Invariants as coreflexive bisimulations — in a coalgebraic setting

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

[1]U. Minho, Braga
[2]CWI, Amsterdam

IFIP WG2.1 meeting #62
Dec. 2006
Namur, Belgium

Back to basics

Examples of areas of computing which have well-established, widespread theories taught in undergraduate courses:

- Parsers and compilers
- Relational databases
- Automata, labelled transition systems

This time we look into the last one in the list.

Example: Bisimulations

Definition 1 (by R. Milner)

(Well-known — this version taken from the Wikipedia)

A **bisimulation** is a simulation between two LTS such that its converse is also a simulation, where a **simulation** between two LTS $(X, \Lambda, \to_X)$ and $(Y, \Lambda, \to_Y)$ is a relation $R \subseteq X \times Y$ such that, if $(p, q) \in R$, then for all α in Λ, and for all $p' \in S$, $p \xrightarrow{\alpha} p'$ implies that there is a q' such that $q \xrightarrow{\alpha} q'$ and $(p', q') \in R$:

$$
\begin{tikzcd}
p \arrow[d, "\alpha"'] & q \arrow[l, "R"'] \arrow[d, "\alpha"] \\
p' & q' \arrow[l, "R"]
\end{tikzcd}
$$

Typical example of classical, descriptive definition.

Example: Bisimulations

Definition 2 (by Aczel & Mendler):

Given two coalgebras $c : X \to F(X)$ and $d : Y \to F(Y)$ 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

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

Simpler and generic (coalgebraic)

Example: Bisimulations

Definition 3 (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 $y \in Y$.

($Rel(F)(R)$ stands for the relational *lifting* of R via functor F.)

Still coalgebraic, pointwise — somewhat disturbed by the *lifting* construct — see details in [4].

Question

Are all these “the same” definition?

We will check the equivalence of these definitions by PF-transformation

Bisimulations PF-transformed

Let us implode the outermost $\forall$ in Jacobs definition by PF-transformation:

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

Related work

Our PF-definition of bisimulation is similar to that presented by Roland Backhouse for dialgebras [2]: given dialgebra

$\mathsf{F}A \xleftarrow{k} \mathsf{G}A$, relation $A \xleftarrow{R} A$ is a bisimulation of k iff

$$\mathsf{G}R \subseteq k^\circ \cdot \mathsf{F}R \cdot k \qquad \begin{tikzcd} \mathsf{F}A & \mathsf{G}A \arrow[l, "k"'] \\ \mathsf{F}A \arrow[u, "\mathsf{F}R"] & \mathsf{G}A \arrow[l, "k"'] \arrow[u, "\mathsf{G}R"'] \end{tikzcd} \qquad (1)$$

About Reynolds arrow

"Reynolds arrow combinator" is a relation on functions

$$f(R \leftarrow S)g \quad \equiv \quad f \cdot S \subseteq R \cdot g$$

cf. diagram

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

useful in expressing properties of functions — namely *monotonicity*

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

lifting

$$f \stackrel{.}{\leq} g \quad \equiv \quad f(\leq \leftarrow id)f$$

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

Recall database projections

$$\pi_{c,d} R \subseteq S$$

$\equiv$ { definition given in the other talk }

$$c \cdot R \cdot d^\circ \subseteq S$$

$\equiv$ { functions (2nd) "al-djabr" rule }

$$c \cdot R \subseteq S \cdot d$$

$\equiv$ { Reynolds combinator }

$$c(S \leftarrow R)d$$

$\equiv$ { Reynolds combinator }

$$c \cdot R \subseteq S \cdot d$$

$\equiv$ { functions (1st) "al-djabr" rule }

"Al-djabr" rule for projections

$$
\begin{array}{ll}
 & R \subseteq c^\circ \cdot S \cdot d \\
\equiv & \qquad \{ \text{ introduce } \bigcirc \text{ } \} \\
 & R \subseteq \bigcirc_{c,d} S
\end{array}
$$

Thus we get GC:

$$\pi_{c,d} R \subseteq S \quad \equiv \quad R \subseteq \bigcirc_{c,d} S \tag{2}$$

In the other talk we were interested in the lower adjoint ($\pi_{c,d}$); this time we will focus on the the upper adjoint:

$$x\ (\bigcirc_{c,d} S)\ y \quad \equiv \quad (c\ x) S (d\ y)$$

“Al-djabr” rule for projections

At once we get:

- $\pi_{c,d}$ and $\bigcirc_{c,d}$ are monotonic
- Distribution properties (can be generalized to $n > 2$ arguments):

$$\pi_{c,d}(R \cup S) = (\pi_{c,d} R) \cup (\pi_{c,d} S) \tag{3}$$

$$\bigcirc_{c,d}(R \cap S) = (\bigcirc_{c,d} R) \cap (\bigcirc_{c,d} S) \tag{4}$$

- etc

Why does Reynolds arrow matter?

Elegant and manageable PF-properties, eg.

$$
\begin{aligned}
id \leftarrow id &= id && (5)\\
(R \leftarrow S)^\circ &= R^\circ \leftarrow S^\circ && (6)\\
R \leftarrow S \subseteq V \leftarrow U &\Leftarrow R \subseteq V \wedge U \subseteq S && (7)\\
(R \leftarrow V) \cdot (S \leftarrow U) &\subseteq (R \cdot S) \leftarrow (V \cdot U) && (8)
\end{aligned}
$$

as well as

$$
(f \leftarrow g^\circ)h = f \cdot h \cdot g \qquad (9)
$$

recalled from Roland and Kevin Backhouse paper [1] — and earlier.

These are immediately applicable to our PF version of Jacobs' definition. For instance, (5) ensures *id* as bisimulation between a given coalgebra and itself (next slide):

Why Reynolds arrow matters

Calculation

$$
\begin{array}{ll}
& c(\mathsf{F}\ id \leftarrow id)d \\
\equiv & \qquad \{ \ \text{relator F preserves the identity} \ \} \\
& c(id \leftarrow id)d \\
\equiv & \qquad \{ \ (5) \ \} \\
& c\ (id)\ d \\
\equiv & \qquad \{ \ id\ x = x \ \} \\
& c = d
\end{array}
$$

Too simple and obvious, even *without* Reynolds arrow in the play.

What about the equivalence between Jacobs's and Aczel-Mendler's definitions?

Why Reynolds arrow matters

To the set of known rules about *Reynolds arrow*, we add the following:

$$
\begin{array}{c}
\text{pair } (r,\ s) \text{ is a tabulation} \\
\Downarrow \\
(r \cdot s^\circ) \leftarrow (f \cdot g^\circ) \quad = \quad (r \leftarrow f) \cdot (s \leftarrow g)^\circ
\end{array}
\tag{10}
$$

Tabulations

A pair of functions

$$
\begin{tikzcd}
 & C \arrow[dl, "r"'] \arrow[dr, "s"] & \\
A & & B
\end{tikzcd}
$$

form a tabulation iff $\langle r, s \rangle$ is injective, that is,

$$r^\circ \cdot r \cap s^\circ \cdot s = id$$

holds

Why Reynolds arrow matters

Example — we check that π_1 and π_2 form a tabulation:

$$
\begin{array}{ll}
& \pi_1^\circ \cdot \pi_1 \cap \pi_2^\circ \cdot \pi_2 = id \\
\equiv & \quad \{ \text{ go pointwise, where } \cap \text{ is conjunction } \} \\
& (b,a)(\pi_1^\circ \cdot \pi_1)(y,x) \wedge (b,a)(\pi_2^\circ \cdot \pi_2)(y,x) \quad \equiv \quad (b,a) = (y,x) \\
\equiv & \quad \{ \text{ PF-transform rule } (f\ b)R(g\ a) \equiv b(f^\circ \cdot R \cdot g)a \text{ twice } \} \\
& \pi_1(b,a) = \pi_1(y,x) \wedge \pi_2(b,a) = \pi_2(y,x) \quad \equiv \quad (b,a) = (y,x) \\
\equiv & \quad \{ \text{ trivia } \} \\
& b = y \wedge a = x \quad \equiv \quad (b,a) = (y,x)
\end{array}
$$

NB: it is a standard result that every R can be factored in tabulation $R = f \cdot g^\circ$, eg. $R = \pi_1 \cdot \pi_2^\circ$.

Jacobs $\equiv$ Aczel & Mendler

$$
\begin{array}{ll}
 & c(\mathsf{F}\,R \leftarrow R)d \\
\equiv & \qquad \{\ \text{tabulate } R = \pi_1 \cdot \pi_2^\circ\ \ \} \\
 & c(\mathsf{F}(\pi_1 \cdot \pi_2^\circ) \leftarrow (\pi_1 \cdot \pi_2^\circ))d \\
\equiv & \qquad \{\ \text{relator commutes with composition and converse}\ \} \\
 & c(((\mathsf{F}\,\pi_1) \cdot (\mathsf{F}\,\pi_2)^\circ) \leftarrow (\pi_1 \cdot \pi_2^\circ))d \\
\equiv & \qquad \{\ \text{new rule (10)}\ \ \} \\
 & c((\mathsf{F}\,\pi_1 \leftarrow \pi_1) \cdot ((\mathsf{F}\,\pi_2)^\circ \leftarrow \pi_2^\circ))d \\
\equiv & \qquad \{\ \text{converse rule (6)}\ \ \} \\
 & c((\mathsf{F}\,\pi_1 \leftarrow \pi_1) \cdot (\mathsf{F}\,\pi_2 \leftarrow \pi_2)^\circ)d \\
\equiv & \qquad \{\ \text{go pointwise (composition)}\ \ \} \\
 & \langle \exists\ a\ \ ::\ \ c(\mathsf{F}\,\pi_1 \leftarrow \pi_1)a \wedge d(\mathsf{F}\,\pi_2 \leftarrow \pi_2)a \rangle
\end{array}
$$

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.

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

Comments:

- One-slide-long proofs are easy to grasp
- Elegance of the calculation lies in the synergy with Reynolds arrow
- Rule (10) does most of the work — its proof is an example of generic, stepwise PF-reasoning (see this later on)

FDs on bisimulations

FD $d \stackrel{R}{\longrightarrow} c$ holds wherever R is a simple bisimulation from coalgebra d to coalgebra c:

$$
\begin{tikzcd}
\mathsf{F}\,B & B \arrow[l, "c"'] \\
\mathsf{F}\,A \arrow[u, "\mathsf{F}\,R"] & A \arrow[l, "d"] \arrow[u, "R"']
\end{tikzcd}
$$

$$
\begin{array}{ll}
 & c(\mathsf{F}\,R \leftarrow R)d \\
\equiv & \qquad \{ \text{ expand Reynolds combinator } \} \\
 & c \cdot R \subseteq (\mathsf{F}\,R) \cdot d \\
\equiv & \qquad \{ \text{ functions (2nd) “al-djabr” rule } \} \\
 & c \cdot R \cdot d^\circ \subseteq \mathsf{F}\,R \\
\equiv & \qquad \{ \text{ duplicate and take converses } \} \\
 & c \cdot R \cdot d^\circ \subseteq \mathsf{F}\,R \wedge d \cdot R^\circ \cdot c^\circ \subseteq \mathsf{F}\,R^\circ \\
\Rightarrow & \qquad \{ \text{ monotonicity of composition ; relators } \} \\
 & c \cdot R \cdot d^\circ \cdot d \cdot R^\circ \cdot c^\circ \subseteq \mathsf{F}(R \cdot R^\circ)
\end{array}
$$

FDs on bisimulations

$$
\begin{array}{ll}
\Rightarrow & \{\ R \text{ is simple ; F } id = id\ \} \\
& c \cdot R \cdot d^\circ \cdot d \cdot R^\circ \cdot c^\circ \subseteq id \\
\equiv & \{\ \text{FD in kernel's version}\ \} \\
& \ker(d \cdot R^\circ) \subseteq \ker c \\
\equiv & \{\ \text{FD in injectivity preorder version}\ \} \\
& c \leq d \cdot R^\circ
\end{array}
$$

In other words: c can be less injective than d as far as “allowed by” R° (which is injective).
So (implementation) d is allowed to distinguish states which (specification) c does not.

Invariants

Fact $c(\mathsf{F}\, id \leftarrow id)c$ above already tells us that id is a (trivial) F-invariant for coalgebra c. In general:

F-invariants

In this setting, an F-invariant Φ simply is a *coreflexive* bisimulation between a coalgebra and itself:

$$c(\mathsf{F}\,\Phi \leftarrow \Phi)c \tag{11}$$

Invariants bring about *modalities*:

$$c(\mathsf{F}\,\Phi \leftarrow \Phi)c \quad \equiv \quad \Phi \subseteq \underbrace{c^\circ \cdot (\mathsf{F}\,\Phi) \cdot c}_{\bigcirc_c \Phi}$$

cf. the *"next time X holds"* modal operator:

$$\bigcirc_c X \quad \stackrel{\mathrm{def}}{=} \quad c^\circ \cdot (\mathsf{F}\,X) \cdot c$$

Invariants — related work

Elegant PF-definition of a (relational) F-invariant already in Gibbons *et al* *"When is a function a fold or an unfold"?* [3]:

F-invariant

Given relation $\mathsf{F}\,A \xleftarrow{S} A$ (a so-called F-*coalgebra*), we say that relation $A \xleftarrow{R} A$ is an F-*invariant for* S iff

$$S \cdot R \;\subseteq\; \mathsf{F}\,R \cdot S \qquad \begin{tikzcd} A \arrow[r, "S"] & \mathsf{F}\,A \\ A \arrow[u, "R"] \arrow[r, "S"'] \arrow[ur, phantom, "\supseteq"] & \mathsf{F}\,A \arrow[u, "\mathsf{F}\,R"'] \end{tikzcd} \tag{12}$$

Invariants and projections

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

What about Milner's original definition?

Milner's definition is recovered via

- the power-transpose relating binary relations and set-valued functions,

$$f = \Lambda R \quad \equiv \quad R = \in \cdot f \tag{13}$$

where $A \xleftarrow{\in} \mathcal{P}A$ is the membership relation.

- the powerset relator:

$$\mathcal{P}R \quad = \quad (\in \backslash (R \cdot \in)) \cap ((\in^\circ \cdot R)/(\in^\circ)) \tag{14}$$

which unfolds to an elaborate pointwise formula:

$$Y(\mathcal{P}R)X \quad \equiv \quad \langle \forall\ a \ :\ a \in Y :\ \langle \exists\ b \ :\ b \in X :\ a\ R\ b \rangle \rangle \wedge \ldots etc$$

Calculation of Milner's definition

$$c(\mathcal{P}R \leftarrow R)d$$

$\equiv$ { powerset coalgebras uniquely transpose relations }

$$(\Lambda S)(\mathcal{P}R \leftarrow R)(\Lambda U)$$

$\equiv$ { Reynolds }

$$(\Lambda S) \cdot R \subseteq (\mathcal{P}R) \cdot (\Lambda U)$$

$\equiv$ { (14) }

$$(\Lambda S) \cdot R \subseteq ((\in \backslash (R \cdot \in)) \cap ((\in^\circ \cdot R)/(\in^\circ))) \cdot (\Lambda U)$$

$\equiv$ { distribution since ΛU is simple }

$$(\Lambda S) \cdot R \subseteq (\in \backslash (R \cdot \in)) \cdot (\Lambda U) \wedge (\Lambda S) \cdot R \subseteq ((\in^\circ \cdot R)/(\in^\circ)) \cdot (\Lambda U)$$

$\equiv$ { "al-djabr" rule (composition/division) and power transpose }

Calculation of Milner's definition

$$S \cdot R \subseteq R \cdot U \wedge (\Lambda S) \cdot R \subseteq ((\in^\circ \cdot R)/(\in^\circ)) \cdot (\Lambda U)$$

$$\equiv \qquad \{ \text{ take converses ; "al-djabr" (functions) } \}$$

$$S \cdot R \subseteq R \cdot U \wedge (\Lambda U) \cdot R^\circ \subseteq ((\in^\circ \cdot R)/(\in^\circ))^\circ \cdot (\Lambda S)$$

$$\equiv \qquad \{ \text{ divisions and power transpose } \}$$

$$S \cdot R \subseteq R \cdot U \wedge U \cdot R^\circ \subseteq R^\circ \cdot S$$

Obs:

- Matteo Vaccari [6] infers the same by direct PF-transforming Milner's original definition
- We obtain the same result by instantiating Jacobs' definition to the power relator.

Follow up

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

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

which shrinks to a greatest (post)fix-point

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

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

- Properties calculated by PF-fixpoint calculation
- etc (currently writing a paper on this)

Summary

- Pointfree / pointwise dichotomy: PF is for reasoning in-the-large, PW is for the small
- Back to basics: need for computer science theory "refactoring"
- Rôle of PF-patterns: clear-cut expression of complex logic structures once expressed in less symbols
- Rôle of PF-patterns: much easier to spot synergies among different theories
- Coalgebraic approach in a relational setting: a win-win approach while putting together coalgebras (functions) + relators (relations).
- Also related: proof obligations on state invariants in VDM discharged by PF- calculation [5].

Annex — Calculation of (10)

Still need to calculate rule

$$\text{pair } (r,\ s) \text{ is a tabulation}$$
$$\Downarrow$$
$$(r \cdot s^\circ) \leftarrow (f \cdot g^\circ) \;=\; (r \leftarrow f) \cdot (s \leftarrow g)^\circ$$

Our approach structures itself in a number of (generic) auxiliary results. First of all, and thanks to (8), only the "fission" part of the consequent of (10)

$$(r \cdot s^\circ) \leftarrow (f \cdot g^\circ) \subseteq (r \leftarrow f) \cdot (s \leftarrow g)^\circ$$

calls for evidence which, for all suitably typed functions c and d, equivales

$$c \cdot f \cdot g^\circ \subseteq r \cdot s^\circ \cdot d \;\;\Rightarrow\;\; \langle \exists\ k \ ::\ c(r \leftarrow f)k \wedge d(s \leftarrow g)k \rangle$$

$$c \cdot f \cdot g^\circ \subseteq r \cdot s^\circ \cdot d \quad \Rightarrow \quad \langle \exists\ k\ ::\ c(r \leftarrow f)k \wedge d(s \leftarrow g)k \rangle$$

$$\equiv \qquad \{ \text{ "al-djabr" and Reynolds arrow } \}$$

$$c \cdot f \subseteq r \cdot s^\circ \cdot d \cdot g \quad \Rightarrow \quad \langle \exists\ k\ ::\ c \cdot f = r \cdot k \wedge d \cdot g = s \cdot k \rangle$$

This, in turn, is an instance of

$$x \subseteq r \cdot s^\circ \cdot y \quad \Rightarrow \quad \langle \exists\ k\ ::\ x = r \cdot k \wedge y = s \cdot k \rangle$$

$$\equiv \qquad \{ \text{ "al-djabr" and split-universal, followed by split-fusion } \}$$

$$x \cdot y^\circ \subseteq r \cdot s^\circ \quad \Rightarrow \quad \langle \exists\ k\ ::\ \langle x, y \rangle = \langle r, s \rangle \cdot k \rangle \tag{15}$$

for $x, y := c \cdot f, d \cdot g$, cf. diagram:

$$\begin{tikzcd}
A \arrow[d, "c"'] & B \arrow[l, "f"'] \arrow[r, "g"] \arrow[dl, "x"'] \arrow[d, "k"'] \arrow[dr, "y"] & C \arrow[d, "d"] \\
D & E \arrow[l, "r"'] \arrow[r, "s"] & F
\end{tikzcd}$$

On function-split fission

The righthand side of implication (15) is an assertion of *split-fission*, an instance of function-fission in general. This can be shown to lead to two concerns:

$$\begin{tikzcd}
A \arrow[d, "c"'] & B \arrow[l, "f"'] \arrow[r, "g"] \arrow[dl, "x"'] \arrow[d, "k"] \arrow[dr, "y"] & C \arrow[d, "d"] \\
D & E \arrow[l, "r"'] \arrow[r, "s"'] & F
\end{tikzcd}$$

- the image of $\langle x, y \rangle$ must be at most the image of $\langle r, s \rangle$ — $\langle r, s \rangle$ "at least as surjective as" $\langle x, y \rangle$
- $\langle r, s \rangle$ must be injective "relative" to $\langle x, y \rangle$.

Concerning the former, we are happy to realize that it exactly matches the antecendent of (15):

$$\begin{array}{ll}
& \mathrm{img}\,\langle x, y \rangle \subseteq \mathrm{img}\,\langle r, s \rangle \\
\equiv & \quad \{ \text{ split image transform, see below } \} \\
& x \cdot y^{\circ} \subseteq r \cdot s^{\circ}
\end{array}$$

On function-split fission

Concerning the latter, we go stronger than required in forcing $\langle r, s \rangle$ to be *everywhere*-injective:

$$
\begin{array}{cl}
 & \ker \langle r, s \rangle \subseteq id \\
\equiv & \{ \text{ kernels of splits ; kernels of functions are reflexive } \} \\
 & \ker r \cap \ker s = id
\end{array}
$$

This is equivalent to saying that pair r, s is a tabulation: thus the side condition of (10).

$\square$

On function fission

Divisibility relation on functions

$f \setminus g$ iff there is a k such that

$$g = f \cdot k \tag{16}$$

holds. □

Of course, $g \setminus g$ holds ($k = id$) and $id \setminus g$ holds ($k = g$).

In general, to establish $f \setminus g$ it is enough to find a *functional* solution k to equation (16).
Clearly, a **relational** upperbound for k always exists, $f^\circ \cdot g$, cf.

On function fission

Divisibility relation on functions

$f \setminus g$ iff there is a k such that

$$g = f \cdot k \tag{16}$$

holds. □

Of course, $g \setminus g$ holds ($k = id$) and $id \setminus g$ holds ($k = g$).

In general, to establish $f \setminus g$ it is enough to find a *functional* solution k to equation (16).
Clearly, a **relational** upperbound for k always exists, $f^{\circ} \cdot g$, cf.

On function fission

$$
\begin{array}{cl}
& g = f \cdot k \\
\equiv & \qquad \{ \ \text{equality of functions} \ \} \\
& f \cdot k \subseteq g \\
\equiv & \qquad \{ \ \text{"al-djabr"} \ \} \\
& k \subseteq f^{\circ} \cdot g
\end{array}
$$

Let us find conditions for such a (maximal) solution $f^{\circ} \cdot g$ to be a function: it must be entire

$$
\begin{array}{cl}
& id \subseteq (f^{\circ} \cdot g)^{\circ} \cdot f^{\circ} \cdot g \\
\equiv & \qquad \{ \ \text{"al-djabr" ; definition of image} \ \} \\
& \mathrm{img}\, g \subseteq \mathrm{img}\, f
\end{array}
$$

On function fission

and simple:

$$
\begin{array}{ll}
 & f^{\circ} \cdot g \cdot (f^{\circ} \cdot g)^{\circ} \subseteq id \\
\equiv & \{ \text{ converses } \} \\
 & f^{\circ} \cdot g \cdot g^{\circ} \cdot f \subseteq id
\end{array}
$$

So, for f divides g wherever

- f at least as surjective as g and
- f "injective within the image (range) of" g.

Last condition back to points: for all a, b

$$\langle \exists\ c\ ::\ f\ a = g\ c = f\ b \rangle \ \Rightarrow\ a = b$$

Images of splits

Generic fact for calculating with images of splits:

$$\mathrm{img}\,\langle R, S\rangle \subseteq \mathrm{img}\,\langle U, V\rangle \equiv R \cdot S^{\circ} \subseteq U \cdot V^{\circ} \qquad (17)$$

Calculation:

$$\begin{array}{cl}
 & \mathrm{img}\,\langle R, S\rangle \subseteq \mathrm{img}\,\langle U, V\rangle \\
\equiv & \qquad \{ \text{ switch to conditions } \} \\
 & \langle R, S\rangle \cdot !^{\circ} \subseteq \langle U, V\rangle \cdot !^{\circ} \\
\equiv & \qquad \{ \text{ ``split twist'' rule (18) } \} \\
 & \langle R, !\rangle \cdot S^{\circ} \subseteq \langle U, !\rangle \cdot V^{\circ} \\
\equiv & \qquad \{ \text{ (19) thanks to !-natural } \} \\
 & \langle id, !\rangle \cdot R \cdot S^{\circ} \subseteq \langle id, !\rangle \cdot U \cdot V^{\circ} \\
\equiv & \qquad \{ \ \langle id, f\rangle \text{ is injective for any } f\text{, thus left-cancellable } \} \\
 & R \cdot S^{\circ} \subseteq U \cdot V^{\circ}
\end{array}$$

Again useful

"Split twist" rule:

$$\langle R, S\rangle \cdot T \subseteq \langle U, V\rangle \cdot X \quad \equiv \quad \langle R, T^\circ\rangle \cdot S^\circ \subseteq \langle U, X^\circ\rangle \cdot V^\circ \tag{18}$$

Conditional split-fusion:

$$\langle R, S\rangle \cdot T = \langle R \cdot T, S \cdot T\rangle \quad \Leftarrow \quad R \cdot (\mathrm{img}\ T) \subseteq R \vee S \cdot (\mathrm{img}\ T) \subseteq S \tag{19}$$

K. Backhouse and R.C. Backhouse.
Safety of abstract interpretations for free, via logical relations and Galois connections.
SCP, 15(1–2):153–196, 2004.

R.C. Backhouse and P.F. Hoogendijk.
Final dialgebras: From categories to allegories.
Informatique Theorique et Applications, 33(4/5):401–426, 1999.
Presented at Workshop on Fixed Points in Computer Science, Brno, August 1998.

Jeremy Gibbons, Graham Hutton, and Thorsten Altenkirch.
When is a function a fold or an unfold?, 2001.
WGP, July 2001 (slides).

Bart Jacobs.
Introduction to Coalgebra. Towards Mathematics of States and Observations.

Draft Copy. Institute for Computing and Information Sciences, Radboud University Nijmegen, P.O. Box 9010, 6500 GL Nijmegen, The Netherlands.

J.N. Oliveira.
Reinvigorating pen-and-paper proofs in VDM: the pointfree approach 2006.
Presentation at the
Third OVERTURE Workshop: Newcastle, UK, 27-28 November 2006
Slides available from the author's website.

M. Vaccari.
Calculational derivation of circuits, 1998.
PhD thesis, Dipartimento di Informatica, Universita degli Studi di Milano.