Components As Coalgebras

Luís S. Barbosa
DI - UNIV. MINHO

UNIVERSITY OF TARTU
December 2003

Aims & Context

We consider state-based software components (as they arise in the model-oriented approach to systems design)

- internal state space,
- possibility of interaction with other components during the overall computation,
- input and output observation universes to ensure flow of data.

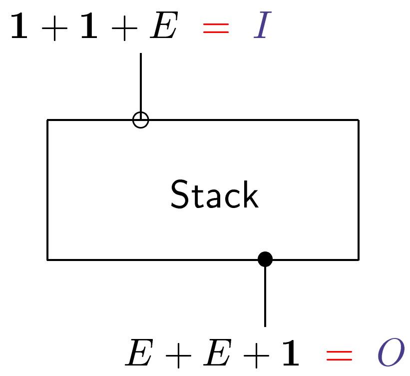
framed as generalised Mealy machines, in the sense that the strict deterministic discipline is relaxed to capture more complex behaviours

Aims & Context

- To look for a semantical model for such components, entailing a notion of observational equivalence and explicitly parametrized by a behaviour model
- ... and a calculus to reason about them in an essentially equational, pointfree way as one gets used to in functional programming
- related work includes [Walters, Sabadini, Katis 97,00], [Abramsky], [Krstic, Launchbury, Pavlovic 01]

Motivation

$$\begin{cases} \mathsf{pop}: & \mathbf{1} \longrightarrow E \\ \mathsf{top}: & \mathbf{1} \longrightarrow E \\ \mathsf{push}: & E \longrightarrow \mathbf{1} \end{cases}$$



$$[\![\mathsf{Stack}]\!] : U \longrightarrow (U \times O + 1)^I$$

... i.e., a *coalgebra* of signature $U \longrightarrow \mathsf{T}\ U$ involving transition 'shape'

$$\mathsf{T} \ = \ ((\mathsf{Id} \times O) + \mathbf{1})^I$$

Going Generic

Replace $X + \mathbf{1}$ by an arbitrary strong monad B in

$$\mathsf{T} \;=\; \mathsf{B}(\mathsf{Id} \times O)^I$$

Examples

- Partiality: $\mathsf{B} = \mathsf{Id} + \mathbf{1}$
- Non determinism: $\mathsf{B} = \mathcal{P}$
- Ordered non determinism: $\mathsf{B} = \mathsf{Id}^*$
- Monoidal labelling: $\mathsf{B} = \mathsf{Id} \times M$, with M a monoid.
- 'Metric' non determinism: $\mathsf{B} = \mathsf{Bag}_M$ based on $\langle M, \oplus, \otimes \rangle$, where $\otimes$ distributes over $\oplus$, both defining Abelian monoids over M.

Going Generic

In particular

- Cost components for $M = \langle \mathbb{N}, +, \times \rangle$
(assigning a cost or performance measure to each alternative)
- Probabilistic components for $M = \langle [0, 1], \mathsf{min}, \times \rangle$ st

$$\sum (\mathcal{P}\pi_2)\, m = 1 \quad \text{for each } \; m \in \mathsf{Bag}_M$$

Going Generic

The structure of the behaviour model is detailed as

$$\langle \mathsf{B}, \eta, \mu \rangle$$

where

$$\begin{aligned} \eta &: \mathsf{Id} \Longrightarrow \mathsf{B} && \text{(to make a behavioural annotation)} \\ \mu &: \mathsf{BB} \Longrightarrow \mathsf{B} && \text{(to flatten nested annotations)} \end{aligned}$$

with right and left strength for context handling:

$$\begin{aligned} \tau_r^{\mathsf{T}} &: \mathsf{T} \times - \Longrightarrow \mathsf{T}(\mathsf{Id} \times -) \\ \tau_l^{\mathsf{T}} &: - \times \mathsf{T} \Longrightarrow \mathsf{T}(- \times \mathsf{Id}) \end{aligned}$$

Going Generic

Furthermore, the Kleisli compositions

$$\delta r_{I,J} = \tau_{r_{I,J}} \bullet \tau_{l_{\mathsf{B}I,J}} \quad \text{and} \quad \delta l_{I,J} = \tau_{l_{I,J}} \bullet \tau_{r_{I,\mathsf{B}J}}$$

map

$$\mathsf{B}I \times \mathsf{B}J \ \text{ to } \ \mathsf{B}(I \times J)$$

specifying a sort of sequential composition of B-computations

B is a commutative monad if $\delta r_{I,J} = \delta l_{I,J}$

... plus a handful of equational laws

Behavioural Casting

- Not all casting is possible: the target monad should have 'enough' structure to do the embeding.
- Every monad morphism $h : \mathsf{B} \Longrightarrow \mathsf{B}'$ lifts to a natural transformation

$$\mathsf{T}^h : \mathsf{T}^{\mathsf{B}} \Longrightarrow \mathsf{T}^{\mathsf{B}'}$$

  transforming each T^{B} component into a $\mathsf{T}^{\mathsf{B}'}$ one, over the same state space
- Example: for $\mathsf{B} = \mathsf{Id} + \mathbf{1}$ and $\mathsf{B}' = \mathcal{P}$: $h = [\mathsf{sing}, \underline{\emptyset}]$

Variants

$$\boxed{\mathsf{T}^{\mathsf{B}} = O'^{I'} \times \mathsf{B}(\mathsf{Id} \times O)^I}$$

- 'Functional' components (Mealy Machines for $\mathsf{B} = \mathsf{Id}$ [Mea57]):

$$\mathsf{T}^{\mathsf{B}}_{I,O} = \mathsf{B}(\mathsf{Id} \times O)^I$$

- 'Silent' components: $\mathsf{T}^{\mathsf{B}}_{I,O} = O^I \times \mathsf{B}$
- 'Action' components: $\mathsf{T}^{\mathsf{B}}_{O} = \mathsf{B}(\mathsf{Id} \times O)$
- 'Object' components (Moore Machines for $\mathsf{B} = \mathsf{Id}$ [Moo66]):

$$\mathsf{T}^{\mathsf{B}}_{I,O} = O \times \mathsf{B}^I$$

Components as Arrows

functions	$f : I \longrightarrow O$	$f \in O^I$
components	$p : I \longrightarrow O$	$p \in \cdots$

Components form a bicategory Cp_{B}:

- objects: interfaces (sets I, J, O ...)
- arrows: components, *i.e.* pointed concrete coalgebras for T in Set

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

Components as Arrows

- 2-cells (*cf.*, each hom-set $\mathsf{Cp}(I, O)$ becomes a category):

$$h : \langle u_p, \overline{a}_p \rangle \longrightarrow \langle u_q, \overline{a}_q \rangle$$

$$\begin{tikzcd} U_p \arrow[r, "\overline{a}_p"] \arrow[d, "h"'] & \mathsf{T}\ U \arrow[d, "\mathsf{T}\ h"] \\ U_q \arrow[r, "\overline{a}_q"'] & \mathsf{T}\ U_q \end{tikzcd}$$

$$\overline{a}_q \cdot h = \mathsf{T}\ h \cdot \overline{a}_p \qquad \text{comorphism condition}$$

$$h\ u_p = u_q \qquad \text{seed preservation}$$

Composition is inherited from Set and the identity $1_p : p \longrightarrow p$ is defined as the identity id_{U_p} on its carrier

Components as Arrows

- Composition: pipelining given by a functor (for each $\langle I, K, O\rangle$)

$$;_{I,K,O} : \mathsf{Cp}(I,K) \times \mathsf{Cp}(K,O) \longrightarrow \mathsf{Cp}(I,O)$$

$$\begin{array}{rrcl} \text{on arrows} & h\,;k & = & h \times k \\ \text{on objects} & p\,;q & = & \langle \langle u_p, u_q\rangle \in U_p \times U_q, \overline{a}_{p;q}\rangle \end{array}$$

where
$$\begin{aligned} a_{p;q} = {} & U_p \times U_q \times I \xrightarrow{\mathsf{xr}} U_p \times I \times U_q \xrightarrow{a_p \times \mathsf{id}} \mathsf{B}(U_p \times K) \times U_q \\ & \xrightarrow{\tau_r} \mathsf{B}(U_p \times K \times U_q) \xrightarrow{\mathsf{B}(\mathsf{a}\cdot\mathsf{xr})} \mathsf{B}(U_p \times (U_q \times K)) \\ & \xrightarrow{\mathsf{B}(\mathsf{id} \times a_q)} \mathsf{B}(U_p \times \mathsf{B}(U_q \times O)) \xrightarrow{\mathsf{B}\tau_l} \mathsf{BB}(U_p \times (U_q \times O)) \\ & \xrightarrow{\mathsf{BBa}^\circ} \mathsf{BB}(U_p \times U_q \times O) \xrightarrow{\mu} \mathsf{B}(U_p \times U_q \times O) \end{aligned}$$

Components as Arrows

- Identities: given by a functor (for each K)

$$\mathsf{copy}_K : \mathbf{1} \longrightarrow \mathsf{Cp}(K,K)$$

whose action on objects is the constant component $\langle * \in \mathbf{1}, \overline{a}_{\mathsf{copy}_K}\rangle$

$$a_{\mathsf{copy}_K} = \eta_{\mathbf{1}\times K}$$

Clearly,

$$\begin{array}{c} \mathsf{copy}_I\,;p \sim p \sim p\,;\mathsf{copy}_O \\ (p\,;q)\,;r \sim p\,;(q\,;r) \end{array}$$

Proof Outline

- Each $\mathsf{Cp}(I,O)$ is the category of $\mathsf{B}(\mathsf{Id}\times O)^I$-coalgebras over Set
- Verify both $;_{I,K,O}$ and copy_K are functors, for each I, K and O. (In particular, given $h : p \longrightarrow p'$ and $k : q \longrightarrow q'$, $h \,;\, k$ is actually a $\mathsf{Cp}(I,O)$-morphism from $p \,;\, q$ to $p' \,;\, q'$)
- Check the existence of natural isomorphisms encoding $;$ associativity and unit:

$$\begin{aligned}
\mathsf{a}_{I,K,L,O} &: ;_{I,K,O} \circ (;_{K,L,O} \times \mathsf{Id}) \Longrightarrow ;_{I,L,O} \circ (\mathsf{Id} \times ;_{I,K,L}) \\
\mathsf{r}_{I,O} &: ;_{I,I,O} \circ (\mathsf{copy}_I \times \mathsf{Id}) \Longrightarrow \mathsf{Id} \\
\mathsf{l}_{I,O} &: ;_{I,O,O} \circ (\mathsf{Id} \times \mathsf{copy}_O) \Longrightarrow \mathsf{Id}
\end{aligned}$$

Behaviour

From $p = \langle u_p, \overline{a}_p \rangle$, behaviour is computed by coinductive extension (*i.e.*,as u_p anamorphic image):

$$[\![(\overline{a}_p)]\!] u_p$$

Bh — a category of behaviours:

- objects: interfaces (sets I, J, O ...)
- arrows: behaviours, *i.e.* elements of $\nu_{I,O}$, the carrier of the final coalgebra $\omega_{I,O}$ for functor T
- composition and identities:

$$\begin{aligned}
;_{I,K,O}{}^{\mathsf{Bh}} &= [\![(\omega_{I,K} \,;\, \omega_{K,O})]\!] \\
\mathsf{copy}_K{}^{\mathsf{Bh}} &= [\![(\overline{a}_{\mathsf{copy}_K})]\!] \ *
\end{aligned}$$

Behaviour

Clearly, Bh is a category:

- $$b \,;\, \mathsf{copy}_O \;=\; [\![(\omega_{I,O} \,;\, \mathsf{copy}_O)]\!] \langle b, * \rangle \;=\; [\![(\omega_{I,O})]\!] b \;=\; b$$

- $$\begin{aligned}(b\,;c)\,;d &= [\![((\omega_{I,K}\,;\omega_{K,L})\,;\omega_{L,O})]\!]\langle\langle b,c\rangle,d\rangle\\ &= [\![(\omega_{I,K}\,;(\omega_{K,L}\,;\omega_{L,O}))]\!]\langle b,\langle c,d\rangle\rangle\\ &= b\,;(c\,;d)\end{aligned}$$

Behaviour

... and $[\![\ ]\!]$ a 2-functor from Cp to Bh:

- $[\![\mathsf{copy}_K{}^{\mathsf{Cp}}]\!] \;=\; \mathsf{copy}_K{}^{\mathsf{Bh}}$ (trivial)

- $$\begin{aligned}[\![(p\,;^{\mathsf{Cp}} q)]\!] &= [\![(\overline{a}_{p;q})]\!]\langle u_p,u_q\rangle\\ &= [\![(\omega_{I,K}\,;\omega_{K,O})]\!]\cdot([\![(\overline{a}_p)]\!]\times[\![(\overline{a}_q)]\!])\langle u_p,u_q\rangle\\ &= ;^{\mathsf{Bh}}\cdot([\![(\overline{a}_p)]\!]\times[\![(\overline{a}_q)]\!])\langle u_p,u_q\rangle\\ &= ;^{\mathsf{Bh}}\langle[\![(\overline{a}_p)]\!]u_p,[\![(\overline{a}_q)]\!]u_q\rangle\\ &= [\![(p)]\!]\;;^{\mathsf{Bh}}[\![(q)]\!]\end{aligned}$$

Behaviour

Bh mirrors whatever structure Cp possesses:

- It arises by quotienting Cp by the greatest T^{B}-bisimulation.
- but, because final coalgebras are fully abstract with respect to bisimulation, the bicategorical structure collapses
- ... making laws to hold 'on the nose', *e.g.*,

$$\mathsf{copy}_I \,;\, b \;=\; b \;=\; b \,;\, \mathsf{copy}_O$$

$$(b \,;\, c) \,;\, d \;=\; b \,;\, (c \,;\, d)$$

component calculus *vs* (typed) 'process' algebra

Function Lifting

$$\ulcorner f \urcorner \;=\; \langle * \in \mathbf{1}, \overline{a}_{\ulcorner f \urcorner} \rangle \qquad \text{for } \; f : A \longrightarrow B$$

$$a_{\ulcorner f \urcorner} = \mathbf{1} \times A \xrightarrow{\mathsf{id} \times f} \mathbf{1} \times B \xrightarrow{\eta_{(\mathbf{1} \times B)}} \mathsf{B}(\mathbf{1} \times B)$$

i.e., a coalgebra over $\mathbf{1}$ given by the currying of

Laws

$$\ulcorner f \cdot g \urcorner \sim \ulcorner g \urcorner \,;\, \ulcorner f \urcorner$$

$$\ulcorner \mathsf{id}_I \urcorner \sim \mathsf{copy}_I$$

Function Lifting

- $?_I : \emptyset \longrightarrow I$ keeps its naturality as

$$\begin{tikzcd} I \arrow[r, "p"] & O \\ \emptyset \arrow[u, "\ulcorner ?_I \urcorner"] \arrow[ur, "\ulcorner ?_O \urcorner"'] & \end{tikzcd}$$

  commutes up to bisimulation

- Therefore

$$\ulcorner ?_I \urcorner \, ; p \;\sim\; \ulcorner ?_O \urcorner$$

  and $\emptyset$ is initial in Bh

Proof Outline

Check $!_{\mathbf{1} \times U_p} : \mathbf{1} \times U_p \longrightarrow \mathbf{1}$ is an arrow in Cp from $\ulcorner ?_I \urcorner \, ; p$ to $?_O$:

$$\begin{tikzcd} \mathbf{1} \times U_p \arrow[r, "!"] \arrow[d, "\overline{a}_{\ulcorner ?_I \urcorner ; p}"'] & \mathbf{1} \arrow[d, "\overline{a}_{\ulcorner ?_O \urcorner}"] \\ \mathsf{B}((\mathbf{1} \times U_p) \times O)^{\emptyset} \arrow[r, "\mathsf{B}(! \times \mathsf{id})^{\emptyset}"] & \mathsf{B}(\mathbf{1} \times O)^{\emptyset} \end{tikzcd}$$

Thus,

$$\begin{aligned} & \mathsf{B}(! \times \mathsf{id})^{\emptyset} \cdot \overline{a}_{\ulcorner ?_I \urcorner ; p} \\ = \quad & \{ \; f = ?_B \cdot \mathsf{zr}_A \text{ for any function } f : A \times \emptyset \longrightarrow B \; \} \\ & \mathsf{B}(! \times \mathsf{id})^{\emptyset} \cdot \overline{?_{\mathsf{B}((\mathbf{1} \times U_p) \times O)} \cdot \mathsf{zr}_{(\mathbf{1} \times U_p)}} \\ = \quad & \{ \; \text{exponential absorption} \; \} \end{aligned}$$

Proof Outline

$$
\begin{aligned}
& \overline{\mathsf{B}(! \times \mathrm{id}) \cdot ?_{\mathsf{B}((\mathbf{1}\times U_p)\times O)} \cdot \mathsf{zr}_{(\mathbf{1}\times U_p)}} \\
= & \quad \{ \ \text{initiality} \ \} \\
& \overline{?_{\mathsf{B}(\mathbf{1}\times O)} \cdot \mathsf{zr}_{\mathbf{1}\times U_p}} \\
= & \quad \{ \ \mathsf{zr} \text{ natural} \ \} \\
& \overline{?_{\mathsf{B}(\mathbf{1}\times O)} \cdot \mathsf{zr}_{\mathbf{1}} \cdot (!_{(\mathbf{1}\times U_p)} \times \mathrm{id})} \\
= & \quad \{ \ f = ?_B \cdot \mathsf{zr}_A \text{ for any } f : A \times \emptyset \longrightarrow B \text{ and } \ulcorner f \urcorner \text{ definition} \ \} \\
& \overline{a_{\ulcorner ?_O \urcorner} \cdot (!_{(\mathbf{1}\times U_p)} \times \mathrm{id})} \\
= & \quad \{ \ \text{exponential fusion} \ \} \\
& \overline{a}_{\ulcorner ?_O \urcorner} \cdot !_{\mathbf{1}\times U_p}
\end{aligned}
$$

Function Lifting

- On the other hand, naturality is lost in the case of $!_I : I \longrightarrow \mathbf{1}$ as

$$
\begin{tikzcd}
I \arrow[r, "p"] \arrow[d, "\ulcorner !_I \urcorner"'] & O \arrow[dl, "\ulcorner !_O \urcorner"] \\
\mathbf{1} &
\end{tikzcd}
$$

  fails to commute for non trivial B

- Isomorphisms, split monos and split epis lift to Cp as, respectively, isomorphisms, split monos and split epis

Proof Outline

- Let $f : A \longrightarrow B$ be a Set-isomorphism. Then

$$\ulcorner f \urcorner ; \ulcorner f^\circ \urcorner \sim \ulcorner f^\circ \cdot f \urcorner \sim \ulcorner \mathsf{id}_A \urcorner \sim \mathsf{copy}_A$$

Conversely,

$$\ulcorner f^\circ \urcorner ; \ulcorner f \urcorner \sim \ulcorner f \cdot f^\circ \urcorner \sim \ulcorner \mathsf{id}_B \urcorner \sim \mathsf{copy}_B$$

- For f a split epi (respectively, a split mono) consider the first (respectively, second) part of the proof above, taking f° as a section (respectively, a retraction) of f.
(Recall: every epi and every mono with non empty source split in Set)

Wiring

- Wires are components over $\mathbf{1}$ defined from identities and structural properties of the underlying category, *e.g.*,
- Liftings of canonical isomorphisms — a, s, l or r
which leads bisimilarity up to an isomorphic rearranging of the interface
- Liftings of embeddings, projections, codiagonals and diagonals:

$$\nabla : I + I \longrightarrow I = [\mathsf{id}_I, \mathsf{id}_I]$$
$$\Delta : O \longrightarrow O \times O = \langle \mathsf{id}_O, \mathsf{id}_O \rangle$$

used to merge input and replicate output types.

- Example: $\ulcorner \nabla \urcorner ; p ; \ulcorner \Delta \urcorner : I + I \longrightarrow O \times O$

Wrapping

pre- and post-composition of a component with Cp-lifted functions encapsulated in an unique combinator defined as a functor

$$-[f,g] : \mathsf{Cp}(I,O) \longrightarrow \mathsf{Cp}(I',O')$$

which is the identity on morphisms and maps a component $\langle u_p, \overline{a}_p \rangle$ into $\langle u_p, \overline{a}_{p[f,g]} \rangle$, where

$$a_{p[f,g]} = U_p \times I' \xrightarrow{\mathsf{id}\times f} U_p \times I \xrightarrow{a_p} \mathsf{B}(U_p \times O) \xrightarrow{\mathsf{B}(\mathsf{id}\times g)} \mathsf{B}(U_p \times O')$$

$\boxed{\text{Laws}}$

$$(p[f,g])[f',g'] \sim p[f \cdot f', g' \cdot g]$$
$$p[f,g] \sim \ulcorner f \urcorner ; p ; \ulcorner g \urcorner$$

Proof Outline

Check successively that

$$p[f,\mathsf{id}] \sim \ulcorner f \urcorner ; p$$
$$p[\mathsf{id},g] \sim p ; \ulcorner g \urcorner$$

to conclude, by the first wrapping law, that

$$p[f,g] = (p[f,\mathsf{id}])[\mathsf{id},g] \sim (\ulcorner f \urcorner ; p)[\mathsf{id},g] \sim \ulcorner f \urcorner ; p ; \ulcorner g \urcorner$$

for example:

Proof Outline

$$\mathsf{B}(\mathsf{l} \times \mathsf{id}) \cdot a_{p;\ulcorner g \urcorner}$$

$=$ { ; and function lifting definitions }

$$\mathsf{B}(\mathsf{l} \times \mathsf{id}) \cdot \mu \cdot \mathsf{BBa}^\circ \cdot \mathsf{B}\tau_l \cdot \mathsf{B}(\mathsf{id} \times \eta) \cdot \mathsf{B}(\mathsf{id} \times (\mathsf{id} \times g)) \cdot \mathsf{B}(\mathsf{a} \cdot \mathsf{xr}) \cdot \tau_r \cdot (a_p \times \mathsf{id}$$

$=$ { η strong and natural }

$$\mathsf{B}(\mathsf{l} \times \mathsf{id}) \cdot \mu \cdot \mathsf{B}\eta \cdot \mathsf{Ba}^\circ \cdot \mathsf{B}(\mathsf{id} \times (\mathsf{id} \times g)) \cdot \mathsf{B}(\mathsf{a} \cdot \mathsf{xr}) \cdot \tau_r \cdot (a_p \times \mathsf{id}) \cdot \mathsf{xr}$$

$=$ { monad unit }

$$\mathsf{B}(\mathsf{l} \times \mathsf{id}) \cdot \mathsf{Ba}^\circ \cdot \mathsf{B}(\mathsf{id} \times (\mathsf{id} \times g)) \cdot \mathsf{B}(\mathsf{a} \cdot \mathsf{xr}) \cdot \tau_r \cdot (a_p \times \mathsf{id}) \cdot \mathsf{xr}$$

$=$ { a natural isomorphism }

$$\mathsf{B}(\mathsf{l} \times \mathsf{id}) \cdot \mathsf{B}(\mathsf{id} \times g) \cdot \mathsf{Bxr} \cdot \tau_r \cdot (a_p \times \mathsf{id}) \cdot \mathsf{xr}$$

$=$ { functors }

Proof Outline

$$\mathsf{B}(\mathsf{id} \times g) \cdot \mathsf{B}(\mathsf{l} \times \mathsf{id}) \cdot \mathsf{Bxr} \cdot \tau_r \cdot (a_p \times \mathsf{id}) \cdot \mathsf{xr}$$

$=$ { routine: $(\mathsf{l} \times \mathsf{id}) \cdot \mathsf{xr} = \mathsf{l}$ }

$$\mathsf{B}(\mathsf{id} \times g) \cdot \mathsf{Bl} \cdot \tau_r \cdot (a_p \times \mathsf{id}) \cdot \mathsf{xr}$$

$=$ { τ_r unit }

$$\mathsf{B}(\mathsf{id} \times g) \cdot \mathsf{l} \cdot (a_p \times \mathsf{id}) \cdot \mathsf{xr}$$

$=$ { l natural and $\mathsf{l} \cdot \mathsf{xr} = \mathsf{l} \times \mathsf{id}$ }

$$\mathsf{B}(\mathsf{id} \times g) \cdot a_p \cdot (\mathsf{l} \times \mathsf{id})$$

$=$ { wrapping definition }

$$a_{p[\mathsf{id},g]} \cdot (\mathsf{l} \times \mathsf{id})$$

Special Components

- $\mathsf{nil} : \emptyset \longrightarrow \emptyset$

$$\mathsf{nil} \;=\; \mathsf{inert}_{\emptyset} \quad \text{where} \;\; \mathsf{inert}_A \;=\; \ulcorner ?_A \urcorner$$

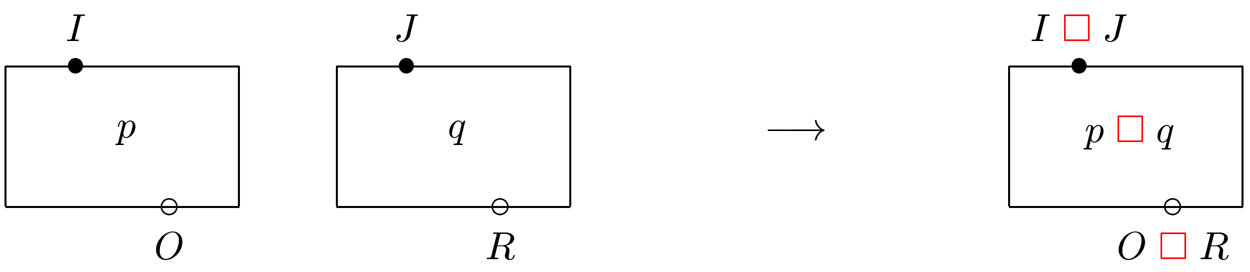
- $\mathsf{idle} : \mathbf{1} \longrightarrow \mathbf{1}$

$$\mathsf{idle} \;=\; \ulcorner \mathsf{id}_{\mathbf{1}} \urcorner$$

which propagates an unstructured stimulus (*e.g.*, the push of a button) leading to a unstructured reaction (*e.g.*, exciting a led)

Tensors

Defined as lax functors: $\square : \mathsf{Cp} \times \mathsf{Cp} \longrightarrow \mathsf{Cp}$



where $\square$ stands for

choice	$p \boxplus q$
parallel	$p \boxtimes q$
concurrent	$p \boxast q$ with $I \boxast J = I + J + I \times J$

Choice

- An action on objects: $I \boxplus J \;=\; I + J$
- A family of functors

$$\boxplus_{I,O,J,R} : \mathsf{Cp}(I,O) \times \mathsf{Cp}(J,R) \longrightarrow \mathsf{Cp}(I+J, O+R)$$

yielding

$$p \boxplus q \;=\; \langle \langle u_p, u_q \rangle \in U_p \times U_q, \overline{a}_{p\boxplus q} \rangle$$

Choice

where

$$a_{p\boxplus q} = \quad U_p \times U_q \times (I+J) \xrightarrow{\mathsf{dr}} U_p \times U_q \times I + U_p \times U_q \times J$$
$$\xrightarrow{\mathsf{xr}+\mathsf{a}} U_p \times I \times U_q + U_p \times (U_q \times J)$$
$$\xrightarrow{a_p \times \mathsf{id} + \mathsf{id} \times a_q} \mathsf{B}\,(U_p \times O) \times U_q + U_p \times \mathsf{B}\,(U_q \times R)$$
$$\xrightarrow{\tau_r + \tau_l} \mathsf{B}\,(U_p \times O \times U_q) + \mathsf{B}\,(U_p \times (U_q \times R))$$
$$\xrightarrow{\mathsf{B}\mathsf{xr} + \mathsf{B}\mathsf{a}^\circ} \mathsf{B}\,(U_p \times U_q \times O) + \mathsf{B}\,(U_p \times U_q \times R)$$
$$\xrightarrow{[\mathsf{B}\,(\mathsf{id} \times \iota_1), \mathsf{B}\,(\mathsf{id} \times \iota_2)]} \mathsf{B}\,(U_p \times U_q \times (O+R))$$

and maps pairs of arrows $\langle h_1, h_2 \rangle$ into $h_1 \times h_2$

Choice

$\boxed{\text{Laws}}$

- $\boxplus$ is a lax functor in Cp and lifts function sums:

$$(p \boxplus p') \,;\, (q \boxplus q') \;\sim\; (p \,;\, q) \boxplus (p' \,;\, q')$$
$$\mathsf{copy}_{K \boxplus K'} \;\sim\; \mathsf{copy}_K \boxplus \mathsf{copy}_{K'}$$
$$\ulcorner f \urcorner \boxplus \ulcorner g \urcorner \;\sim\; \ulcorner f + g \urcorner$$

- $\boxplus$ is a symmetric tensor in each hom-cat up to isomorphic wiring:

$$(p \boxplus q) \boxplus r \sim (p \boxplus (q \boxplus r))[\mathsf{a}_+, \mathsf{a}_+{}^\circ]$$
$$\mathsf{nil} \boxplus p \;\sim\; p[\mathsf{r}_+, \mathsf{r}_+{}^\circ] \quad \text{and} \quad p \boxplus \mathsf{nil} \;\sim\; p[\mathsf{l}_+, \mathsf{l}_+{}^\circ]$$
$$p \boxplus q \sim (q \boxplus p)[\mathsf{s}_+, \mathsf{s}_+]$$

Proof Outline (1)

- $\boxplus_{I,O,J,R}$ is a functor from $\mathsf{Cp}(I,O) \times \mathsf{Cp}(J,R)$ to $\mathsf{Cp}(I+J, O+R)$
- Prove, for each $\langle I, I', K, K', O, O' \rangle$, a nt

$$\mathsf{m}_\boxplus : ;_{I+I',K+K',O+O'} \circ (\boxplus_{I,K,I',K'} \times \boxplus_{K,O,K',O'}) \Longrightarrow \boxplus_{I,O,I',O'} \circ (;_{I,K,O} \times ;_{I',K'}$$

whose components $\mathsf{m}_{p,q,p',q'}$, are Cp 2-cells, *i.e.*, a seed preserving comorphisms from $(p \boxplus q) \,;\, (p' \boxplus q')$ to $(p \,;\, p') \boxplus (q \,;\, q')$.

- Similarly, for each pair $\langle K, K' \rangle$ of objects,

$$\mathsf{u}_\boxplus : \mathsf{copy}_{K \boxplus K'} \Longrightarrow \boxplus_{K,K',K,K'} \circ (\mathsf{copy}_K \times \mathsf{copy}_{K'})$$

i.e., a Cp 2-cell from $\mathsf{copy}_{K \boxplus K'}$ to $\mathsf{copy}_K \boxplus \mathsf{copy}_{K'}$

- In both cases coherence condition are easily verified

Proof Outline (2)

- To prove a bisimulation equation as $p \sim q$ a function $h : U_p \longrightarrow U_q$ is identified and shown to form a seed preserving comorphism. In this case establish Set natural isomorphisms a, r, l and s as Cp 2-cells.
- Verification of comorphism conditions as usual.
  Seed preservation is obvious in all cases.

Looking for Universals

An either construction

$$[p, q] \;=\; (p \boxplus q)\,;\ulcorner \triangledown \urcorner \quad \text{where } \triangledown = [id, id]$$

st that the diagram commutes up to bisimulation

$$\begin{tikzcd} I \arrow[r, "\ulcorner \iota_1 \urcorner"] \arrow[dr, "p"'] & I \boxplus J \arrow[d, "{[p,q]}"] & J \arrow[l, "\ulcorner \iota_2 \urcorner"'] \arrow[dl, "q"] \\ & O & \end{tikzcd}$$

$$\begin{aligned} \ulcorner \iota_1 \urcorner ; [p, q] &\sim p \\ \ulcorner \iota_2 \urcorner ; [p, q] &\sim q \end{aligned}$$

- But $[p, q]$ fails to be unique
- ... therefore, $\boxplus$ lifts to a weak coproduct in Bh

Looking for Universals

Uniqueness fails even for $\mathsf{B} = \mathsf{Id}$ because $\triangledown$ does not keep its naturality when lifted to Cp

$\boxed{\text{Refutation}}$

- Let $p = \langle 0 \in \mathbb{N}, \overline{a}_p \rangle : \mathbb{N} \longrightarrow \mathbb{N}$ st, upon receiving an input i, i is added to the current state value and the result sent to the output.
- Let $s = \langle \iota_1 5, \iota_2 3, \iota_1 4, ... \rangle$ be a sequence of inputs (of type $\mathbb{N} + \mathbb{N}$)
- The reaction to s of $\ulcorner \triangledown \urcorner ; (p \boxplus p)$ is $\langle 5, 3, 9, \ldots \rangle$
- while $p \,; \ulcorner \triangledown \urcorner$, resorting only to one copy of p, produces $\langle 5, 8, 12, \ldots \rangle$

Looking for Universals

- Failing universality means there is not a *fusion* law for $\boxplus$, even in the deterministic case
- ... but still we have cancellation,

$$
\begin{array}{rl}
\text{reflection} & [\ulcorner \iota_1 \urcorner, \ulcorner \iota_2 \urcorner] \;\sim\; \mathsf{copy}_{I+J} \\
\text{absorption} & (p \boxplus q) \,; [p', q'] \;\sim\; [p \,; p', q \,; q'] \\
 & \\
\text{definition} & p \boxplus q \;\sim\; [p \,; \ulcorner \iota_1 \urcorner, p \,; \ulcorner \iota_2 \urcorner] \\
\text{naturality} & \ulcorner \iota_1 \urcorner ; (p \boxplus q) \;\sim\; p \,; \ulcorner \iota_1 \urcorner \\
 & \ulcorner \iota_2 \urcorner ; (p \boxplus q) \;\sim\; q \,; \ulcorner \iota_2 \urcorner \\
\text{`idempotency'} & p \,; \ulcorner \iota_1 \urcorner \;\sim\; \ulcorner \iota_1 \urcorner ; (p \boxplus p)
\end{array}
$$

Looking for Universals

Proof (Absorption):

$$
\begin{array}{ll}
 & (p \boxplus q) \,;\, [p', q'] \\
\sim & \quad \{ \text{ definition of either in } \mathsf{Cp} \} \\
 & (p \boxplus q) \,;\, ((p' \boxplus q') \,;\, \ulcorner \nabla \urcorner) \\
\sim & \quad \{ \text{ ; associative } \} \\
 & ((p \boxplus q) \,;\, (p' \boxplus q')) \,;\, \ulcorner \nabla \urcorner \\
\sim & \quad \{ \boxplus \text{ functor } \} \\
 & ((p \,;\, p') \boxplus (q \,;\, q')) \,;\, \ulcorner \nabla \urcorner \\
\sim & \quad \{ \text{ definition of either in } \mathsf{Cp} \} \\
 & [p \,;\, p', q \,;\, q']
\end{array}
$$

Parallel

- An action on objects: $I \boxtimes J \;=\; I \times J$
- A family of functors

$$\boxtimes_{IOJR} : \mathsf{Cp}(I, O) \times \mathsf{Cp}(J, R) \longrightarrow \mathsf{Cp}(I \times J, O \times R)$$

yielding

$$p \boxtimes q \;=\; \langle \langle u_p, u_q \rangle \in U_p \times U_q, \overline{a}_{p \boxtimes q} \rangle$$

Parallel

where

$$a_{p \boxtimes q} = \quad U_p \times U_q \times (I \times J) \xrightarrow{\mathsf{m}} U_p \times I \times (U_q \times J)$$
$$\xrightarrow{a_p \times a_q} \mathsf{B}\,(U_p \times O) \times \mathsf{B}\,(U_q \times R)$$
$$\xrightarrow{\delta_l} \mathsf{B}\,(U_p \times O \times (U_q \times R))$$
$$\xrightarrow{\mathsf{B}\ \mathsf{m}} \mathsf{B}\,(U_p \times U_q \times (O \times R))$$

and maps pairs of arrows $\langle h_1, h_2 \rangle$ into $h_1 \times h_2$

Parallel

Laws

$$\begin{array}{rl}
\text{lax} & (p \boxtimes p') \,;\, (q \boxtimes q') \ \sim\ (p \,;\, q) \boxtimes (p' \,;\, q') \\
 & \mathsf{copy}_{K \boxtimes K'} \ \sim\ \mathsf{copy}_K \boxtimes \mathsf{copy}_{K'} \\
 & \\
\text{functions} & \ulcorner f \urcorner \boxtimes \ulcorner g \urcorner \ \sim\ \ulcorner f \times g \urcorner \\
\text{assoc} & (p \boxtimes q) \boxtimes r \ \sim\ (p \boxtimes (q \boxtimes r))[\mathsf{a}, \mathsf{a}^\circ] \\
\text{id} & \mathsf{idle} \boxtimes p \ \sim\ p[\mathsf{r}, \mathsf{r}^\circ] \\
\text{zero} & \mathsf{nil} \boxtimes p \ \sim\ \mathsf{nil}[\mathsf{zl}, \mathsf{zl}^\circ] \\
 & \\
\text{comm} & p \boxtimes q \ \sim\ (q \boxtimes p)[\mathsf{s}, \mathsf{s}] \quad \text{if } \mathsf{B} \text{ is commutative}
\end{array}$$

Parallel

A split construction

$$\langle p, q \rangle \;=\; \ulcorner \triangle \urcorner \,;(p \boxtimes q) \quad \text{where } \triangle = \langle id, id \rangle$$

but, in general, it does not guarantee the commutativity of

$$\begin{tikzcd}
 & I \arrow[dl, "p"'] \arrow[d, "\langle p,q \rangle"] \arrow[dr, "q"] & \\
O & O \boxtimes R \arrow[l, "\ulcorner \pi_1 \urcorner"] \arrow[r, "\ulcorner \pi_2 \urcorner"'] & R
\end{tikzcd}$$

- It does, and cancellation $\langle p, q \rangle \,; \ulcorner \pi_1 \urcorner \sim p$ holds for commutative B excluding failure.
  Let's check $\pi_1 : U_p \times U_q \longrightarrow U_p$ to establish bisimilarity:

Proof Outline

$$\begin{aligned}
& \mathsf{B}(\pi_1 \times \mathsf{id}) \cdot a_{\langle p,q \rangle ; \ulcorner \pi_1 \urcorner} \\
= & \quad \{ \text{ definitions } \} \\
& \mathsf{B}(\pi_1 \times \pi_1) \cdot \mathsf{Bm} \cdot \delta_l \cdot (a_p \times a_q) \cdot \mathsf{m} \cdot (\mathsf{id} \times \triangle) \\
= & \quad \{ \text{ routine: } \pi_1 \times \pi_1 = \pi_1 \cdot \mathsf{m},\ \mathsf{m}^{\circ} = \mathsf{m} \} \\
& \mathsf{B}\pi_1 \cdot \delta_l \cdot (a_p \times a_q) \cdot \mathsf{m} \cdot (\mathsf{id} \times \triangle) \\
= & \quad \{ \star \} \\
& \pi_1 \cdot (a_p \times a_q) \cdot \mathsf{m} \cdot (\mathsf{id} \times \triangle) \\
= & \quad \{ \times \text{ cancellation } \} \\
& a_p \cdot \pi_1 \cdot \mathsf{m} \cdot (\mathsf{id} \times \triangle) \\
= & \quad \{ \text{ routine: } \pi_1 \times \pi_1 = \pi_1 \cdot \mathsf{m} \text{ and } \pi_1 \cdot \triangle = \mathsf{id} \}
\end{aligned}$$

Parallel

- However

reflection	$\langle \ulcorner \pi_1 \urcorner, \ulcorner \pi_2 \urcorner \rangle \sim \mathsf{copy}_{O \times R}$
absorption	$\langle p, q \rangle \,;\, (p' \boxtimes q') \sim \langle p \,;\, p', q \,;\, q' \rangle$ for B commutative
definition	$p \boxtimes q \sim \langle \ulcorner \pi_1 \urcorner \,;\, p, \ulcorner \pi_2 \urcorner \,;\, q \rangle$

- Whenever the lifting $\ulcorner \triangle \urcorner$ keeps naturality

$$\ulcorner \triangle \urcorner \,;\, (p \boxtimes p) \sim p \,;\, \ulcorner \triangle \urcorner$$

fusion $\quad r \,;\, \langle p, q \rangle \sim \langle r \,;\, p, r \,;\, q \rangle$

holds — $\boxed{\text{this is the case of B a non-collection monad}}$

Proof Outline

$$a_{(p \boxtimes p)} \cdot (\triangle \times \triangle)$$

$= \quad \{\ \boxtimes$ definition, $\mathsf{m} \cdot (\triangle \times \triangle) = \triangle\ \}$

$$\mathsf{B}\mathsf{m} \cdot \delta_l \cdot (a_p \times a_p) \cdot \triangle$$

$= \quad \{\ \triangle$ natural $\}$

$$\mathsf{B}\mathsf{m} \cdot \delta_l \cdot \triangle \cdot a_p$$

$= \quad \{\ \star\ \}$

$$\mathsf{B}\mathsf{m} \cdot \mathsf{B} \triangle \cdot a_p$$

$= \quad \{\ \mathsf{m} \cdot \triangle = \triangle \times \triangle$ and definition $\}$

$$\mathsf{B}(\triangle \times \mathsf{id}) \cdot a_{p;\ulcorner \triangle \urcorner}$$

Proof Outline

However,

- $\delta_l \cdot \triangle =$ B $\triangle$ does not hold for any monad involving the notion of a collection
- Counterexample:

$$(\delta_l \cdot \triangle)\{5,2\} = \{\{5,5\},\{5,2\},\{2,5\},\{2,2\}\}$$
$$(\mathcal{P}\triangle)\{5,2\} = \{\{5,5\},\{2,2\}\}$$

Proof Outline (Fusion)

$$\begin{array}{ll}
 & r \,;\, \langle p, q\rangle \\
\sim & \quad \{\ \text{definition}\ \} \\
 & r \,;\, (\ulcorner\triangle\urcorner \,;\, (p \boxtimes q)) \\
\sim & \quad \{\ \text{assumption and assoc}\ \} \\
 & \ulcorner\triangle\urcorner \,;\, ((r \boxtimes r) \,;\, (p \boxtimes q)) \\
\sim & \quad \{\ ;\ \text{lax functor}\} \\
 & \ulcorner\triangle\urcorner \,;\, ((r \,;\, p) \boxtimes (r \,;\, q)) \\
\sim & \quad \{\ \text{definition}\ \} \\
 & \langle r \,;\, p, r \,;\, q\rangle
\end{array}$$

Parallel

- Summing up, $\boxtimes$ lifts to a product in the category of behaviours of deterministic components
- Product projections keep naturality only when cancellation holds
- ... but always

$$(\ulcorner f \urcorner \boxtimes q) \,;\, \ulcorner \pi_2 \urcorner \;\sim\; \ulcorner \pi_2 \urcorner \,;\, q$$
$$(p \boxtimes \ulcorner f \urcorner) \,;\, \ulcorner \pi_1 \urcorner \;\sim\; \ulcorner \pi_1 \urcorner \,;\, p$$

Concurrent

- An action on objects: $I \circledast J \;=\; I + J + I \times J$
- A family of functors

$$\circledast_{IOJR} : \mathsf{Cp}(I,O) \times \mathsf{Cp}(J,R) \longrightarrow \mathsf{Cp}(I + J + I \times J, O + R + O \times R)$$

yielding

$$p \circledast q \;=\; \langle \langle u_0, v_0 \rangle \in U_p \times U_q, \overline{a}_{p \circledast q} \rangle$$

Concurrent

where

$$a_{p \boxast q} = \quad U_p \times U_q \times (I \boxast J) \xrightarrow{[\textcolor{red}{\mathsf{B}}(\mathsf{id} \times \iota_1), \textcolor{red}{\mathsf{B}}(\mathsf{id} \times \iota_2)] \cdot (a_{p \boxplus q} + a_{p \boxtimes q}) \cdot \mathsf{dr}} \textcolor{red}{\mathsf{B}}\ (U_p \times U_q \times (O \boxast R))$$

and maps pairs of arrows $\langle h_1, h_2 \rangle$ into $h_1 \times h_2$

Concurrent

Laws

$$
\begin{array}{rl}
\text{lax} & (p \boxast p')\,;(q \boxast q') \ \sim\ (p\,;q) \boxast (p'\,;q') \\
 & \mathsf{copy}_{K \boxast K'} \ \sim\ \mathsf{copy}_K \boxast \mathsf{copy}_{K'} \\
\text{assoc} & (p \boxast q) \boxast r \ \sim\ (p \boxast (q \boxast r))[\mathsf{a}_*, \mathsf{a}_*{}^\circ] \\
\text{id} & \mathsf{nil} \boxast p \ \sim\ p[\mathsf{r}_*, \mathsf{r}_*{}^\circ] \\
 & p \boxast \mathsf{nil} \ \sim\ p[\mathsf{l}_*, \mathsf{l}_*{}^\circ] \\
\text{comm} & p \boxast q \ \sim\ (q \boxast p)[\mathsf{s}_*, \mathsf{s}_*] \quad \text{if } \textcolor{red}{\mathsf{B}} \text{ is commutative} \\
\text{reduction} & \ulcorner \iota_1 \urcorner\,;(p \boxast q) \ \sim\ (p \boxplus q)\,;\ulcorner \iota_1 \urcorner \\
 & \ulcorner \iota_2 \urcorner\,;(p \boxast q) \ \sim\ (p \boxtimes q)\,;\ulcorner \iota_2 \urcorner
\end{array}
$$

Proof Outline

$$
\begin{array}{ll}
 & a_{(p \boxast q)[\iota_1, \mathsf{id}]} \\
= & \quad \{ \ \boxast \text{ and wrapping definitions } \} \\
 & [\mathsf{B}(\mathsf{id} \times \iota_1), \mathsf{B}(\mathsf{id} \times \iota_2)] \cdot (a_{(p \boxplus q)} + a_{(p \boxtimes q)}) \cdot \mathsf{dr} \cdot (\mathsf{id} \times \iota_1) \\
= & \quad \{ \ \text{routine } \} \\
 & [\mathsf{B}(\mathsf{id} \times \iota_1), \mathsf{B}(\mathsf{id} \times \iota_2)] \cdot (a_{(p \boxplus q)} + a_{(p \boxtimes q)}) \cdot \iota_1 \\
= & \quad \{ \ + \text{ absorption } \} \\
 & [\mathsf{B}(\mathsf{id} \times \iota_1) \cdot a_{(p \boxplus q)}, \mathsf{B}(\mathsf{id} \times \iota_1) \cdot a_{(p \boxtimes q)}] \cdot \iota_1 \\
= & \quad \{ \ + \text{ cancellation, } \boxast \text{ and wrapping definition } \} \\
 & \mathsf{B}(\mathsf{id} \times \iota_1) \cdot a_{(p \boxplus q)} \ = \ a_{(p \boxplus q)}[\mathsf{id}, \iota_1]
\end{array}
$$

Interaction

Partial combinators, which generalise pipeline, defined as functors between some families of Cp hom-categories, acting as the identity on arrows

Feedback

$$p \Lsh : Z \longrightarrow Z \ = \langle u_p \in U_p, \overline{a}_{p\Lsh} \rangle$$

where

$$a_{p\Lsh} \ = \qquad U_p \times Z \xrightarrow{a_p} \mathsf{B}(U_p \times Z) \xrightarrow{\mathsf{B}a_p} \mathsf{BB}(U_p \times Z) \xrightarrow{\mu} \mathsf{B}(U_p \times Z)$$

i.e., $a_{p\Lsh} = a_p \bullet a_p,$

Interaction

$\boxed{\text{Hook}}$

$$p \Lsh_Z : I + Z \longrightarrow O + Z \;\; = \langle u_p \in U_p, \overline{a}_{p\Lsh_Z} \rangle$$

where

$$\begin{aligned}
\overline{a}_{p\Lsh_Z} = \quad & U_p \times (I + Z) \xrightarrow{\;a_p\;} \mathsf{B}(U_p \times (O + Z)) \\
& \xrightarrow{\;\mathsf{B}\mathsf{dr}\;} \mathsf{B}(U_p \times O + U_p \times Z) \\
& \xrightarrow{\;\mathsf{B}(\mathsf{id}\times\iota_1 + \mathsf{id}\times\iota_2)\;} \mathsf{B}(U_p \times (O + Z) + U_p \times (I + Z)) \\
& \xrightarrow{\;\mathsf{B}(\eta + a_p)\;} \mathsf{B}(\mathsf{B}(U_p \times (O + Z)) + \mathsf{B}(U_p \times (O + Z))) \\
& \xrightarrow{\;\mathsf{B}\nabla\;} \mathsf{BB}(U_p \times (O + Z)) \\
& \xrightarrow{\;\mu\;} \mathsf{B}(U_p \times (O + Z))
\end{aligned}$$

Interaction

$\boxed{\text{Laws}}$

functions	$\ulcorner f \urcorner \Lsh \;\sim \ulcorner f \cdot f \urcorner$
	$\ulcorner g \urcorner \Lsh_Z \;\sim \ulcorner [\iota_1, g \cdot \iota_2] \cdot g \urcorner$ for $g : I + Z \longrightarrow O + Z$
wrap	$p \Lsh \;\sim p[\mathsf{r}_+, \mathsf{r}_+{}^\circ] \Lsh_Z \; [\mathsf{r}_+{}^\circ, \mathsf{r}_+]$
choice	$q \boxplus p \Lsh \;\sim (q \boxplus p) \Lsh_Z$
	$p \Lsh \boxplus q \;\sim\; (q \boxplus p) \Lsh_Z \; [\mathsf{s}_+, \mathsf{s}_+]$

for components $p : Z \longrightarrow Z$ and $q : I \longrightarrow O$

Interaction

- All but the last law above are strict Cp arrow equalities, and not just bisimulations.
- Last law generalises to

$$(q \boxplus p)[\mathsf{a}_+, \mathsf{a}_+{}^\circ] \Lsh_Z \ \sim (q \boxplus p \Lsh_Z)[\mathsf{a}_+, \mathsf{a}_+{}^\circ]$$

  for $p : J + Z \longrightarrow R + Z$
- For $p : Z \longrightarrow Z$ and $q : R \longrightarrow R$, and $i : W \longrightarrow Z$ iso,

$$\begin{aligned} p \Lsh [i, i^\circ] &\sim p[i, i^\circ] \Lsh \\ (p \boxplus q) \Lsh &\sim p \Lsh \boxplus q \Lsh \\ (p \boxtimes q) \Lsh &\sim p \Lsh \boxtimes q \Lsh \\ (p \boxast q) \Lsh &\sim p \Lsh \boxast q \Lsh \end{aligned}$$

Interaction

- For $p : I + K \longrightarrow O + K$ and $q : J \longrightarrow R$, functions $f : I' \longrightarrow I$, $g : O \longrightarrow O'$ and iso $i : W \longrightarrow K$. Then

$$\begin{aligned} p \Lsh_K [f + i, g + i^\circ] &\sim p[f + i, g + i^\circ] \Lsh_W \\ (p \boxplus q)[\mathsf{xr}_+, \mathsf{xr}_+] \Lsh_K \ &\sim (p \Lsh_K \boxplus q)[\mathsf{xr}_+, \mathsf{xr}_+] \end{aligned}$$

Ex: The Game of Life

Simple model of cellular behaviour based on a grid of cells each of which sends and receives Boolean stimulus to and from its four adjacent neighbours

Rules

- Each living cell with less than two or more than three living neighbours dies in the next generation.
- Each dead cell with exactly three living neighbours becomes alive.
- Each living cell with less than two or three living neighbours survives until the next generation.

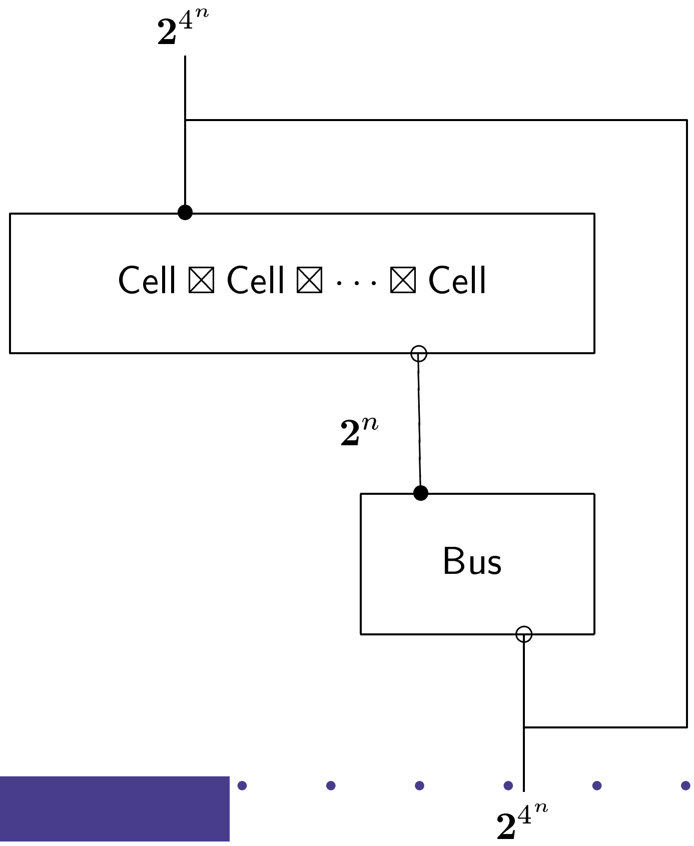
Ex: The Game of Life

$$\mathsf{Cell} : \mathbf{2} \times \mathbf{2} \times \mathbf{2} \times \mathbf{2} \longrightarrow \mathbf{2} \quad = \quad \langle \mathsf{tt} \in \mathbf{2}, \overline{a}_{\mathsf{Cell}} \rangle$$

$$a_{\mathsf{Cell}}\ \langle u, t \rangle \ = \ \text{let } n = \mathsf{living}\ t$$
$$\text{in} \begin{cases} \langle \mathsf{ff}, \mathsf{ff} \rangle & \text{if } u = \mathsf{tt} \wedge (n < 1 \vee n > 3) \\ \langle \mathsf{tt}, \mathsf{tt} \rangle & \text{if } u = \mathsf{ff} \wedge n = 3) \\ \langle u, u \rangle & \text{otherwise} \end{cases}$$

- The n cells are organised as a fully connected matrix of k rows and l columns ($n = k \times l$), so that the neighbours of cell $\langle i, j \rangle$ are $\langle i-1, j \rangle$, $\langle i+1, j \rangle$, $\langle i, j-1 \rangle$ and $\langle i, j+1 \rangle$ (in the 'west' to 'south' directions, respectively) computed in the k and l rings (*i.e.*, $1 - 1 = k$, $k + 1 = 1$ and $1 - 1 = l$, $l + 1 = 1$).

Ex: The Game of Life



Ex: The Game of Life

$$\mathsf{w} \;=\; \langle \mathsf{in}_{\langle i,j \rangle} \mid i,j \in 1..n \rangle$$

where input to cell $\langle i,j \rangle$ is the *split* of the outputs of its neighbours, *i.e.*,

$$\mathsf{in}_{\langle i,j \rangle} : \mathbf{2}^n \longrightarrow \mathbf{2}^4$$
$$\mathsf{in}_{\langle i,j \rangle} = \langle \mathsf{out}_{\langle i, \mathsf{dec}_n j \rangle}, \mathsf{out}_{\langle \mathsf{dec}_n i, j \rangle}, \mathsf{out}_{\langle i, \mathsf{inc}_n j \rangle}, \mathsf{out}_{\langle \mathsf{inc}_n i, j \rangle} \rangle$$

for

$$\mathsf{out}_{\langle i,j \rangle} : \mathbf{2}^n \longrightarrow \mathbf{2}$$
$$\mathsf{out}_{\langle i,j \rangle} = \pi_{j+(n \times (i-1))}$$

the $j + (n \times (i-1))$-projection.

Ex: Asynchronous Merging

$$\mathsf{Merge} : I+J \longrightarrow \mathbf{1}+(I\times J) \quad = \quad \langle\langle \iota_2\, *, \iota_2\, *\rangle \in (I+\mathbf{1})\times(J+\mathbf{1}), \overline{a}_{\mathsf{Merge}}\rangle$$

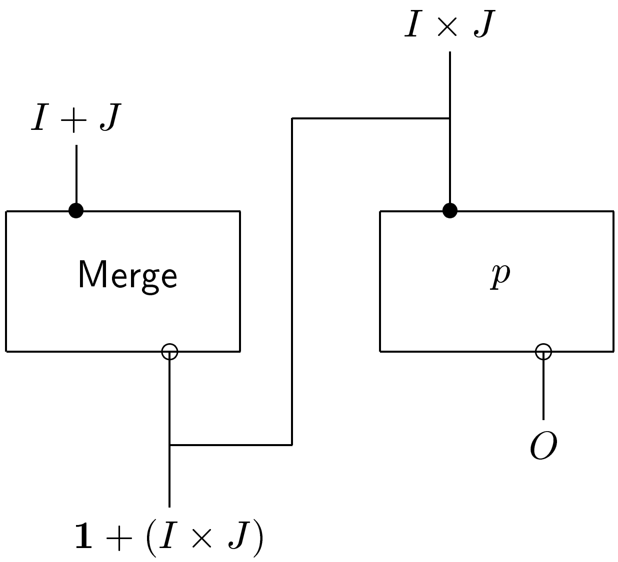
over state space $C_{I,J} = I \times J + I + J + \mathbf{1} \cong (I+\mathbf{1}) \times (J+\mathbf{1})$, where

$$a_{\mathsf{Merge}}\ \langle u, x\rangle \ = \eta_{\mathsf{B}} \begin{cases} \langle\langle x, (\pi_2\ u)\rangle, ((\text{is}\text{—}J\ \pi_2 u'\ ) \ \rightarrow\ (\iota_2\ u'),\ (\iota_1\ *))\rangle & \text{if } (\text{is}\text{—}I\ x) \\ \langle\langle(\pi_1\ u), x\rangle, ((\text{is}\text{—}I\ \pi_2 u'\ ) \ \rightarrow\ (\iota_2\ u'),\ (\iota_1\ *))\rangle & \text{if } (\text{is}\text{—}J\ x) \end{cases}$$

Idea: Composing Merge with a component p makes the relevant stimuli to be passed to p whenever possible

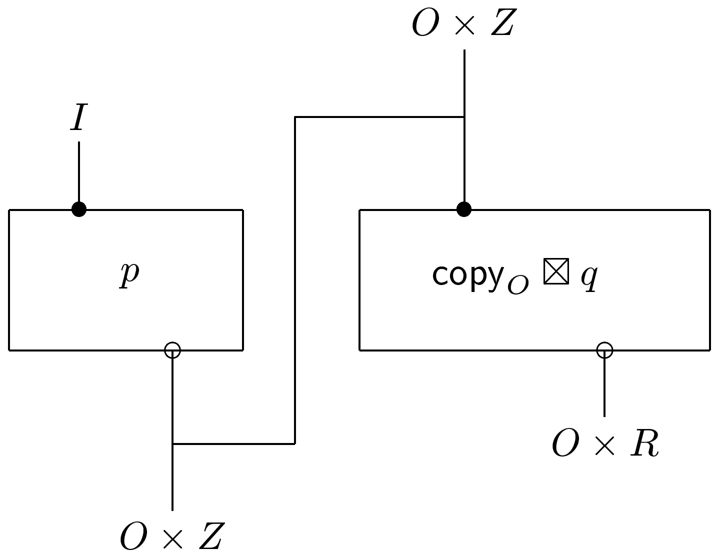
Ex: Asynchronous Merging

$$((\mathsf{Merge} \boxplus p)\ [\mathsf{id}, \mathsf{xr}_+])\ \Lsh_{I\times J}$$



Ex: Delayed Projections

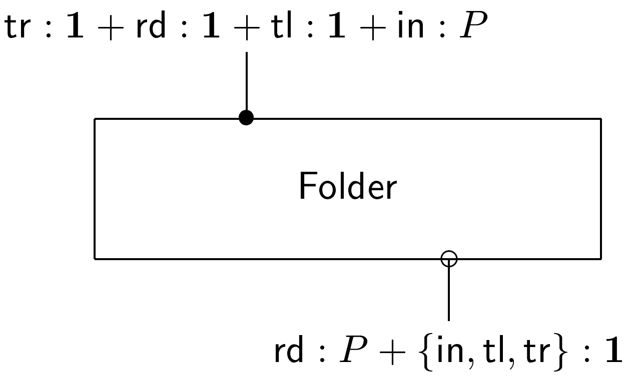
- Concerns the lost of part of a multiplicative output
- Achieved by packing q with an identity component for O, *i.e.*$p\,;(\mathsf{copy}_O \boxtimes q)$:



Ex: A Folder

$$\begin{cases} \mathsf{in}: & P \longrightarrow \mathbf{1} \\ \mathsf{rd}: & \mathbf{1} \longrightarrow P \\ \mathsf{tl}, \mathsf{tr}: & \mathbf{1} \longrightarrow \mathbf{1} \end{cases}$$

depicted as



Ex: A Folder

- Right stack: replicate input to push by wrapping with ∇_P

$$\mathsf{RightS} \;=\; \mathsf{Stack}[\mathsf{id}+\nabla,\mathsf{id}] \;:\; \mathbf{1}+\mathbf{1}+(P+P) \longrightarrow P+P+\mathbf{1}$$

- Left stack: remove access to the top button and collapse unit outputs

$$\mathsf{LeftS} \;=\; \mathsf{Stack}[\iota_2+\mathsf{id},(\mathsf{id}+!_{P+\mathbf{1}})\cdot \mathsf{a}_+] \;:\; \mathbf{1}+P \longrightarrow P+\mathbf{1}$$

- Compose both stacks in an additive context

$$\mathsf{LeftS} \boxplus \mathsf{RightS} :\; \mathbf{1}+P+(\mathbf{1}+\mathbf{1}+(P+P)) \longrightarrow P+\mathbf{1}+(P+P+\mathbf{1})$$

Ex: A Folder

- Connect by hook, which requires wrapping by a pair of suitable isomorphisms

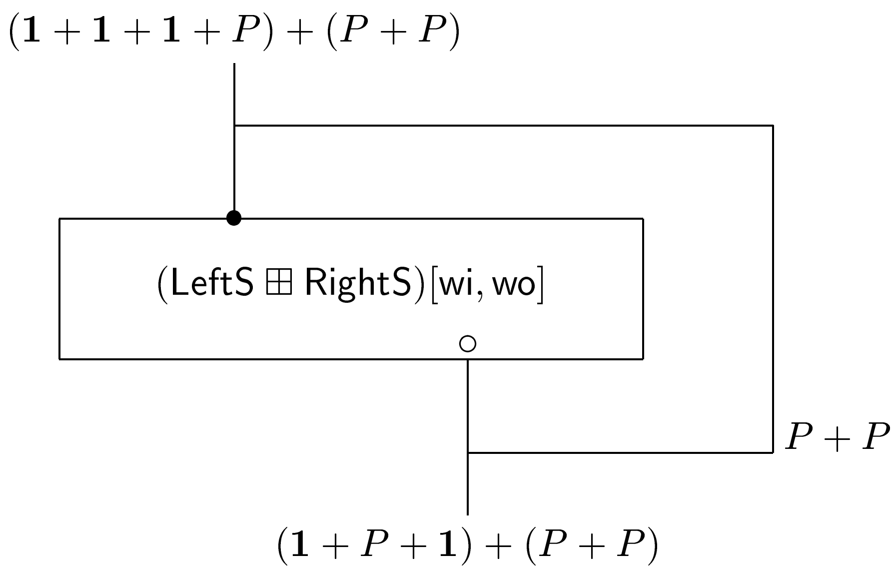
$$\mathsf{AlmostFolder} \;=\; ((\mathsf{LeftS} \boxplus \mathsf{RightS})[\mathsf{wi},\mathsf{wo}])\,{}^{\Lsh}{}_{P+P}$$

where, denoting by ι_{ij} the composite $\iota_i \cdot \iota_j$,

$$\mathsf{wi} \;=\; \Big[[[[\iota_{11},\iota_{112}],\iota_{212}],\iota_{222}],[\iota_{21},\iota_{122}]\Big]$$

$$\mathsf{wo} \;=\; \Big[[\iota_{12},\iota_{111}],[[\iota_{211},\iota_{22}],\iota_{21}]\Big]$$

A Folder



Ex: A Folder

- Conform to Folder interface: restrict the feed back input — by pre-composing with $\mathsf{fi} = \iota_1$ — and collapse both the trivial output and the feed back one to $\mathbf{1}$, by post-composing with $\mathsf{fo} = [[[\iota_2, \iota_1], \iota_2], \iota_2 \cdot !_{P+P}]$

$$\mathsf{Folder} \;=\; \mathsf{AlmostFolder}[\mathsf{fi}, \mathsf{fo}]$$

Animations

Examples for Deterministic Components

```
data U -> cp(I,O) =  a : U -> I => U * O.

def pipe: cp(I, K) * cp(K, O) -> cp(I, O)
     = pp => (| (p, q) => a: i => a(i,p)
                           ; (p',k) => a(k,q)
                           ; (q',o) => ((p', q'), o)
              |) pp.

def copy: 1 -> cp(K,K)
     = () => (| () => a: k => ((), k) |) ().
```


Animations

```
def lift{f:I -> O}: 1 -> cp(I,O)
     = () => (| _ => a: i => ((), f i ) |) ().


def choice: cp(I, O) * cp(J, R) -> cp(I+J, O+R)
  = pp =>
   (| (p, q)
    => a: b0 i => a(i,p) ; (p',o) => ((p', q), b0 o)
        | b1 j => a(j,q) ; (q',r) => ((p, q'), b1 r)
    |) pp.
```

Testing

- Direct:

```
>> a(4, pipe (cell 5, cell 2)).
((a: <function>), 11) : cp(int, int) * int
```

- With a testing structure:

$$
\begin{tikzcd}
\mathtt{stream\ O} \arrow[rr, "\omega_{\mathtt{O}\times\mathsf{Id}}"] & & \mathtt{O} \times \mathtt{stream\ O} \\
\mathtt{stream\ I} \times \mathtt{cp(I,O)} \arrow[u, "\mathtt{obs} = [\![(\mathtt{obs\_gen})]\!]"] \arrow[rr, "\mathtt{obs\_gen}"] & & \mathtt{O} \times (\mathtt{stream\ I} \times \mathtt{cp(I,O)}) \arrow[u, "\mathsf{id} \times \mathtt{obs}"']
\end{tikzcd}
$$

```
def obs: stream I * cp(I,O) -> stream O
 = (s,p) => (| (s,p) => head: p1 a(head s, p)
             |      tail: (tail s, p0 a(head s,p))
             |) (s, p).
```


Testing

Printout

```
>> obs(ints, cell 4).
(head: ..., tail: ...)

(head: 4, tail: (head: ..., tail: ...))

(head: 4, tail: (head: 9, tail: (head: ..., tail: ..

(head: 4, tail: (head: 9, tail: (head: 18, tail: ...

(head: 4, tail: (head: 9, tail: (head: 18, tail: (he
...
```

Testing

Testing Feedback

```
def feed: cp(K, K) -> cp(K,K)
    = pp => (| p => a
             : k => a(k,p) ; (p',k') => a(k',p')
             |) pp.

def fd_obs: K * cp(K,K)  -> stream(K)
    = (k, p)  =>
      (| (k, p) => head: k
       |            tail: (p1 a(k, p), p0 a(k, p))
       |) (k, p).
```


Going Generic

```
data U -> cp(I,O) =  a : U -> I => B (U * O).

def pipe: cp(I, K) * cp(K, O) -> cp(I, O)
 = pp =>
  (| (p, q) =>
      a: i => mu B{B{iassoc}} B{taul} B{fp{fid,a}}
                B{fp{fid,swap}} B{assoc} B{xr}
                taur (a(i,p), q) |) pp.

def par: cp(I, O) * cp(J, R) -> cp(I*J, O*R)
 = pp => (| (p, q) =>
              a: (i,j) => B{m} deltal (a(i,p), a(j,q)
           |) pp.
```

Going Generic

Ex: Non Deterministic Components

$$\text{circle} : \mathbb{Z} \longrightarrow \mathbb{Z} \quad = \quad \langle 0 \in \mathbb{Z}, \overline{a}_{\text{Circle}} \rangle$$

$$a_{\text{circle}}\ \langle u, n \rangle \ = ((u = 0)\ \rightarrow\ \emptyset,\ \{\langle u-1, n-u \rangle, \langle u-1, n+u \rangle\})$$

```
def circle: int -> cp(int,int)
 = u =>
   (| u =>
      a: n =>
       {0 => emptyset
       |_ => [(dec u, sub(n,u)), (dec u, add(n,u))]}
    |) u.
```


Going Generic

Compare pipe — in which the distances to the intermediate inputs are computed in a new component (and a ‘fresh’ state value) — and feedback, in which case such computation is relative to the updated state value:

```
>> a(5, pipe (circle 2, circle 2)).
[((a: <function>), 1), ((a: <function>), 5),
 ((a: <function>), 5), ((a: <function>), 9)]
 : set(cp(int, int) * int)

>> a(5, feed circle 2).
[((a: <function>), 2), ((a: <function>), 4),
  ((a: <function>), 6), ((a: <function>), 8)]
 : set(cp(int, int) * int)
```

Going Generic

Testing structures depend on the behaviour model

Partial Components

```
def obs: stream I * cp(I,O) -> colist O
 = (s,p)  =>
   (| (s,p) => delist:
             { ff => ff
             | ss(q,o) => ss (o, (tail s, q))
             }  a(head s, p)
    |) (s,p).
```


Going Generic

Non Deterministic Components

```
data T -> cotree X = cot: T -> set(X * T).
```

$$
\begin{tikzcd}
\mathtt{cotree\ O} \arrow[r, "\omega_{\mathcal{P}(\mathtt{O} \times \mathsf{Id})}"] & \mathcal{P}(\mathtt{O} \times \mathtt{cotree\ O}) \\
\mathtt{stream\ I} \times \mathtt{cp(I,O)} \arrow[u, "\mathtt{obs} = [\![\mathtt{obs\_gen}]\!]"] \arrow[r, "\mathtt{obs\_gen}"] & \mathcal{P}(\mathtt{O} \times (\mathtt{stream\ I} \times \mathtt{cp(I,O)})) \arrow[u, "\mathsf{id} \times \mathtt{obs}"']
\end{tikzcd}
$$

where $\mathtt{obs\_gen}$ is the 'gene' coalgebra:

$$\mathsf{B}(\mathsf{id} \times \mathsf{s}) \cdot \mathsf{Ba} \cdot \mathsf{B}(\mathsf{s} \times \mathsf{id}) \cdot \tau_r \cdot (\mathsf{a} \times \mathsf{id}) \cdot \mathsf{xr} \cdot (\langle \mathtt{head}, \mathtt{tail} \rangle \times \mathsf{id})$$

Going Generic

```
def obs: stream I * cp(I,O) -> cotree O
 = (s,p)  => (| (s,p) => cot: B{fp{fid,swap}}
                B{assoc} B{fp{swap,fid}} taur
                (a(head s, p), tail s)
             |) (s, p).
```

Printout

```
>> obs(even, circle 5).
(cot: ...)

(cot: [(-3, (cot: ...)), (7, (cot: ...))])

(cot: [(-3, (cot: [(0, (cot: ...)), (8, (cot: ...))])
 (7, (cot: [(0, (cot: ...)), (8, (cot: ...))]))])
```


The Folder Example

```
def folder: cp (1+1+E, E+E+1) * cp (1+1+E, E+E+1)
            -> cp (1+1+1+E, E+1)

 = (lf, rg) =>
    wrap{b0, x => {b0 b0 b1 e => b0 e
                  |_         => b1 ()
                  } x
      }  hook wrap{wi,wo} choice(LeftS lf, RightS rg).
```

with

```
def LeftS: cp (1+1+E, E+E+1)  ->  cp (1+E, E+1)
    = s => wrap{fsum{b0, fid},
                fcomp{fsum{fid,bang}, sassoc} } s.
```

...

The Folder Example

Wrapping by the translator of a suitable, API-like interaction language:

```
data FLang E -> L =  turnr: 1 -> L | turnl: 1 -> L
                    | rd: 1 -> L | in: E -> L.
```

```
def vfolder: cp (1+1+1+E, E+1) ->  cp (FLang E, E+1)
    = f => wrap{ t =>  { turnr    => b0 b0 b0 ()
                        | turnl    => b0 b0 b1 ()
                        | rd    => b0 b1 ()
                        | in e => b1 e
                        } t, fid} f.
```


Separable Components

- Specified as collections of actions over a shared state space, each of which exhibits its own input and output types
- Independent actions are packed into a T^B-coalgebra leading to an *additive* interface such that each input produces a result whose type is known and unique (among the possible results).
- Arise typically in the practical use of model oriented specification methods, such as VDM or Z.

Separable Components

$p : I + J \longrightarrow O + R$ is separable if its dynamics $\overline{a}_p$ can be split into two independent coalgebras:

$$\overline{a}_{1.p} \; : U_p \longrightarrow \mathsf{B}(U_p \times O)^I \quad \text{and} \quad \overline{a}_{2.p} \; : U_p \longrightarrow \mathsf{B}(U_p \times R)^J$$

This notation suggests $1.p$ and $2.p$ as two independent components, with different interfaces, but defined over the same state space U_p and seed value u_p.
Therefore the dynamics of p arises as the currying of

$$a_p \;=\; [\mathsf{B}(\mathsf{id} \times \iota_1) \cdot a_{1.p}, \mathsf{B}(\mathsf{id} \times \iota_2) \cdot a_{2.p}] \cdot \mathsf{dr}$$

Separable Components

- For each B and objects I, J, O and R, separable components form a subcategory of $\mathsf{Cp}_{\mathsf{B}}(I + J, O + R)$.
- Let $p, q : I + J \longrightarrow O + R$ be separable.
  Then $h : p \longrightarrow q$ iff the same h, seen as an arrow in the underlying category, is also a comorphism from $1.p$ to $1.q$ and $2.p$ to $2.q$.
- Separability propagates under the component algebra
- ... leading to a handfull of new operators

Restriction

$$\blacktriangleright p : I \longrightarrow O \;=\; \langle u_p \in U_p, \overline{a}_{1.p} \rangle$$

$$\blacktriangleright h \;=\; h$$

for each $p : I + J \longrightarrow O + R$ separable

and

$$p^{\blacktriangleright} \;=\; \blacktriangleright(p[\mathsf{s}_+, \mathsf{s}_+])$$

Restriction Laws

For $p : I + J \longrightarrow O + R$, $q : O + R \longrightarrow P + T$ separable and $r : A \longrightarrow B$ any,

$$\begin{aligned}
\ulcorner \iota_1 \urcorner \,;\, p \;&\sim\; \blacktriangleright p \,;\, \ulcorner \iota_1 \urcorner \\
\blacktriangleright p \,;\, \blacktriangleright q \;&\sim\; \blacktriangleright(p \,;\, q) \\
\blacktriangleright p[f, f'] \;&\sim\; \blacktriangleright(p[f + g, f' + g']) \\
\blacktriangleright p \boxplus r \;&\sim\; \blacktriangleright((p \boxplus r)[\mathsf{xr}_+, \mathsf{xr}_+]) \\
\blacktriangleright p \boxtimes r \;&\sim\; \blacktriangleright((p \boxtimes r)[\mathsf{dl}^\circ, \mathsf{dl}]) \\
\blacktriangleright p \boxast r \;&\sim\; \blacktriangleright((p \boxast r)[(\mathsf{xr}_+ + \mathsf{dr}^\circ) \cdot \mathsf{m}_+, \mathsf{m}_+ \cdot (\mathsf{xr}_+ + \mathsf{dr})])
\end{aligned}$$

Restriction Laws

- For $p : I + Z \longrightarrow I + Z$ and $q : I + Z \longrightarrow O + Z$ separable,

$$
\begin{aligned}
({}^{\blacktriangleright}p) \Lsh &\sim {}^{\blacktriangleright}p \Lsh \\
({}^{\blacktriangleright}q) \Lsh_Z &\sim {}^{\blacktriangleright}q \Lsh_Z
\end{aligned}
$$

Variants on Feedback

- Hiding Feedback

$$\left(p\right)_Z \; : I \longrightarrow O \;=\; \langle u_p \in U_p, \overline{a}_{\left(p\right)_Z} \rangle$$

$$a_{\left(p\right)_Z} \;=\; U_p \times I \xrightarrow{a_{2.p} \bullet a_{1.p}} \mathsf{B}(U_p \times O)$$

for $p : I + Z \longrightarrow Z + O$ separable where $\bullet$ denotes the Kleisli composition for monad B.

A particular case of a separable component is $q \boxplus r : I + Z \longrightarrow Z + O$, where $q : I \longrightarrow Z$ and $r : Z \longrightarrow O$. Clearly,

$$\left(q \boxplus r\right)_Z \;\sim\; q \,;\, r$$

Variants on Feedback

- Hiding Hook Let $p : I + J + Z \longrightarrow Z + O + R$. separable into three 'threads' $1.p$, $2.p$ and $3.p$, with $1.p$ and $3.p$ composable.

$$\left(p\right)_Z : \mathsf{Cp}_{\mathsf{B}}(I+J+Z, Z+O+R)_{Sp} \longrightarrow \mathsf{Cp}_{\mathsf{B}}(I+J, O+R)_{Sp}$$

$$\left(p\right)_Z : I + J \longrightarrow O + R \;=\; \langle u_p \in U_p, \overline{a}_{\left(p\right)_Z} \rangle$$

with

$$a_{\left(p\right)_Z} \;=\; U_p \times (I+J) \xrightarrow{\mathsf{dr}} U_p \times I + U_p \times J$$
$$\xrightarrow{(a_{3.p} \bullet a_{1.p}) + a_{2.p}} \mathsf{B}(U_p \times R) + \mathsf{B}(U_p \times O)$$
$$\xrightarrow{[\mathsf{B}(\mathsf{id} \times \iota_2), \mathsf{B}(\mathsf{id} \times \iota_1)]} \mathsf{B}(U_p \times (O+R))$$

Separable Components

This definition subsumes the previous one: by isomorphic wiring the interface of

$$p : I + Z \longrightarrow Z + O$$

can be re-written as

$$I + \emptyset + O \longrightarrow Z + \emptyset + O$$

which is separable, with the original p threads in the first and third positions and $?_{\mathsf{B}(U_p \times \emptyset)}$ in the second.

Laws

- (feedback consistency)

$$\big(p\big)_Z ; \ulcorner \iota_1 \urcorner \sim \ulcorner \iota_1 \urcorner ; \big(p[\mathsf{id}, \mathsf{s}_+]\big) \Lsh_Z$$

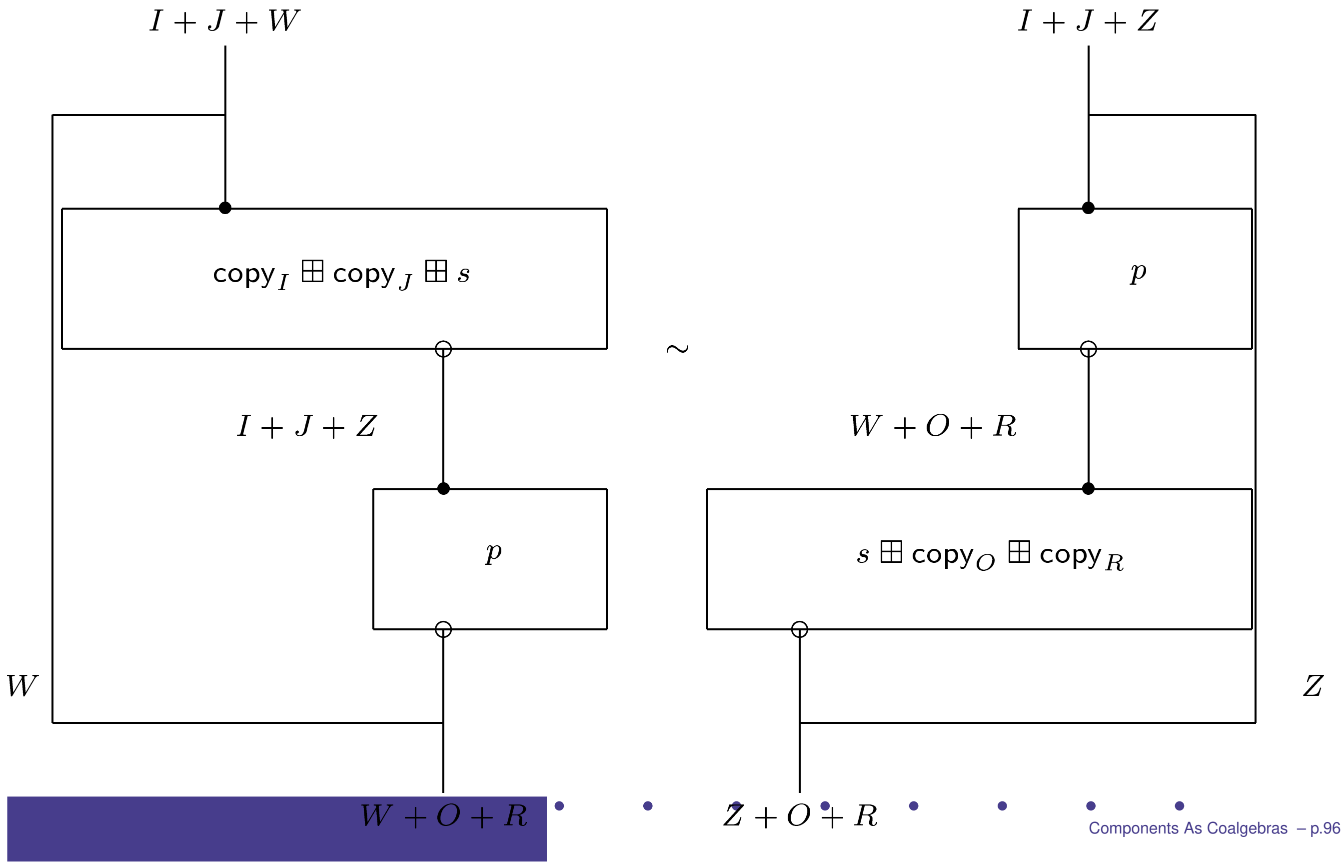
- (preserve structural wrapping)

$$\big(p[f+g+i, i^\circ+h+k]\big)_W \sim \big(p\big)_Z[f+g, k+h]$$
$$\big(\blacktriangleright p[\mathsf{xr}_+, \mathsf{xr}_+]\big)_Z \sim \blacktriangleright \big(p\big)_Z$$

- (pin swaping)

$$\big((\mathsf{copy}_I \boxplus \mathsf{copy}_J \boxplus s) ; p\big)_W \sim \big(p ; (s \boxplus \mathsf{copy}_O \boxplus \mathsf{copy}_R)\big)_Z$$
$$\big(((r ; s) \boxplus p)\,[\mathsf{a}_+, \mathsf{a}_+]\big)_Z \sim \big((r \boxplus ((\mathsf{copy}_I \boxplus s) ; p))\,[\mathsf{a}_+, \mathsf{a}_+]\big)_W$$

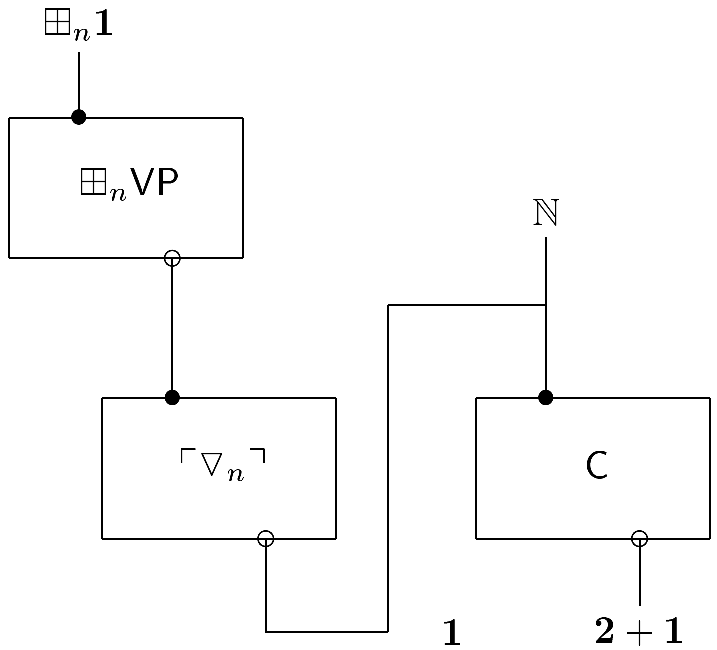
Laws



Proof Idea

- As p is separable, component s is activated in both expressions only once and always triggered by a p response to an I typed input
- A lengthy, but trivial, expression manipulation establishes $\mathsf{s} \bullet (\mathsf{s} \times \mathsf{id}) : \mathbf{1} \times \mathbf{1} \times U_s \times U_p \longrightarrow U_p \times (U_s \times \mathbf{1} \times \mathbf{1})$ as a comorphism from the left to the right hand side of the equation.

Ex: A Voting System



$$\mathsf{VS}_n \;=\; (((\boxplus_n \mathsf{VP} \,;\, \ulcorner \nabla_n \urcorner) \boxplus \mathsf{C})\,[\mathsf{a}_+, \mathsf{a}_+])_{\mathbf{1}} \quad : \quad \boxplus_n \mathbf{1} + \mathbb{N} \longrightarrow \mathbf{2} + \mathbf{1}$$

Ex: A Voting System

Voting Pad

$$\mathsf{VP} : \mathbf{1} \longrightarrow \mathbf{1} \quad = \quad \langle n \in \mathbb{N}, \overline{a}_{\mathsf{VP}} \rangle$$

where

$$a_{\mathsf{VP}}\ \langle n, *\rangle \ = (n \neq 0\ \rightarrow\ \iota_1\ \langle n-1, *\rangle,\ \iota_2\ *)$$

Concentrator

$$\mathsf{C} : \mathbb{N} + \mathbf{1} \longrightarrow \mathbf{1} + \mathbf{2} \quad = \quad \langle 0 \in \mathbb{N}, \overline{a}_{\mathsf{C}} \rangle$$

with

$$\begin{aligned} \mathsf{reset}\ \langle u, m\rangle \ &= \iota_1\ \langle m, *\rangle \\ \mathsf{vote}\ \langle u, *\rangle \ &= \iota_1\ \langle u-1, u' = 1\rangle \end{aligned}$$

Ex: A Voting System

Vote Chunking

- Replace $\boxplus$ by $\circledast$ as the 'gluing' combinator of the voting pads
- Re-designed vote to accept, instead of a single stimulus, a natural number encoding the number of incoming votes, ie,

  $$\mathsf{vote}\ \langle u, n\rangle \ = \iota_1\ \langle u-n, u' \leq 0\rangle$$

  defining a new concentrator $\mathsf{NC} : \mathbb{N} + \mathbb{N} \longrightarrow \mathbf{1} + \mathbf{2}$.
- Repalce codiagonal wiring by a counting function $\mathsf{count}_n : \circledast_n \mathbf{1} \longrightarrow \mathbb{N}$
- Assemble

  $$\mathsf{CVS}_n \ = \ (((\circledast_n \mathsf{VP}; \ulcorner \mathsf{count}_n \urcorner) \boxplus \mathsf{NC})\ [\mathsf{a}_+, \mathsf{a}_+])_{\mathbb{N}} \quad : \quad \circledast_n \mathbf{1} + \mathbb{N} \longrightarrow \mathbf{2} + \mathbf{1}$$

Ex: A Voting System

Property:
CVS_n exhibits a 'richer' behavioural pattern than VS_n, in a very precise sense:
if input to CVS_n is restricted to a sum of stimuli, the resulting component becomes bisimilar to VS_n.

- Proof for $n = 2$ with

$$\mathsf{count}_2 : (\mathbf{1} + \mathbf{1}) + \mathbf{1} \times \mathbf{1} \longrightarrow \mathbb{N} \;=\; [\underline{1}\cdot!, \underline{2}\cdot!]$$

Ex: A Voting System

$$
\begin{array}{ll}
 & \ulcorner \iota_1 + \mathsf{id} \urcorner \,;\, \mathsf{CVS}_2 \\
\sim & \quad \{ \text{ CVS}_2 \text{ definition and wrapping law } \} \\
 & (((\mathsf{VP} \boxast \mathsf{VP} \,;\, \ulcorner \mathsf{count}_n \urcorner) \boxplus \mathsf{NC})\,[\mathsf{a}_+, \mathsf{a}_+])_{\mathbb{N}}\,[\iota_1 + \mathsf{id}, \mathsf{id}] \\
\sim & \quad \{ \text{ structuring wrapping preservation law } \} \\
 & ((((\mathsf{VP} \boxast \mathsf{VP}) \,;\, \ulcorner \mathsf{count}_n \urcorner) \boxplus \mathsf{NC})\,[\mathsf{a}_+, \mathsf{a}_+][\iota_1 + \mathsf{id} + \mathsf{id}, \mathsf{id}])_{\mathbb{N}} \\
\sim & \quad \{ \text{ a}_+ \text{ natural and wrapping law } \} \\
 & ((\ulcorner \iota_1 + \mathsf{id} \urcorner \,;\, ((\mathsf{VP} \boxast \mathsf{VP}) \,;\, \ulcorner \mathsf{count}_2 \urcorner) \boxplus \mathsf{NC})\,[\mathsf{a}_+, \mathsf{a}_+])_{\mathbb{N}} \\
\sim & \quad \{ \boxplus \text{ on functions law and } \mathsf{copy}_X \text{ definition } \}
\end{array}
$$

Ex: A Voting System

$$(((\ulcorner \iota_1 \urcorner \boxplus \mathsf{copy}_N) \mathbin{;} ((\mathsf{VP} \circledast \mathsf{VP}) \mathbin{;} \ulcorner \mathsf{count}_2 \urcorner) \boxplus \mathsf{NC})\,[\mathsf{a}_+, \mathsf{a}_+])_{\mathbb{N}}$$

$\sim$ { $\boxplus$ lax functor }

$$(((\ulcorner \iota_1 \urcorner \mathbin{;} ((\mathsf{VP} \circledast \mathsf{VP}) \mathbin{;} \ulcorner \mathsf{count}_2 \urcorner)) \boxplus (\mathsf{copy}_N \mathbin{;} \mathsf{NC}))\,[\mathsf{a}_+, \mathsf{a}_+])_{\mathbb{N}}$$

$\sim$ { components bicategory }

$$(((\ulcorner \iota_1 \urcorner \mathbin{;} (\mathsf{VP} \circledast \mathsf{VP}) \mathbin{;} \ulcorner \mathsf{count}_2 \urcorner) \boxplus \mathsf{NC})\,[\mathsf{a}_+, \mathsf{a}_+])_{\mathbb{N}}$$

$\sim$ { $\boxplus$ from $\circledast$ law }

$$((((\mathsf{VP} \boxplus \mathsf{VP}) \mathbin{;} \ulcorner \iota_1 \urcorner \mathbin{;} \ulcorner \mathsf{count}_2 \urcorner) \boxplus \mathsf{NC})\,[\mathsf{a}_+, \mathsf{a}_+])_{\mathbb{N}}$$

$\sim$ { components bicategory }

Ex: A Voting System

$$((((\mathsf{VP} \boxplus \mathsf{VP}) \mathbin{;} (\ulcorner \iota_1 \urcorner \mathbin{;} \ulcorner \mathsf{count}_2 \urcorner)) \boxplus \mathsf{NC})\,[\mathsf{a}_+, \mathsf{a}_+])_{\mathbb{N}}$$

$\sim$ { $\mathsf{count}_2 \cdot \iota_1 = \underline{1} \cdot !_{1+1} = \underline{1} \cdot \nabla$ }

$$((((\mathsf{VP} \boxplus \mathsf{VP}) \mathbin{;} (\ulcorner \nabla \urcorner \mathbin{;} \ulcorner \underline{1} \urcorner)) \boxplus \mathsf{NC})\,[\mathsf{a}_+, \mathsf{a}_+])_{\mathbb{N}}$$

$\sim$ { pin swap law}

$$((((\mathsf{VP} \boxplus \mathsf{VP}) \mathbin{;} \ulcorner \nabla \urcorner) \boxplus ((\mathsf{copy}_{\mathbb{N}} \boxplus \ulcorner \underline{1} \urcorner) \mathbin{;} \mathsf{NC}))\,[\mathsf{a}_+, \mathsf{a}_+])_{\mathbf{1}}$$

$\sim$ { $(\mathsf{copy}_{\mathbb{N}} \boxplus \ulcorner \underline{1} \urcorner) \mathbin{;} \mathsf{NC} \sim \mathsf{C}$ }

$$((((\mathsf{VP} \boxplus \mathsf{VP}) \mathbin{;} \ulcorner \nabla \urcorner) \boxplus \mathsf{C})\,[\mathsf{a}_+, \mathsf{a}_+])_{\mathbf{1}}$$

$\sim$ { VS_2 definition }

$$\mathsf{VS}_2$$

Extensions & Further Work

- Heterogeneous composition pattterns and separate connectors [Meng, Barbosa]
- Prototyping: interconnection of RSL components
- Component classification and sub-calculi:
  - *separable* components
  - *restartable* components
  - ...
- Refinement at different levels (interface, behaviour, static)