Greatest Bisimulations for Binary Methods

Hendrik Tews

Institute of Theoretical Computer Science, TU Dresden

tews@tcs.inf.tu-dresden.de

http://wwwtcs.inf.tu-dresden.de/~tews/

Contents

I. Introduction
II. Previous related work
III. Review some definitions of "Coalgebras for binary Methods" [14]
IV. a structural condition (on signatures) for greatest bisimulations
V. a semantical condition (on coalgebras)
VI. Conclusion

Coalgebras can model Classes of object-oriented languages

class Point
  methods
    get_x : **Self** $\longrightarrow$ real
    get_y : **Self** $\longrightarrow$ real
    move : **Self** $\times$ real $\times$ real $\longrightarrow$ **Self**
end Point

Coalgebra $c : X \longrightarrow T_P(X)$ with

$$T(X) = \mathbb{R} \times \mathbb{R} \times (\mathbb{R} \times \mathbb{R} \Rightarrow X)$$

Abstract theory delivers

- **Bisimulation** as behavioural indistinguishability
- **Invariance**
- **Modal Operators** for an expressive logic
- **Coinduction** as definition and reasoning principle
- **Semantics for class specifications** using the final (most abstract) coalgebra

Built into the Coalgebraic Class Specification Language CCSL [12].

Binary Methods—Example

Class A = {

method equal(x : A) = x == this ;

method memory(env : $\mathbb{N} \longrightarrow A$) = $\cdots$ env(5).equal(this) $\cdots$;

method strange(f : A $\longrightarrow$ A) = $\cdots$;

}

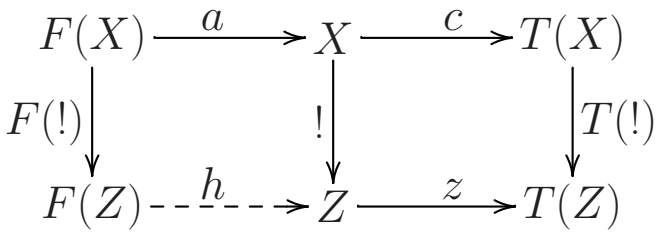
Assume, that X is the state space of the class

$$\begin{array}{lcl} \text{equal} & : & X \longrightarrow (X \longrightarrow \text{bool}) \\ \text{memory} & : & X \longrightarrow (\mathbb{N} \longrightarrow X) \longrightarrow \boxed{\cdots} \\ \text{strange} & : & X \longrightarrow (X \longrightarrow X) \longrightarrow \boxed{\cdots} \end{array}$$

1. Allow definitional extensions only

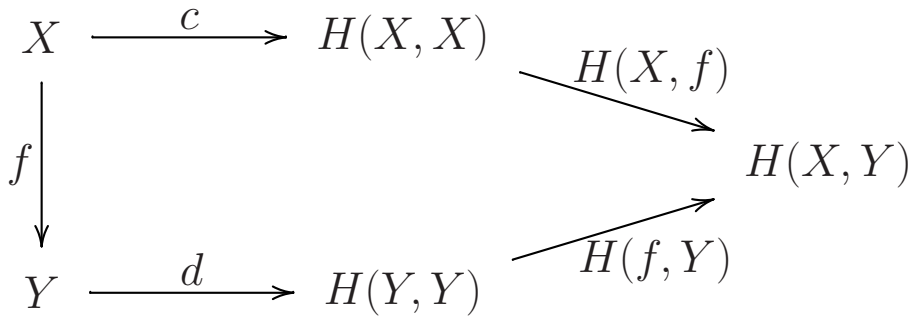
2. Use algebraic extensions

see [3]



3. Extend underlying framework see [14]

- use bivariant functors $H : \mathbf{Set}^{op} \times \mathbf{Set} \longrightarrow \mathbf{Set}$
- Coalgebras are morphisms $X \xrightarrow{c} H(X, X)$
- Morphism of coalgebras



- Many results in [14] but nothing about greatest bisimulations

4. Dialgebras from Poll & Zwanenburg in [10]

Consider functors $F : \mathbf{Set} \longrightarrow \mathbf{Set}$ built from the following grammar:

$$F \;=\; A \;\mid\; \mathsf{Id} \;\mid\; F_1 + F_2 \;\mid\; F_1 \times F_2$$

A Dialgebras is a family $f_i : F_i^I(X) \longrightarrow F_i^O(X)$

Theorem [Poll & Zwanenburg] *On dialgebras bisimulation equivalences form a complete lattice, the greatest bisimulation is* $\overline{\bigcup_i R_i}$.

Every dialgebra corresponds to a pair $\langle c, a \rangle$, where $c : X \longrightarrow G^K(X, X)$ is a coalgebra for some extended cartesian functor.

Partial Equivalence Relations

Let $R \subseteq X \times X$ be a relation.

Domain of R: $\mathsf{dom}(R) = \{x \mid \exists y \,.\, x \,R\, y \vee y \,R\, x\}$

Partial equivalence relation (PER) $\equiv$ symmetric and transitive
(i.e., an equivalence relation on its domain)

Least partial equivalence relation containing R is denoted with $\overline{R}$

Zigzag in R is a finite sequence

$x \,\overline{R}\, y \iff \exists$ a zigzag $x, \ldots, y$ in R

Extending polynomial functors

Class of Functors	Property	Example
cartesian $\mathbf{Set} \longrightarrow \mathbf{Set}$ $K \;=\; A \mid \mathsf{Id} \mid \mathsf{List} \mid K_1 \overset{+}{\times} K_2$	no arguments	$\mathbf{Self} \longrightarrow \boxed{\cdots}$
polynomial $\mathbf{Set} \longrightarrow \mathbf{Set}$ $F \;=\; A \mid \mathsf{Id} \mid \mathsf{List} \mid F_1 \overset{+}{\times} F_2 \mid A \Rightarrow F_1$	constant arguments	$\mathbf{Self} \times A \longrightarrow \boxed{\cdots}$
extended cartesian $\mathbf{Set}^{op} \times \mathbf{Set} \longrightarrow \mathbf{Set}$ $G^K \;=\; A \mid \mathsf{Id} \mid \mathsf{List} \mid G_1^K \overset{+}{\times} G_2^K \mid K \Rightarrow G_1^K$	no functional arguments	$\mathbf{Self} \times \mathbf{Self} \times \mathbf{Self} \longrightarrow \boxed{\cdots}$ $\mathbf{Self} \times (\mathbf{Self} + A) \longrightarrow \boxed{\cdots}$
extended polynomial $\mathbf{Set}^{op} \times \mathbf{Set} \longrightarrow \mathbf{Set}$ $G^F \;=\; A \mid \mathsf{Id} \mid \mathsf{List} \mid G_1^F \overset{+}{\times} G_2^F \mid F \Rightarrow G_1^F$	functional arguments with constant domain	$\mathbf{Self} \times (A \Rightarrow \mathbf{Self}) \longrightarrow \boxed{\cdots}$
higher-order polynomial $\mathbf{Set}^{op} \times \mathbf{Set} \longrightarrow \mathbf{Set}$ $H \;=\; A \mid \mathsf{Id} \mid \mathsf{List} \mid H_1 \overset{+}{\times} H_2 \mid H_1 \Rightarrow H_2$	arbitrary arguments	$\mathbf{Self} \times (\mathbf{Self} \Rightarrow A) \longrightarrow \boxed{\cdots}$ $\mathbf{Self} \times (\mathbf{Self} \Rightarrow \mathbf{Self}) \longrightarrow \boxed{\cdots}$

Relation lifting (for a h.-o. pol. functor H)

is an operation on relations

$$\big(S \subseteq U \times V, \quad R \subseteq X \times Y\big) \longmapsto \mathsf{Rel}(H)(S,R) \subseteq H(U,X) \times H(V,Y)$$

Defined by induction on the structure of H:

H	$\mathrm{Rel}(H)$	$a \in H(U,X) \quad b \in H(V,Y)$
Id	R	$R(a,b)$
A	$=_A$	$a =_A b$
$\mathsf{List} \circ H_1$	$\mathsf{RelList}(\mathrm{Rel}(H_1))$	$\mathsf{len}(a) = \mathsf{len}(b) \ \wedge$ $\forall i \,.\, \mathrm{Rel}(H_1)(S,R)(\mathsf{nth}(a,i),\ \mathsf{nth}(b,i))$
$H_1 \times H_2$	$\mathrm{Rel}(H_1) \times_{\mathrm{R}} \mathrm{Rel}(H_2)$	$\mathsf{Rel}(H_1)(S,R)(\pi_1\, a, \pi_1\, b) \ \wedge \ \mathsf{Rel}(H_2)(S,R)(\pi_2\, a, \pi_2\, b)$
$H_1 + H_2$	$\mathrm{Rel}(H_1) +_{\mathrm{R}} \mathrm{Rel}(H_2)$	$\begin{cases} \mathsf{if}\ (a,b) = (\kappa_1\, a_1, \kappa_1\, b_1) & \mathsf{Rel}(H_1)(S,R)(a_1,b_1) \\ \mathsf{if}\ (a,b) = (\kappa_2\, a_2, \kappa_2\, b_2) & \mathsf{Rel}(H_2)(S,R)(a_2,b_2) \\ \mathsf{otherwise} & \bot \end{cases}$
$H_1 \Rightarrow H_2$	$\mathrm{Rel}(H_1) \Rightarrow_{\mathrm{R}} \mathrm{Rel}(H_2)$	$\forall (u,v) \in \mathsf{Rel}(H_1)(R,S) \ . \ \mathsf{Rel}(H_2)(S,R)(a(u), b(v))$

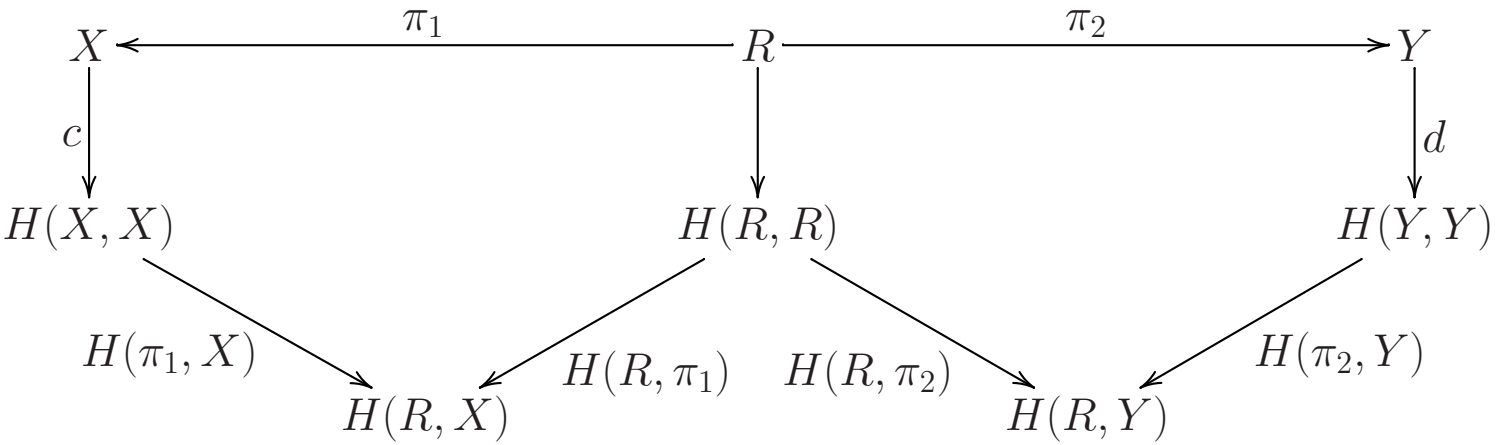
Consider a higher-order polynomial functor H and two coalgebras $c : X \longrightarrow H(X, X)$ and $d : Y \longrightarrow H(Y, Y)$.

Bisimulations according to Hermida & Jacobs

- R is a bisimulation for c and d, if

$$\forall x \in X, y \in Y \;.\; R(x, y) \quad \text{implies} \quad \mathrm{Rel}(H)(R, R)(c(x), d(y))$$

Bisimulations according to Aczel & Mendler



Both notions coincide for extended polynomial functors
(in general they are different)

Example: Union of Bisimulations

Functor $K(Y,X) \stackrel{\text{def}}{=} Y \Rightarrow \text{bool}$

corresponding to one method equal : **Self** $\times$ **Self** $\longrightarrow$ bool

K–bisimulation $R \subseteq X \times X \quad c : X \longrightarrow X \Rightarrow \text{bool}$

$$\forall x, x', y, y' : X \,.\, x\, R\, x' \;\wedge\; y\, R\, y' \;\Longrightarrow\; c\, x\, y \;=\; c\, x'\, y'$$

Defining a K–coalgebra $X \stackrel{\text{def}}{=} \{x_1, x_2\}$

$$c \;:\; X \longrightarrow X \Rightarrow \text{bool} \qquad\qquad c\, x\, y \stackrel{\text{def}}{=} \text{if } x = y \text{ then } \top \text{ else } \bot \text{ endif}$$

- $\mathrm{Eq}(X) = \{(x_1, x_1),\, (x_2, x_2)\}$ is a bisimulations for c
- $R \subseteq X \times X \stackrel{\text{def}}{=} \{(x_1, x_2), (x_2, x_1)\}$ is a bisimulation for c
- However, $\mathrm{Eq}(X) \cup R = X \times X$ is not a bisimulation

$$\top \quad = \quad c\, x_1\, x_1 \quad \neq \quad c\, x_1\, x_2 \quad = \quad \bot$$

Consider merge3 : **Self** $\times$ **Self** $\times$ **Self** $\longrightarrow$ **Self**

Three bisimulations $R_1, R_2, R_3 \subseteq$ **Self** $\times$ **Self**

with $x\, R_1\, x'$, $y\, R_2\, y'$, $z\, R_3\, z'$

Intuitive reasoning

$$
\begin{array}{cc}
 & \mathsf{merge3}(x, y, z) \\
x\, R_1\, x' & \| \wr \\
 & \mathsf{merge3}(x', y, z) \\
y\, R_2\, y' & \| \wr \\
 & \mathsf{merge3}(x', y', z) \\
z\, R_3\, z' & \| \wr \\
 & \mathsf{merge3}(x', y', z')
\end{array}
$$

$$
x\, R_1\, x' \;\wedge\; y\, R_2\, y' \;\wedge\; z\, R_3\, z' \;\Longrightarrow\; \mathsf{merge3}(x, y, z) \;\cong\; \mathsf{merge3}(x', y', z')
$$

Implicit assumptions

- $y\, R_1\, y$, $z\, R_1\, z$, $x\, R_2\, x$, ... further $\mathsf{dom}(R_1) = \mathsf{dom}(R_2) = \mathsf{dom}(R_3)$
- bisimulations are partial equivalence relations
- $((x, y, z), (x', y', z')) \in \mathrm{Rel}(\cdot\cdot)(\overline{\bigcup_i R_i})$ implies $\exists$ zigzag in $\overline{\bigcup_i \mathrm{Rel}(\cdot\cdot)(R_i)}$

Assumption: All relations are PER's (on the same domain)!

Lemma 1 $\mathrm{Rel}(H)(S,R)$ *is a PER for all higher-order polynomial functors* H*.*

Lemma 2 *For all polynomial functors* F*:*

$$\mathsf{dom}(R_1) \;=\; \mathsf{dom}(R_2) \qquad \text{implies} \qquad \mathsf{dom}(\mathrm{Rel}(F)(R_1)) \;=\; \mathsf{dom}(\mathrm{Rel}(F)(R_2))$$

Lemma 3

$$\begin{aligned} \overline{S +_{\mathrm{R}} R} &= \overline{S} +_{\mathrm{R}} \overline{R} \\ \overline{S \times_{\mathrm{R}} R} &= \overline{S} \times_{\mathrm{R}} \overline{R} \\ \overline{\mathrm{RelList}(R)} &= \mathrm{RelList}\big(\overline{R}\big) \end{aligned}$$

Lemma 4

$$\begin{aligned} \overline{\textstyle\bigcup_i (S_i +_{\mathrm{R}} R_i)} &= \overline{(\textstyle\bigcup_i S_i) +_{\mathrm{R}} (\bigcup_i R_i)} \\ \overline{\textstyle\bigcup_i (S_i \times_{\mathrm{R}} R_i)} &= \overline{(\textstyle\bigcup_i S_i) \times_{\mathrm{R}} (\bigcup_i R_i)} \\ \overline{\textstyle\bigcup_i \mathrm{RelList}(R_i)} &= \overline{\mathrm{RelList}(\textstyle\bigcup_i R_i)} \end{aligned}$$

Lemma 5 *Assume*

- *a cartesian functor* K
- *a family* $(R_i \subseteq X \times X)$ *of PER's on the same domain*

Then

$$\mathrm{Rel}(K)(\overline{\bigcup_i R_i}) \quad = \quad \overline{\bigcup_i \mathrm{Rel}(K)(R_i)}$$

Proof for $K = K_1 \times K_2$:

$$\begin{aligned}
\mathrm{Rel}(K_1 \times K_2)(\overline{\textstyle\bigcup_i R_i}) &= \mathrm{Rel}(K_1)(\overline{\textstyle\bigcup_i R_i}) \times_{\mathrm{R}} \mathrm{Rel}(K_2)(\overline{\textstyle\bigcup_i R_i}) \\
&= \overline{\textstyle\bigcup_i \mathrm{Rel}(K_1)(R_i)} \times_{\mathrm{R}} \overline{\textstyle\bigcup_i \mathrm{Rel}(K_2)(R_i)} \\
&= \overline{\textstyle\bigcup_i \mathrm{Rel}(K_1)(R_i) \times_{\mathrm{R}} \bigcup_i \mathrm{Rel}(K_2)(R_i)} \\
&= \overline{\textstyle\bigcup_i \left(\mathrm{Rel}(K_1)(R_i) \times_{\mathrm{R}} \mathrm{Rel}(K_2)(R_i)\right)} \\
&= \overline{\textstyle\bigcup_i \mathrm{Rel}(K_1 \times K_2)(R_i)} \qquad \square
\end{aligned}$$

Lemma 6 *Let G be an extended cartesian functor and*

- $(S_i) \subseteq Y \times Y$, $(R_i) \subseteq X \times X$ *families of PER's on the same domain*
- $\exists j \,.\, (s, r) \in \mathrm{Rel}(G)(S_j, R_j)$
- $\forall i \,.\, (r, r) \in \mathrm{Rel}(G)(S_i, R_i)$
- $\forall i \,.\, (s, s) \in \mathrm{Rel}(G)(S_i, R_i)$

then also $(s, r) \in \mathrm{Rel}(G)(\overline{\bigcup_i S_i}, \overline{\bigcup_i R_i})$

Proof for $G = K \Rightarrow G_1$: Assuming

$$\begin{array}{lll}
\exists j \,.\, \forall a, b \in K(Y) \,. & \mathrm{Rel}(K)(S_j)(a, b) & \text{implies} \quad \mathrm{Rel}(G_1)(S_j, R_j)(s\,a, r\,b) \\
\forall i \,.\, \forall a, b \in K(Y) \,. & \mathrm{Rel}(K)(S_i)(a, b) & \text{implies} \quad \mathrm{Rel}(G_1)(S_i, R_i)(s\,a, s\,b) \\
\forall i \,.\, \forall a, b \in K(Y) \,. & \mathrm{Rel}(K)(S_i)(a, b) & \text{implies} \quad \mathrm{Rel}(G_1)(S_i, R_i)(r\,a, r\,b)
\end{array}$$

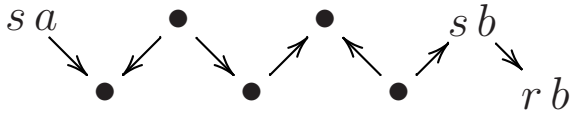
it remains to show

$$\forall a, b \in K(Y) \,. \quad \mathrm{Rel}(K)(\overline{\textstyle\bigcup_i S_i})(a, b) \quad \text{implies} \quad \mathrm{Rel}(G_1)(\overline{\textstyle\bigcup_i S_i}, \overline{\textstyle\bigcup_i R_i})(s\,a,\, r\,b)$$

Let $(a, b) \in \mathrm{Rel}(K)(\overline{\bigcup_i S_i})$:

in $\bigcup_i \mathrm{Rel}(K)(S_i)$

this gives



in $\mathrm{Rel}(G_1)(\overline{\bigcup_i S_i}, \overline{\bigcup_i R_i})$ □

Theorem 7 *For a coalgebra of an extended cartesian functor the partial bisimulation equivalences on a fixed domain form a complete lattice. (The join of a family (R_i) of bisimulations is $\overline{\bigcup_i R_i}$.)*

Proof Let $c : X \longrightarrow G(X, X)$ for an extended cartesian functor G:

- if R_i is a bisimulation then $(c, c) \in \mathrm{Rel}(\mathsf{Id} \Rightarrow G)(R_i, R_i)$
- thus, $(c, c) \in \mathrm{Rel}(\mathsf{Id} \Rightarrow G)(\overline{\bigcup_i R_i}, \overline{\bigcup_i R_i})$ □

Functor $G(Y,X) \stackrel{\text{def}}{=} (\mathbb{N} \Rightarrow Y) \Rightarrow \mathsf{bool}$

Defining a G–coalgebra on $\mathbb{N}$

$$c \;:\; \mathbb{N} \longrightarrow (\mathbb{N} \Rightarrow \mathbb{N}) \Rightarrow \mathsf{bool} \qquad\qquad c\,x\,f \;=\; \begin{cases} \top & \exists n \,.\, \forall i \,.\, f\,i < n \\ \bot & \text{otherwise} \end{cases}$$

G–bisimulation for c $\quad R \subseteq \mathbb{N} \times \mathbb{N}$

$$\forall x, y : \mathbb{N},\, f, g : \mathbb{N} \longrightarrow \mathbb{N} \,.\, x\,R\,y \;\wedge\; \big(\forall n \,.\, (f\,n)\,R\,(g\,n)\big) \;\Longrightarrow\; c\,x\,f \;=\; c\,y\,g$$

Family $(S_n \subseteq \mathbb{N} \times \mathbb{N})_{n \in \mathbb{N}}$

$$i\,S_n\,j \quad\Longleftrightarrow\quad i = j \;\vee\; (i < n \wedge j < n)$$

- all S_n are bisimulations
- every *finite* subset of S_n's has an upper bound (which is a bisimulation)
- $\bigcup_n S_n \;=\; \mathbb{N} \times \mathbb{N}$ is *not* a bisimulation, take $f_0 : \mathbb{N} \longrightarrow \mathbb{N}$ with $f_0\,n = 0$ and $\mathsf{id}_{\mathbb{N}}$: $\forall n \,.\, (n, 0) \in \bigcup_n S_n$ but

$$\top \quad = \quad c\;0\;\mathsf{id}_{\mathbb{N}} \quad \neq \quad c\;0\;f_0 \quad = \quad \bot$$

Problem $(f_0, \mathsf{id}_{\mathbb{N}}) \in \left(\mathrm{Eq}(\mathbb{N}) \Rightarrow_{\mathrm{R}} \overline{\bigcup_n S_n}\right)$ but $(f_0, \mathsf{id}_{\mathbb{N}}) \notin \overline{\bigcup_n (\mathrm{Eq}(\mathbb{N}) \Rightarrow_{\mathrm{R}} S_n)}$

Observation Let $f, g : \mathbb{N} \longrightarrow \mathbb{N}$

If $\{a \mid f\, a \neq f\, b\}$ is finite and $I \neq \emptyset$ + the usual assumptions

$$(f, g) \in \left(\mathrm{Eq}(A) \Rightarrow_{\mathrm{R}} \overline{\bigcup_i R_i}\right) \implies (f, g) \in \overline{\bigcup_i (\mathrm{Eq}(A) \Rightarrow_{\mathrm{R}} R_i)}$$

For discussion

- In the context of the preceding example assume a coalgebra

$$d : X \longrightarrow (\mathbb{N} \Rightarrow X) \Rightarrow \mathsf{bool}$$

with the following property $\forall x : X, f : \mathbb{N} \longrightarrow X$:

$$\exists A \subseteq \mathbb{N} \,.\, A \text{ is finite } \wedge\ \forall g : \mathbb{N} \longrightarrow X \,.\, f|_A = g|_A \implies d\, x\, f = d\, x\, g$$

- $\forall (f, g) \in \left(\mathrm{Eq}(A) \Rightarrow_{\mathrm{R}} \overline{\bigcup_i R_i}\right)$ $\exists f'$ with $d\, x\, f = d\, x\, f'$ and $(f', g) \in \overline{\bigcup_i (\mathrm{Eq}(A) \Rightarrow_{\mathrm{R}} R_i)}$
- d has a greatest bisimulation!

Argument descriptor desc^F for a polynomial functor F

$$\begin{aligned}
\mathsf{desc}^A = \mathsf{desc}^{\mathsf{ID}} &= \mathbf{1} = \{*\} \\
\mathsf{desc}^{\mathsf{List}(F_1)} &= \mathsf{List}(\mathsf{desc}^{F_1}) \\
\mathsf{desc}^{F_1+F_2} &= \mathsf{desc}^{F_1} \times \mathsf{desc}^{F_2} \\
\mathsf{desc}^{F_1\times F_2} &= \mathsf{desc}^{F_1} \times \mathsf{desc}^{F_2} \\
\mathsf{desc}^{A\Rightarrow F_1} &= \{R \subseteq A \times \mathsf{desc}^{F_1} \mid a\,R\,b_1 \wedge a\,R\,b_2 \ \text{ implies } \ b_1 = b_2\}
\end{aligned}$$

Environment

F	$\mathsf{env}^D(x) \subseteq F(X)$ for $x \in F(X)$
A / Id	$\{x\}$
$F_1 \times F_2$	$\mathsf{env}^{\pi_1 D}(\pi_1\,x) \times \mathsf{env}^{\pi_2 D}(\pi_2\,x)$
$F_1 + F_2$	if $x = \kappa_1\,x_1$: $\{\kappa_1\,y \mid y \in \mathsf{env}^{\pi_1 D}(x_1)\}$ if $x = \kappa_2\,x_2$: $\{\kappa_2\,y \mid y \in \mathsf{env}^{\pi_2 D}(x_2)\}$
$\mathsf{List}(F_1)$	if $x = \mathsf{nil}$: $\{\mathsf{nil}\}$ if $\left\{\begin{array}{l} x = \mathsf{cons}(a_1, x') \wedge \\ D = \mathsf{cons}(A, D') \end{array}\right\}$: $\{\mathsf{cons}(a_2, y') \mid a_2 \in \mathsf{env}^A(a_1) \wedge y' \in \mathsf{env}^{D'}(x')\}$ if $D = \mathsf{nil}$: $\{y \mid \mathsf{length}\,x = \mathsf{length}\,y\}$
$A \Rightarrow F_1$	$\{y \mid \forall (a, D') \in D\,.\,y\,a \in \mathsf{env}^{D'}(x\,a)\}$

Finitely Based for an extended polynomial functor G

G	$x \in G(Y, X)$ is finitely based if
A	always
Id	always
List	all list elements are finitely based
$G_1 \times G_2$	$\pi_1\, x$ and $\pi_2\, x$ are finitely based
$G_1 + G_2$	$x = \kappa_1\, x_1$ implies x_1 is finitely based $x = \kappa_2\, x_2$ implies x_2 is finitely based
$F \Rightarrow G_1$	• $\forall a \in F(Y)$ the result $x(a)$ is finitely based • $\forall a \in F(Y)\ .\ \exists$ hereditarily finite $D \in \mathsf{desc}^F$ such that $\forall a' \in \mathsf{env}^D(a)\ .\ x\, a = x\, a'$

A coalgebra $c : X \longrightarrow G(X, X)$ is finitely based if
it is finitely based as $c \in\ G'(X, X)\ =\ X \Rightarrow G(X, X)$.

Coalgebras for extended cartesian functors are *always* finitely based.

Lemma 8 *Assume*

- *a polynomial functor* F
- *nonempty index set* I
- *a family* $(R_i \subseteq X \times X)$ *of PER's on the same domain*
- $D \in \mathsf{desc}^F$ *hereditarily finite*

For all $x, y \in F(X)$ *there exists* $x' \in \mathsf{env}^D(x)$ *such that*

$$(x, y) \;\in\; \mathrm{Rel}(F)\big(\overline{\textstyle\bigcup_i R_i}\big) \qquad \textit{implies} \qquad (x', y) \;\in\; \overline{\textstyle\bigcup_i \mathsf{Rel}(F)(R_i)}$$

Lemma 9 *Let* G *be an extended* **polynomial** *functor and*

- $(S_i) \subseteq Y \times Y$, $(R_i) \subseteq X \times X$ *families of PER's on the same domain* **for** $\boldsymbol{I}$ **nonempty**
- $\exists j \,.\, (s, r) \in \mathrm{Rel}(G)(S_j, R_j)$ **with** $\boldsymbol{s}$ **finitely based**
- $\forall i \,.\, (r, r) \in \mathrm{Rel}(G)(S_i, R_i)$
- $\forall i \,.\, (s, s) \in \mathrm{Rel}(G)(S_i, R_i)$

then also $(s, r) \in \mathrm{Rel}(G)(\overline{\bigcup_i S_i}, \overline{\bigcup_i R_i})$

Proof almost identical, use preceding lemma to get the zigzag in $\bigcup_i \mathrm{Rel}(F)(S_i)$ □

Theorem 10 *For a* **finitely based** *coalgebra of an extended* **polynomial** *functor the partial bisimulation equivalences on a fixed domain form a complete lattice. (The join of a family* (R_i) *of bisimulations is* $\overline{\bigcup_i R_i}$*.)*

Two place Buffer I

Begin CartBuf[A : **Type**] : **ClassSpec**

Method
put : **Self** × (A + **1**) –> **Self**;
get : **Self** –> A + **1**;

Assertion SelfVar x : **Self**
get_put : **Forall**(a : A) . get(put(x, up a)) = up a;

attribute
next_buffer : **Self** –> Addr;

method
push : **Self** × list[nat × **Self**] –> **Self** × list[nat × **Self**];

end CartBuf

Two place Buffer II

Begin Buf[A : **Type**] : **ClassSpec**

Method
put : **Self** × (A + **1**) –> **Self**;
get : **Self** –> A + **1**;

Assertion SelfVar x : **Self**
get_put : **Forall**(a : A) . get(put(x, up a)) = up a;

attribute
next_buffer : **Self** –> Addr;

method
push : **Self** × [Addr –> **Self** + **1**] –> **Self** × [Addr –> **Self** + **1**];

assertion SelfVar x : **Self**
finitely_based : **Forall**(mem1, mem2 : [Addr –> **Self** + **1**]) :
mem1(next_buffer(x)) = mem2(next_buffer(x))
IMPLIES
push(x, mem1) = push(x, mem2);

end Buf

Open problems

- w.r.t. List as an ingredient functor:
  How about arbitrary least fixed points?
- How about bisimulations $R \subseteq X \times Y$?

[3] Hennicker, R. and A. Kurz, (Ω, Ξ)*–Logic: On the algebraic extension of coalgebraic specifications*, in: B. Jacobs and J. Rutten, editors, *Coalgebraic Methods in Computer Science '99*, ENTCS **19** (1999), pp. 195–212.

[10] Poll, E. and J. Zwanenburg, *From algebras and coalgebras to dialgebras*, in: A. Corradini, M. Lenisa and U. Montanari, editors, *Coalgebraic Methods in Computer Science '01*, ENTCS **44** (2001).

[12] Rothe, J., H. Tews and B. Jacobs, *The coalgebraic class specification language CCSL*, Journal of Universal Computer Science **7** (2001), pp. 175–193.

[14] Tews, H., *Coalgebras for binary methods: Properties of bisimulations and invariants*, Theoretical informatics and applications **35** (2001), pp. 83–111.