary equation).*
2nd step. *The subsidiary equation is solved by* **purely algebraic** *manipulations.*
3rd step. *The solution of the subsidiary equation is transformed back to obtain the solution of the given problem.*

In this way the Laplace transformation reduces the problem of solving a differential equation to an **algebraic problem***".*

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, $\bot$, ...?

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
- 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.

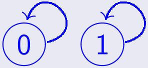
$$\begin{array}{ccc} 1 & \leq & 2 \\ \mathtt{John} & \mathit{IsFatherOf} & \mathtt{Mary} \\ 3 & = (1+) & 2 \end{array}$$

Reflexive and coreflexive relations

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

Sets

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



Algebraic manipulation

Algebraic (“al-djabr”) rules, as **Galois connections**

$$\begin{tikzcd}[column sep=tiny] |[draw, circle]| f \arrow[rr, bend right] & \cdot R \subseteq S \quad \equiv R \subseteq & |[draw, circle]| f^{\circ} & \cdot S \end{tikzcd}$$

$$\begin{tikzcd}[column sep=tiny] R \cdot & |[draw, circle]| f^{\circ} \arrow[rr, bend right] & \subseteq S \quad \equiv R \subseteq S \cdot & |[draw, circle]| f \end{tikzcd}$$

$$\begin{tikzcd}[column sep=tiny] |[draw, circle]| T \arrow[rr, bend right] & \cdot R \subseteq S \quad \equiv R \subseteq & |[draw, circle]| T & \setminus S \end{tikzcd}$$

or **closure** rules, eg. (for Φ coreflexive),

$$\begin{tikzcd}[column sep=tiny] |[draw, circle]| \Phi \arrow[rr, bend right] & \cdot R \subseteq S \quad \equiv \Phi \cdot R \subseteq & |[draw, circle]| \Phi & \cdot S \end{tikzcd}$$

Invariants PF-transformed

Imploding the outermost $\forall$ in Jacobs definition:

$$
\begin{array}{ll}
 & \langle \forall\ x\ ::\ x \in P \Rightarrow c(x) \in\ Pred(F)(P) \rangle \\
\equiv & \qquad \{\ \text{sets as coreflexive relations}\ \} \\
 & \langle \forall\ x\ ::\ x\ P\ x \Rightarrow (c\ x)\ Pred(F)(P)\ (c\ x) \rangle \\
\equiv & \qquad \{\ \text{PF-transform rule } (f\ \ b)R(g\ \ a) \equiv b(f^{\circ} \cdot R \cdot g)a\ \ \} \\
 & \langle \forall\ x\ ::\ x\ P\ x \Rightarrow x(c^{\circ} \cdot Pred(F)(P) \cdot c)x) \rangle \\
\equiv & \qquad \{\ \text{drop variables (PF-transform of inclusion)}\ \ \} \\
 & P \subseteq c^{\circ} \cdot Pred(F)(P) \cdot c \\
\equiv & \qquad \{\ \ \text{introduce relator ; shunting rule}\ \ \} \\
 & c \cdot P \subseteq (\mathsf{F}\ P) \cdot c \\
\equiv & \qquad \{\ \ \text{introduce Reynolds combinator}\ \ \} \\
 & c(\mathsf{F}\ P \leftarrow P)c
\end{array}
$$

About Reynolds arrow

Reynolds arrow combinator is a relation on functions

$$f(R \leftarrow S)g \quad \equiv \quad f \cdot S \subseteq R \cdot g \qquad \text{cf. diagram}$$

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

useful in expressing properties of functions — namely
monotonicity

$$B \xleftarrow{f} A \text{ is monotonic} \quad \equiv \quad f(\leq_B \leftarrow \leq_A)f$$

polymorphism (free theorem):

$$\mathsf{G}\, A \xleftarrow{f} \mathsf{F}\, A \text{ is polymorphic} \quad \equiv \quad \langle \forall\ R\ ::\ f(\mathsf{G}\, R \leftarrow \mathsf{F}\, R)f \rangle$$

etc

Invariants as coreflexive bisimulations

Re-working the calculation backwards, and considering two coalgebras c and d and a relation R on their state spaces:

$$
\begin{array}{ll}
 & c(\mathsf{F}\,R \leftarrow R)d \\
\equiv & \qquad \{ \text{ Reynolds combinator } \} \\
 & c \cdot R \subseteq (\mathsf{F}\,R) \cdot d \\
\equiv & \qquad \{ \text{ shunting rule; drop variables (PF-transform of inclusion) } \} \\
 & \langle \forall\ x, y\ ::\ x\ R\ y \Rightarrow x(c^{\circ} \cdot \mathsf{F}\,R \cdot d)y) \rangle \\
\equiv & \qquad \{ \text{ PF-transform rule } (f\ \ b)R(g\ \ a) \equiv b(f^{\circ} \cdot R \cdot g)a \ \} \\
 & \langle \forall\ x, y\ ::\ x\ R\ y \Rightarrow (c\ \ x)\ \mathsf{F}\,R\ (d\ \ y) \rangle
\end{array}
$$

Invariants as coreflexive bisimulations

... arrive at:

Definition (by Bart Jacobs):

A bisimulation for coalgebras $c : X \to F(X)$ and $d : Y \to F(Y)$ is a relation $R \subseteq X \times Y$ which is “closed under c and d”:

$$(x, y) \in R \quad \Rightarrow \quad (c(x), d(y)) \in Rel(F)(R)$$

for all $x \in X$ and $y \in Y$.

Question

Having put both *invariants* and *bisimulations* in a common setting

— as Reynolds arrows —

how can our *reasoning power* be enriched?

Why Reynolds arrow matters?

Useful and manageable PF-properties

For example

$$\begin{aligned} id \leftarrow id &= id && (1)\\ (R \leftarrow S)^\circ &= R^\circ \leftarrow S^\circ && (2)\\ R \leftarrow S \subseteq V \leftarrow U &\Leftarrow R \subseteq V \wedge U \subseteq S && (3)\\ (R \leftarrow V) \cdot (S \leftarrow U) &\subseteq (R \cdot S) \leftarrow (V \cdot U) && (4) \end{aligned}$$

recalled from Backhouse's "*On a relation on functions*" (1990)

Why Reynolds arrow matters

Get monotony on the consequent side and thus,

$$S \leftarrow R \subseteq (S \cup V) \leftarrow R \tag{5}$$

$$\top \leftarrow S = \top \tag{6}$$

anti-monotony on the antecedent one

$$R \leftarrow \bot = \top \tag{7}$$

and two distributive laws:

$$S \leftarrow (R_1 \cup R_2) = (S \leftarrow R_1) \cap (S \leftarrow R_2) \tag{8}$$

$$(S_1 \cap S_2) \leftarrow R = (S_1 \leftarrow R) \cap (S_2 \leftarrow R) \tag{9}$$

Why Reynolds arrow matters

Ex: *id* is a bisimulation

$$
\begin{array}{cl}
 & c(\mathsf{F}\, id \leftarrow id)d \\
\equiv & \quad \{ \ \text{relator F preserves the identity} \ \} \\
 & c(id \leftarrow id)d \\
\equiv & \quad \{ \ (1) \ \} \\
 & c\ (id)\ d \\
\equiv & \quad \{ \ id\ x = x \ \} \\
 & c = d
\end{array}
$$

Why Reynolds arrow matters

Ex: the converse of a bisimulation is a bisimulation

$$
\begin{array}{ll}
& c(\mathsf{F}\,R \leftarrow R)d \\
\equiv & \qquad \{ \ \text{converse} \ \} \\
& d(\mathsf{F}\,R \leftarrow R)^\circ c \\
\equiv & \qquad \{ \ (2) \ \} \\
& d((\mathsf{F}\,R)^\circ \leftarrow R^\circ)c \\
\equiv & \qquad \{ \ \text{relator } \mathsf{F} \ \} \\
& d(\mathsf{F}(R^\circ) \leftarrow R^\circ)c
\end{array}
$$

Why Reynolds arrow matters

Ex: bisimulations are closed under composition

Therefore,

$$
\begin{array}{cl}
 & (\mathsf{F}\,R_1 \leftarrow R_1) \cap (\mathsf{F}\,R_2 \leftarrow R_2) \\
\subseteq & \qquad \{ \ (5)\ (\text{twice})\ ;\ \text{monotonicity of meet} \ \} \\
 & ((\mathsf{F}\,R_1 \cup \mathsf{F}\,R_2) \leftarrow R_1) \cap ((\mathsf{F}\,R_1 \cup \mathsf{F}\,R_2) \leftarrow R_2) \\
= & \qquad \{ \ (8) \ \} \\
 & (\mathsf{F}\,R_1 \cup \mathsf{F}\,R_2) \leftarrow (R_1 \cup R_2) \\
= & \qquad \{ \ \text{relators} \ \} \\
 & \mathsf{F}(R_1 \cup R_2) \leftarrow (R_1 \cup R_2)
\end{array}
$$

Why Reynolds arrow matters

Ex: behavioural equivalence is a bisimulation

$$u R v \;\equiv\; [\![c]\!] u = [\![d]\!] v \qquad R \text{ is a bisimulation}$$

$$\begin{array}{cl}
 & c (\mathsf{F}\,([\![c]\!]^\circ \cdot [\![d]\!]) \leftarrow [\![c]\!]^\circ \cdot [\![d]\!]) d \\
\equiv & \qquad \{ \text{ definition } \} \\
 & [\![c]\!]^\circ \cdot [\![d]\!] \subseteq c^\circ \cdot \mathsf{F}\,([\![c]\!]^\circ \cdot [\![d]\!]) \cdot d \\
\equiv & \qquad \{ \text{ relators } \} \\
 & [\![c]\!]^\circ \cdot [\![d]\!] \subseteq c^\circ \cdot \mathsf{F}\,[\![c]\!]^\circ \cdot \mathsf{F}\,[\![d]\!] \cdot d \\
\equiv & \qquad \{ \text{ converse } \} \\
 & [\![c]\!]^\circ \cdot [\![d]\!] \subseteq (\mathsf{F}\,[\![c]\!] \cdot c)^\circ \cdot \mathsf{F}\,[\![d]\!] \cdot d
\end{array}$$

Why Reynolds arrow matters

Ex: behavioural equivalence is a bisimulation

$$
\begin{array}{cl}
& [\![c]\!]^\circ \cdot [\![d]\!] \subseteq (\mathsf{F}\, [\![c]\!] \cdot c)^\circ \cdot \mathsf{F}\, [\![d]\!] \cdot d \\
\equiv & \quad \{ \text{ universal property of coinductive extension } \} \\
& [\![c]\!]^\circ \cdot [\![d]\!] \subseteq (\omega \cdot [\![c]\!])^\circ \cdot \omega \cdot [\![d]\!] \\
\equiv & \quad \{ \text{ converse } \} \\
& [\![c]\!]^\circ \cdot [\![d]\!] \subseteq [\![c]\!]^\circ \cdot \omega^\circ \cdot \omega \cdot [\![d]\!] \\
\equiv & \quad \{ \text{ Lambek (final coalgebra is an isomorphism) } \} \\
& \mathit{true}
\end{array}
$$

Why Reynolds arrow matters

... too simple and obvious, even *without* Reynolds arrow in the play. But, consider now the equivalence between Jacobs and Aczel-Mendler's definition of *bisimulation*

Definition (by Aczel & Mendler)

Given two coalgebras $c : X \longrightarrow F(X)$ and $d : Y \longrightarrow F(Y)$ an F-bisimulation is a relation $R \subseteq X \times Y$ which can be extended to a coalgebra ρ such that projections π_1 and π_2 lift to F-comorphisms, as expressed by

$$\begin{tikzcd}
 & R \arrow[dl, "\pi_1"'] \arrow[dr, "\pi_2"] \arrow[dd, "\rho"'] & \\
X \arrow[dd, "c"'] & & Y \arrow[dd, "d"] \\
 & \mathsf{F}\, R \arrow[dl, "\mathsf{F}\,\pi_1"'] \arrow[dr, "\mathsf{F}\,\pi_2"] & \\
\mathsf{F}\, X & & \mathsf{F}\, Y
\end{tikzcd}$$

Jacobs ≡ Aczel & Mendler

$$
\begin{array}{ll}
 & c(\mathsf{F}\,R \leftarrow R)d \\
\equiv & \qquad \{ \text{ tabulate } R = \pi_1 \cdot \pi_2^\circ \ \} \\
 & c(\mathsf{F}(\pi_1 \cdot \pi_2^\circ) \leftarrow (\pi_1 \cdot \pi_2^\circ))d \\
\equiv & \qquad \{ \text{ relator commutes with composition and converse } \} \\
 & c(((\mathsf{F}\,\pi_1) \cdot (\mathsf{F}\,\pi_2)^\circ) \leftarrow (\pi_1 \cdot \pi_2^\circ))d \\
\equiv & \qquad \{ \text{ fusion [CIC'06] law } \} \\
 & c((\mathsf{F}\,\pi_1 \leftarrow \pi_1) \cdot ((\mathsf{F}\,\pi_2)^\circ \leftarrow \pi_2^\circ))d \\
\equiv & \qquad \{ \ (2) \ \} \\
 & c((\mathsf{F}\,\pi_1 \leftarrow \pi_1) \cdot (\mathsf{F}\,\pi_2 \leftarrow \pi_2)^\circ)d \\
\equiv & \qquad \{ \text{ go pointwise (composition) } \} \\
 & \langle \exists\ a\ ::\ c(\mathsf{F}\,\pi_1 \leftarrow \pi_1)a \wedge d(\mathsf{F}\,\pi_2 \leftarrow \pi_2)a \rangle
\end{array}
$$

cf.

$$\begin{tikzcd}
X \arrow[ddd, "c"'] & & Y \arrow[ll, "R"'] \arrow[ddd, "d"] \\
& Z \arrow[ul, "\pi_1"] \arrow[ur, "\pi_2"'] \arrow[d, "a"] & \\
& \mathsf{F}\,Z \arrow[dl, "\mathsf{F}\,\pi_1"'] \arrow[dr, "\mathsf{F}\,\pi_2"] & \\
\mathsf{F}\,X & & \mathsf{F}\,Y \arrow[ll, "\mathsf{F}\,R"]
\end{tikzcd}$$

Why Reynolds arrow matters

Meaning of $\langle \exists\, a \ ::\ c(\mathsf{F}\,\pi_1 \leftarrow \pi_1)a \wedge d(\mathsf{F}\,\pi_2 \leftarrow \pi_2)a \rangle$:

> there exists a coalgebra a whose carrier is the "graph" of bisimulation R and which is such that projections π_1 and π_2 lift to the corresponding coalgebra morphisms.

Comments:

- One-slide-long proofs are easier to grasp — for a (longer) bi-implication proof of the above see Backhouse & Hoogendijk's paper on *dialgebras* (1999)
- Rule $(r \cdot s^\circ) \leftarrow (f \cdot g^\circ) = (r \leftarrow f) \cdot (s \leftarrow g)^\circ$ does most of the work — its proof is an example of generic, stepwise PF-reasoning [CIC'06, paper to appear]

Why Reynolds arrow matters

Ex: coalgebra morphisms entail bisimulation
Immediate, since inclusion of functions is equality:

$$c(\mathsf{F}\,h \leftarrow h)d \quad \equiv \quad c \cdot h = (\mathsf{F}\,h) \cdot d \tag{10}$$

However, in the Aczel & Mendler setting becomes:
Let $h : d \longleftarrow c$ a coalgebra morphism and conjecture $\gamma : \mathsf{F}\,h \longleftarrow h$

$$\gamma \;=\; \mathsf{F}\,(\pi_2)^\circ \cdot d \cdot \pi_2 \tag{11}$$

Now prove the diagram commutes: i.e., both π_1 and π_2 are coalgebra morphisms, i.e.,

$$\mathsf{F}\,\pi_1.\gamma = c \cdot \pi_1 \qquad \mathsf{F}\,\pi_2.\gamma = d \cdot \pi_2 \tag{12}$$

Clearly, π_2 is a coalgebra *iso*morphism. Then, prove that π_1 is also a colagebra morphism, i.e.,

$$c \cdot \pi_1 \;=\; \mathsf{F}\,\pi_1 \cdot \gamma \tag{13}$$

Why Reynolds arrow matters

$$
\begin{array}{ll}
 & c \cdot \pi_1 \;=\; \mathsf{F}\,\pi_1 \cdot \gamma \\
\equiv & \qquad \{ \text{ conjecture on } \gamma\text{; functors } \} \\
 & c \cdot \pi_1 \;=\; \mathsf{F}\,(\pi_1 \cdot (\pi_2)^\circ) \cdot d \cdot \pi_2 \\
\equiv & \qquad \{ \; h = \pi_1 \cdot (\pi_2)^\circ \; \} \\
 & c \cdot \pi_1 \;=\; \mathsf{F}\,h \cdot d \cdot \pi_2 \\
\equiv & \qquad \{ \; h \text{ morphism } \} \\
 & c \cdot \pi_1 \;=\; c \cdot h \cdot \pi_2 \\
\equiv & \qquad \{ \; \pi_2 \text{ iso, } h = \pi_1 \cdot (\pi_2)^\circ \; \} \\
 & c \cdot \pi_1 \;=\; c \cdot \pi_1
\end{array}
$$

Why Reynolds arrow matters

Now the converse direction: if h is a function st the diagram commutes, h is a coalgebra morphism.

$$
\begin{array}{cl}
 & c \cdot h \;=\; \mathsf{F}\,h \cdot d \\
\equiv & \qquad \{\; h = \pi_1 \cdot (\pi_2)^\circ, \text{ functors } \} \\
 & c \cdot \pi_1 \cdot (\pi_2)^\circ \;=\; \mathsf{F}\,\pi_1 \cdot \mathsf{F}\,(\pi_2)^\circ \cdot d \\
\equiv & \qquad \{\; \text{hyp: } (12) \;\} \\
 & \mathsf{F}\,\pi_1 . \gamma \cdot (\pi_2)^\circ \;=\; \mathsf{F}\,\pi_1 \cdot \mathsf{F}\,(\pi_2)^\circ \cdot d \\
\equiv & \qquad \{\; \gamma \text{ definition and } \pi_2 \text{ is iso } \} \\
 & \mathsf{F}\,\pi_1 . \gamma \;=\; \mathsf{F}\,\pi_1 \cdot \gamma
\end{array}
$$

Invariants

Invariants are coreflexive bisimulations

$$c(\mathsf{F}\,\Phi \leftarrow \Phi)c$$

Get for free:

- *id* (everywhere true predicate) is largest invariant
- $\perp$ (everywhere false) is the least one
- Invariants are closed by disjunction (ie. union), ...

Modalities

Invariants bring about *modalities*:

$$\begin{array}{rcl} c(\mathsf{F}\,\Phi \leftarrow \Phi)c & \equiv & c \cdot \Phi \subseteq \mathsf{F}\,\Phi \cdot c \\ & \equiv & \{ \ \text{shunting rule} \ \} \\ & & \Phi \subseteq \underbrace{c^\circ \cdot (\mathsf{F}\,\Phi) \cdot c}_{\bigcirc_c \Phi} \end{array}$$

since we define the *"next time X holds"* modal operator as

$$\bigcirc_c X \overset{\text{def}}{=} c^\circ \cdot (\mathsf{F}\,X) \cdot c$$

Φ invariant $\equiv$ $\Phi \subseteq \bigcirc \Phi$

$$\begin{array}{rcl} c(\mathsf{F}\,\Phi \leftarrow \Phi)c & \equiv & c \cdot \Phi \subseteq \mathsf{F}\,\Phi \cdot c \\ & \equiv & \Phi \subseteq c^\circ \cdot \mathsf{F}\,\Phi \cdot c \\ & \equiv & \Phi \subseteq \bigcirc \Phi \end{array}$$

Modalities

In PF-refactoring of *database theory* [Oli06] has derived Galois connection

$$\pi_{g,f} R \subseteq S \quad \equiv \quad R \subseteq g^{\circ} \cdot S \cdot f \tag{14}$$

in order to get (for free) properties of lower adjoint $\pi_{g,f}$.

Interesting enough, an instance of such a connection

$$\begin{tikzcd} \pi_c \arrow[rrrr, bend right] & \Phi \subseteq \Psi & \equiv & \Phi \subseteq & \bigcirc_c & \Psi \end{tikzcd} \tag{15}$$

(within coreflexives) can be re-used to obtain (again for free) properties — now — of the upper adjoint $\bigcirc_c$:

Modalities

As as upper adjoint in a Galois connection,

- $\bigcirc_c$ is **monotonic** — thus simple proofs such as

$$\begin{array}{cl} & \Phi \text{ is an invariant} \\ \equiv & \qquad \{ \text{ PF-definition of invariant } \} \\ & \Phi \subseteq \bigcirc_c \Phi \\ \Rightarrow & \qquad \{ \text{ monotonicity } \} \\ & \bigcirc_c \Phi \subseteq \bigcirc_c (\bigcirc_c \Phi) \\ \equiv & \qquad \{ \text{ PF-definition of invariant } \} \\ & \bigcirc_c \Phi \text{ is an invariant} \end{array}$$

- $\bigcirc_c$ **distributes** over conjunction, that is PF-equality

$$\bigcirc_c (\Phi \cdot \Psi) \quad = \quad (\bigcirc_c \Phi) \cdot (\bigcirc_c \Psi)$$

holds, etc

Modalities

Further modal operators, for instance $\Box\Psi$ — *henceforth* Ψ — usually defined as *the largest invariant at most* Ψ:

$$\Box\Psi \quad = \quad \langle \bigcup \ \Phi \ :: \ \Phi \subseteq \Psi \cap \bigcirc_c \Phi \rangle$$

which shrinks to a greatest (post)fix-point

$$\Box\Psi \quad = \quad \langle \nu \ \Phi \ :: \ \Psi \cdot \bigcirc_c \Phi \rangle$$

where meet (of coreflexives) is replaced by composition, as this paves the way to agile reasoning

Modalities

Ex: $\Box\Phi = \Phi \;\equiv\; \Phi\ inv$

(cf, [Jacobs,06] Lemma 4.2.6, pg 109)

$\Box\Phi \subseteq \Phi$ is obvious from the definition, but

$$\begin{array}{ll}
 & \Phi\ inv \\
\equiv & \quad \{\ \text{just proved}\ \} \\
 & \Phi \subseteq \bigcirc\Phi \\
\equiv & \quad \{\ \Phi\cdot\ \text{monotonic; composition of coreflexives is involutive}\ \} \\
 & \Phi \subseteq \Phi \cdot \bigcirc\Phi \\
\Rightarrow & \quad \{\ \text{greatest fixed point induction: } x \leq fx \;\Rightarrow\; x \leq \nu f\ \} \\
 & \Phi \subseteq \Box\Phi
\end{array}$$

Modalities

$$
\begin{array}{ll}
 & \Phi \subseteq \Box\Phi \\
\Rightarrow & \qquad \{ \ \Box\Phi \subseteq f(\Box\Phi) \text{ for } fx = \Phi \cdot \bigcirc x \text{ and gfp induction: } \nu_f \leq f\,\nu_f \ \} \\
 & \Phi \subseteq \Phi \cdot \bigcirc(\Box\Phi) \\
\equiv & \qquad \{ \ \text{shunting of coreflexives} \ \} \\
 & \Phi \subseteq \bigcirc(\Box\Phi) \\
\Rightarrow & \qquad \{ \ \text{monotony; } \Box\Phi \subseteq \Phi \ \} \\
 & \Phi \subseteq \bigcirc\Phi \\
\equiv & \qquad \{ \ \text{definition} \ \} \\
 & \Phi \ \mathit{inv}
\end{array}
$$

Modalities

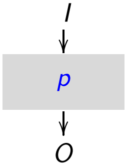
Ex: $\Box\Phi \subseteq \Box\Box\Phi$

$$
\begin{array}{ll}
 & \Box\Phi \subseteq \Box\Box\Phi \\
\equiv & \qquad \{ \text{ definition } \} \\
 & \Box\Phi \subseteq (\nu_X :: \Box\Phi \cdot \bigcirc X) \\
\Leftarrow & \qquad \{ \text{ gfp induction } \} \\
 & \Box\Phi \subseteq \Box\Phi \cdot \bigcirc(\Box\Phi) \\
\equiv & \qquad \{ \Box\Phi \cdot \Phi = \Box\Phi \text{ because } \cap \text{ is composition and } \Box\Phi \subseteq \Phi \} \\
 & \Box\Phi \subseteq \Box\Phi \cdot \Phi \cdot \bigcirc(\Box\Phi) \\
\equiv & \qquad \{ \text{ shunting of coreflexives and } \nu_f \leq f\,\nu_f \} \\
 & \Box\Phi \subseteq \Phi \cdot \bigcirc(\Box\Phi) \ \equiv \ \mathit{true}
\end{array}
$$

Recall: Components as coalgebras

A (generic) component p with input interface I and output interface O

$$p : O \longleftarrow I$$



is a pair

$$\langle u_p \in U_p, \overline{a}_p : \mathsf{B}(U_p \times O)^I \longleftarrow U_p \rangle$$

where

- point u_p is the ‘initial’ or ‘seed’ state.
- B is an arbitrary **strong** monad.

Recall: Components as coalgebras

The semantics of p is the behaviour produced by starting at initial state u_p and **unfolding** over coalgebra $\overline{a}_p$:

$$\llbracket p \rrbracket \quad = \quad (\!|\overline{a}_p|\!)\, u_p$$

$$\begin{tikzcd}
\mathsf{B}(\nu \times O)^I & \nu \arrow[l, "\omega"'] \\
\mathsf{B}(U_p \times O)^I \arrow[u, "\mathsf{B}((\!|\overline{a}_p|\!) \times O)^I"] & U_p \arrow[l, "\overline{a}_p"] \arrow[u, "(\!|\overline{a}_p|\!)"']
\end{tikzcd}$$

That is, an action will not simply produce an output and a continuation state, but a B -structure of such pairs.

Monad B's **unit** (η) and **multiplication** (μ) provide, respectively, a value embedding and a ‘flatten’ operation to unravel nested behavioural annotations.

Invariants as types

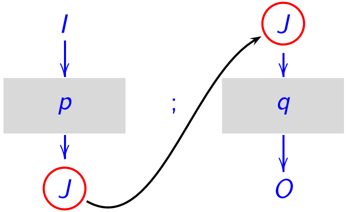
- Each (elementary) component is an aggregation of methods over a shared state space, typically restricted by an (often complex) invariant,
- whose underlying mathematical space can be organised as a category whose
  - objects are coreflexives (representing invariants)
  - arrows

$$f : \Psi \longleftarrow \Phi \ \equiv\ f(\Psi \leftarrow \Phi)f \ \equiv\ f \cdot \Phi \subseteq \Psi \cdot f$$

- Current work: on the structure of this category [paper in preparation]

Combinators preserve invariants

Ex: Pipeline $p\,;q$:



$$a_{p;q} : \mathsf{B}(U_p \times U_q \times O) \longleftarrow U_p \times U_q \times I$$

$$\begin{aligned}
a_{p;q} = U_p \times U_q \times I &\xrightarrow{\cong} U_p \times I \times U_q \xrightarrow{a_p \times \mathrm{id}} \mathsf{B}(U_p \times K) \times U_q \\
&\xrightarrow{\tau_r} \mathsf{B}(U_p \times K \times U_q) \xrightarrow{\cong} \mathsf{B}(U_p \times (U_q \times K)) \\
&\xrightarrow{\mathsf{B}(\mathrm{id} \times a_q)} \mathsf{B}(U_p \times \mathsf{B}(U_q \times O)) \xrightarrow{\mathsf{B}\tau_l} \mathsf{BB}(U_p \times (U_q \times O)) \\
&\xrightarrow{\cong} \mathsf{BB}(U_p \times U_q \times O) \xrightarrow{\mu} \mathsf{B}(U_p \times U_q \times O)
\end{aligned}$$

Combinators preserve invariants

Invariants are preserved $\equiv$ the following is a well-typed arrow:

$$\begin{aligned}
a_{p;q} = \Phi_p \times \Phi_q \times I &\xrightarrow{\cong} \Phi_p \times I \times \Phi_q \xrightarrow{a_p \times \mathrm{id}} \mathsf{B}(\Phi_p \times K) \times \Phi_q \\
&\xrightarrow{\tau_r} \mathsf{B}(\Phi_p \times K \times \Phi_q) \xrightarrow{\cong} \mathsf{B}(\Phi_p \times (\Phi_q \times K)) \\
&\xrightarrow{\mathsf{B}(\mathrm{id} \times a_q)} \mathsf{B}(\Phi_p \times \mathsf{B}(U_q \times O)) \xrightarrow{\mathsf{B}\tau_l} \mathsf{BB}(\Phi_p \times (\Phi_q \times O)) \\
&\xrightarrow{\cong} \mathsf{BB}(\Phi_p \times \Phi_q \times O) \xrightarrow{\mu} \mathsf{B}(\Phi_p \times \Phi_q \times O)
\end{aligned}$$

Combinators preserve invariants

which is an immediate consequence of the (generic) way in which combinators are defined:

- natural transformations are trivial: each polymorphic construction α verifies $\alpha(S \leftarrow R)\alpha$ for all R, S.
- functorial arrows:

$$
\begin{array}{cl}
& \mathsf{F}\,f(\mathsf{F}\,\Phi \leftarrow \mathsf{F}\,\Psi)\mathsf{F}\,f \\
\equiv & \{ \text{ Reynolds combinator } \} \\
& \mathsf{F}\,f \cdot \mathsf{F}\,\Psi \subseteq \mathsf{F}\,\Phi \cdot \mathsf{F}\,f \\
\equiv & \{ \text{ functors } \} \\
& \mathsf{F}\,(f \cdot \Phi) \subseteq \mathsf{F}\,(\Psi \cdot f) \\
\Leftarrow & \{ \text{ monotonicity } \} \\
& f(\Phi \leftarrow \Psi)f
\end{array}
$$

- component actions which, by hypothesis, preserve their own invariants

Summary

Such conceptual tools are applicable at different design levels:

- micro: synthesising component invariants from the individual methods over complex data strucutures
  (cf, Necco & Oliveira & Visser, *Extended Static Checking by Rewriting Pointfree Relations*, 2007
  and Oliveira, *Reinvigorating pen-and-paper proofs in VDM: the pointfree approach*, 2006)
- macro: invariant preservation in the component calculus.
- architectural: global (non structural) "bussiness rules" over components' aggregations

Summary

- Rôle of PF-patterns: clear-cut expression of complex logic structures once expressed in less symbols
- Stress the syntactic aspect of formal reasoning, a kind of "let-the-symbols-do-the-work" style of calculation, often neglected by too much emphasis on domain-specific, semantic concerns.
- Rôle of PF-patterns: much easier to spot synergies among different theories

In particular, a synergy between a relational construct, traditionally employed in explaining and reasoning about parametric polymorphism, and the coalgebraic approach to bisimulations and invariants emerged.