Structure and Behaviour – Dual System Aspects

Uwe Wolter

University of Bergen, Norway

PAMT 2014, May 27th 2014, Bergen, Norway

Content

Motivation

Personal Summary

I covered in teaching and research a broad spectrum of formal specifications. Structure and behaviour are dual phenomena. My personal conclusion is that Algebraic Specifications are adequate only for static, structural aspects. Dually, coalgebras are more appropriate do describe and reason about behaviour.

Aim and Vision

Develop a better understanding of the duality between structure and behaviour by working out some more examples. Try to find an appropriate formalization of this duality. Based on this (and in parallel) develop a generic and well-founded methodolgy for combining specifications of structure and of behaviour, respectively.

Part-Whole Relation

Observations

- The direction of a binary relationship is always a matter of taste and arbitrariness. Look, for example, at inheritance.
- If we decide, however, for a certain concept to describe binary relationships, as total functions for example, it appears that the part-whole relationship goes in opposite directions w.r.t. structure or behaviour, respectively.
- Turning the direction of a relationship, however, is exactly what constitutes **duality** in orders and categories.

Working Hypothesis

Structure and Behaviour are **dual** phenomena.

Part-Whole Relation

Structure

Each constituent of a subsystem is also a constituent of the (overall) system. That is, structurally the part-whole relation is an inclusion map from the part to the whole.

$$Part \xrightarrow[inclusion]{structure} Whole$$

State and Behaviour

Each state of a system comprises a corresponding unique state of a subsystem. In contrast, a state of a certain subsystem doesn't determine a unique state of the system. The same holds for state transitions. That is, behaviourally the part-whole relation is a projection map from the whole to the part.

$$Part \xleftarrow[projection]{behaviour} Whole$$

Construction of Systems

Question

What consequences has the duality of structure and behaviour for the construction of systems out of subsystems?

Working Hypothesis

The construction of systems out of subsystems is reflected by dual constructions on the structural level and the behavioural level, respectively.

Example – Puting together memory

As simple **structures** we consider two sets M_1 und M_2 of memory addresses. In case $M_1 \cap M_2 = \emptyset$, the union of the two sets is a categorical sum (coproduct).
As **states** we consider sets of assignments of bits $2 = \{0, 1\}$ to the addresses. In such a way, the set of states of the system becomes the Cartesian product of the states of the two substems.

Structure **State**

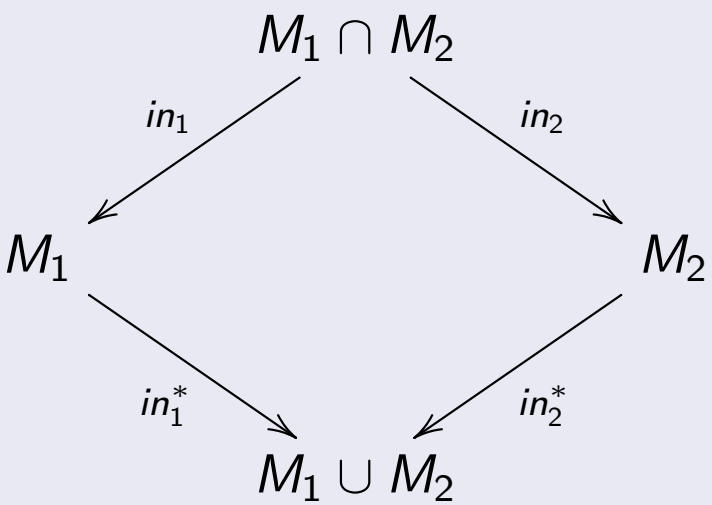
$$M_1 \xrightarrow{in_1} M_1 + M_2 \xleftarrow{in_2} M_2 \qquad 2^{M_1} \xleftarrow{pr_1} 2^{M_1+M_2} \xrightarrow{pr_2} 2^{M_2}$$

This follows from the isomorphy $2^{M_1+M_2} \cong 2^{M_1} \times 2^{M_2}$.

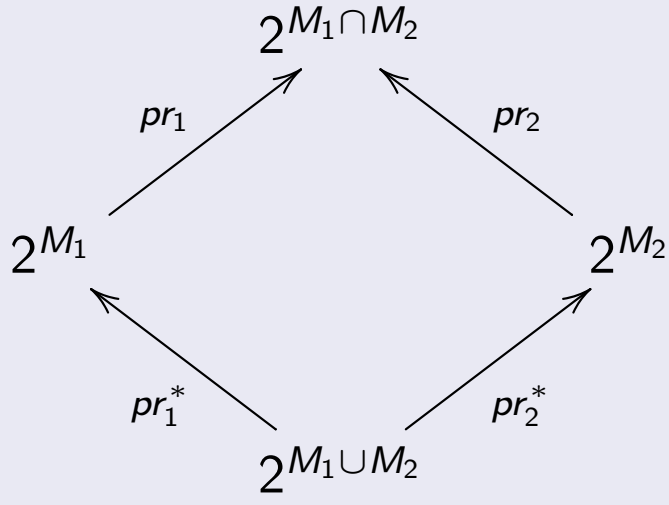
Example – Puting together memory

In case of **sharing**, i.e., $M_1 \cap M_2 \neq \emptyset$, we have a pushout of structures and a pullback of sets of states.

Structure



States



This follows from $2^{M_1 \cup M_2} \cong \{(b_1, b_2) \in 2^{M_1} \times 2^{M_2} \mid pr_1(b_1) = pr_2(b_2)\}$.

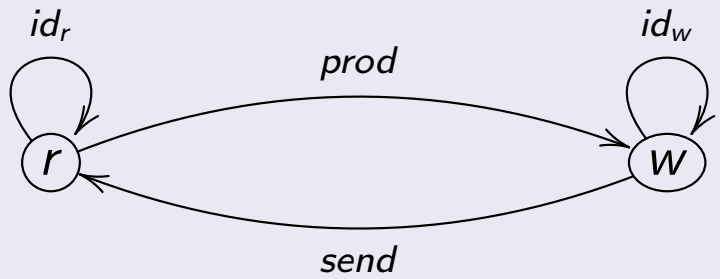
Example – Program design (á la CommUnity)

The **structure** of an abstract program design is given by a set V of "communication channels". The **behaviour** of an abstract program we describe by a set Γ of action names and a finite state automaton.

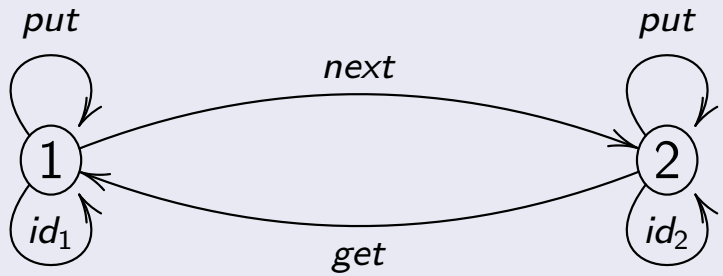
SENDER:

Structure

Behaviour



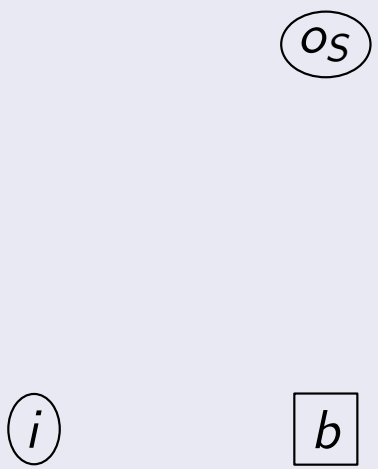
BUFFER:



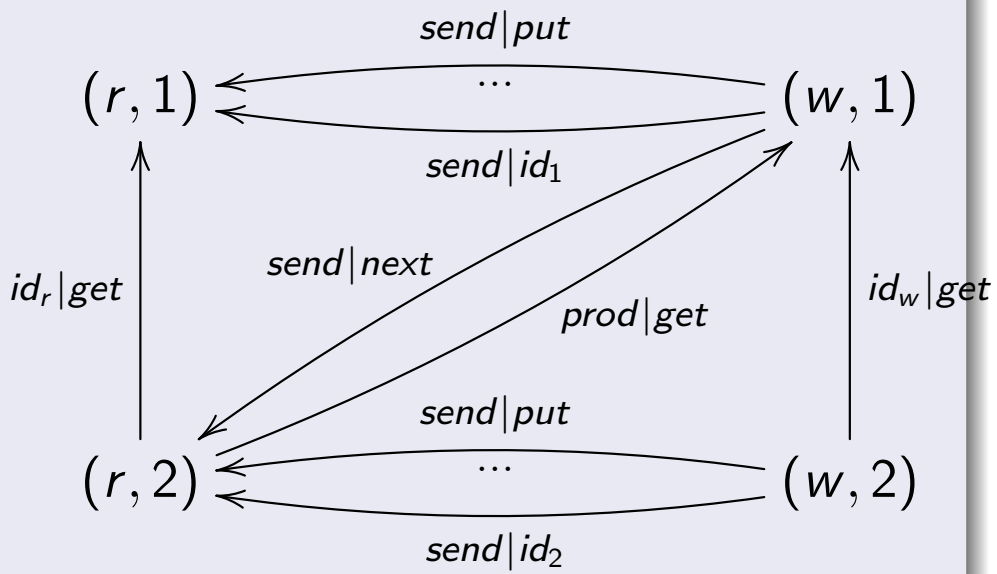
Example – Sum of Sender and Buffer

If we just screw together the two systems Sender and Buffer, we do have structurally a sum (disjoint union) SENDER+BUFFER and behaviourally we have a product automaton $A_S \times A_P$ with $4 = 2 \times 2$ states and $24 = 4 \times 6$ parallel actions.

Structure



Behaviour



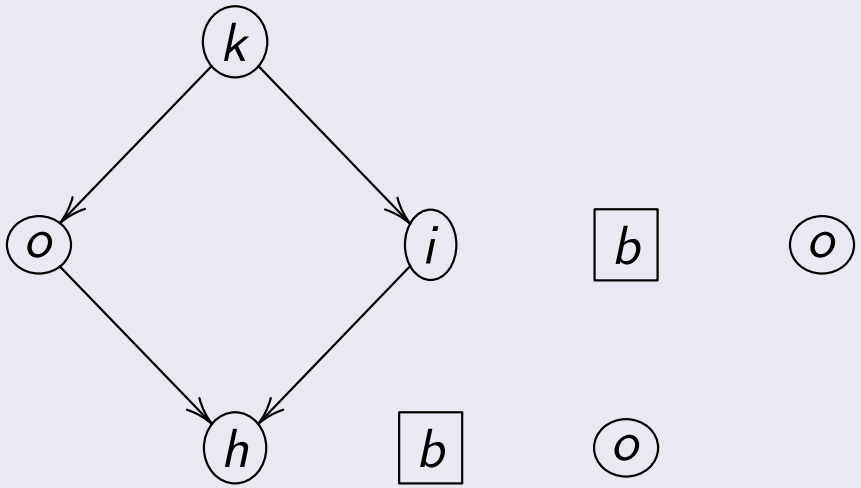
Example – Connecting Sender and Buffer by a Cable

What happens, if we connect the out channel of the sender with the in channel of the buffer by means of a cable"?

CABLE: 

Structure – Pushout

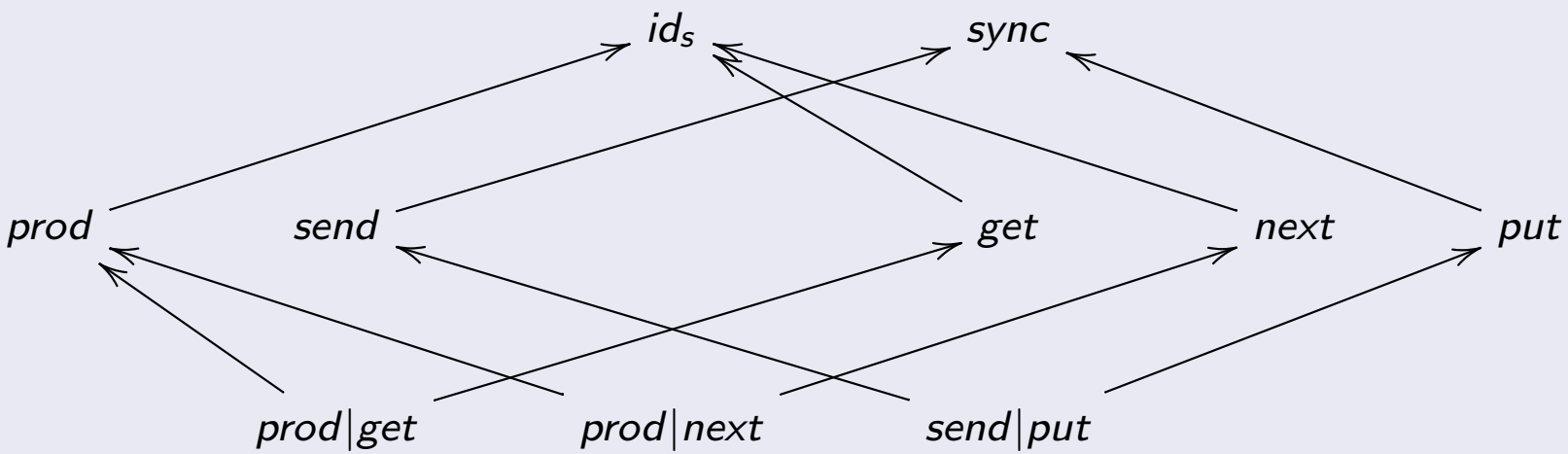
The out channel of the sender and the in channel of the buffers are identifies, i.e., we construct a pushout.



Example – Connecting Sender and Buffer by a Cable

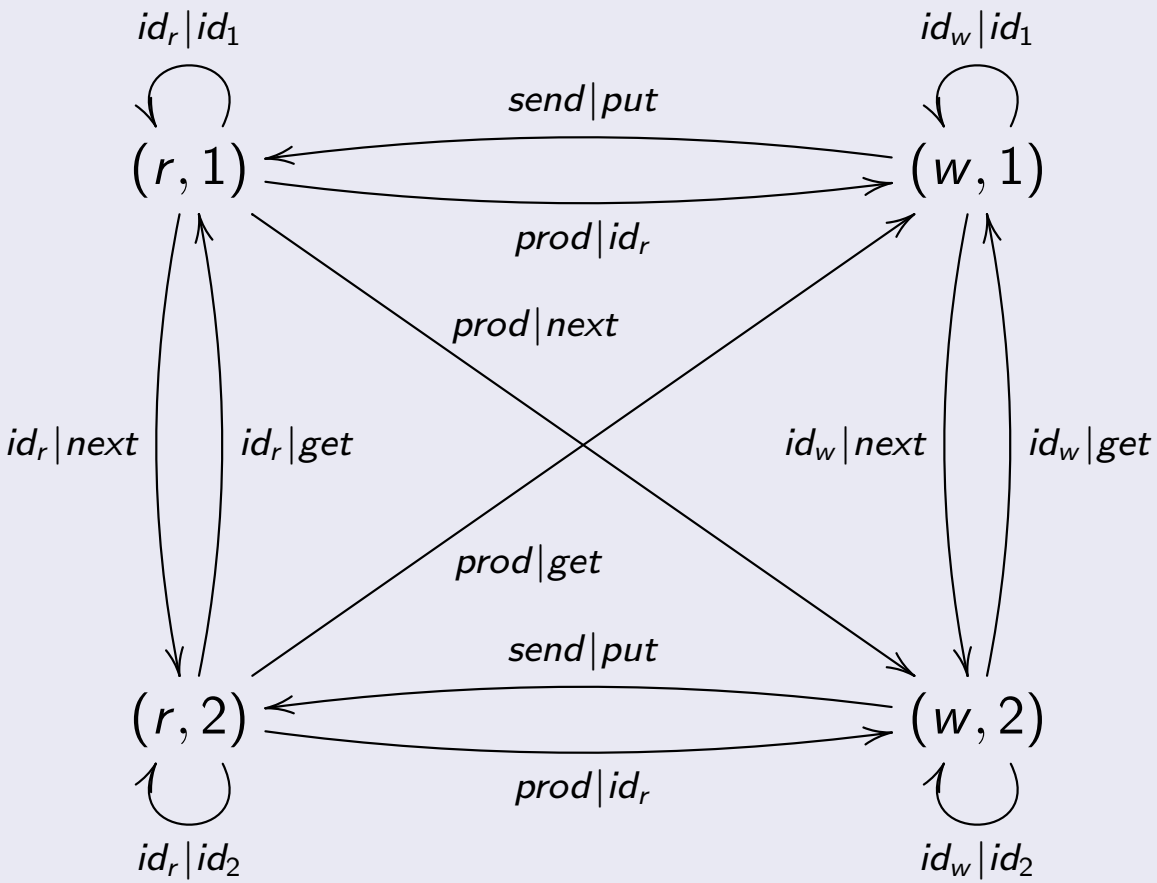
Behaviour = Synchronisation = Pullback

Sender action *send* and buffer action *put* must synchronize. This is a pullback construction and as the result we keep from the $24 = 4 \times 6$ parallel actions in $A_S \times A_P$ only $14 = 1 \times 2 + 3 \times 4$ actions.



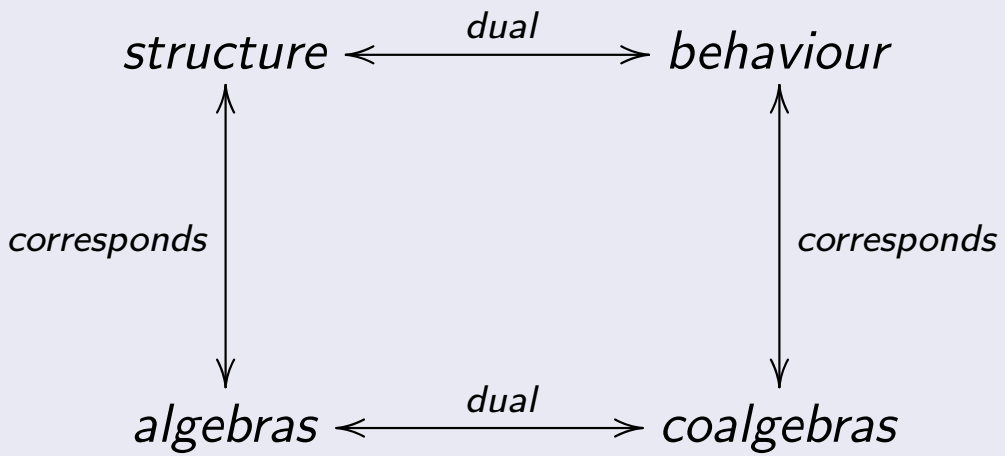
Example – Connecting Sender and Buffer by a Cable

Synchronized Automaton (Pullback-Automaton)



An idealized working hypothesis

Two dualities



Algebras and the Duality Limit-Colimit

Definitions

- Algebraic Signature $\Sigma = (S, OP, sc : OP \to S^*, tg : OP \to S)$. $op : w \to s$ stands for $sc(op) = w = s_1 \ldots s_n$ and $tg(op) = s$.
- Σ-algebra $\mathcal{A} = (\mathcal{A}(S), \mathcal{A}(OP))$ with $\mathcal{A}(S) = (\mathcal{A}(s) \mid s \in S)$ and $\mathcal{A}(OP) = (\mathcal{A}(op) : \mathcal{A}(w) \to \mathcal{A}(s) \mid op \in OP)$ where $\mathcal{A}(w) = \mathcal{A}(s_1) \times \ldots \times \mathcal{A}(s_n)$.
- Σ-homomorphismus $f : \mathcal{A} \to \mathcal{B}$ given by a family of maps $f = (f(s) : \mathcal{A}(s) \to \mathcal{B}(s) \mid s \in S)$ such that for all $op \in OP$ the following diagram commutes where $f(w) = f(s_1) \times \ldots \times f(s_n)$,

$$\begin{array}{ccc} \mathcal{A}(w) & \xrightarrow{\mathcal{A}(op)} & \mathcal{A}(s) \\ {\scriptstyle f(w)}\downarrow & & \downarrow{\scriptstyle f(s)} \\ \mathcal{B}(w) & \xrightarrow{\mathcal{B}(op)} & \mathcal{B}(s) \end{array}$$

- $\mathrm{Alg}(\Sigma)$ is the category of all Σ-algebras.

Algebras and Limits

Target of all operations is a single set and not a product of sets!

Therefore all limit constructions are easy,i.e., inherited from the corresponding constructions for sets and maps:

- final Σ-algebra and all products of Σ-algebras
- Σ-subalgebras (and thus also Σ-congruence relations) are closed under intersection
- Σ-subalgebras are closed under pre-images w.r.t. Σ-homomorphisms

$$\begin{array}{ccc} f^{-1}(\mathcal{C}) \\ \downarrow f \\ \mathcal{C} \end{array} \qquad \begin{array}{ccc} f^{-1}(\mathcal{C}(S)) & \hookrightarrow & \mathcal{A}(S) \\ \downarrow f & & \downarrow f \\ \mathcal{C}(S) & \hookrightarrow & \mathcal{B}(S) \end{array} \qquad \begin{array}{ccc} \mathcal{B}(S) \\ \downarrow f \\ \mathcal{B} \end{array}$$

- ...

Algebras and Colimits

Sources of operations can be proper products!

Therefore are colimit constructions, in general, difficult and complicated, i.e., those constructions are based on something new that goes beyond the corresponding constructions for sets and maps:

- quotient algebras
- sums of Σ-algebras (if we have in Σ constants and/or n-ary operations with $n \geq 2$)
- initial und free Σ-algebras (Σ-term algebras and their quotients)

Initial und free Algebras

Variable system and Terms

For a **variable system** $X = (X(s) \mid s \in S)$ is the S-indexed family $T_\Sigma(X) = (T_\Sigma(X)(s) \mid s \in S)$ of **Σ-terms over X** inductively defined:

- $X(s) \subseteq T_\Sigma(X)(s)$ for all $s \in S$.
- $op\langle t_1, \ldots, t_n\rangle \in T_\Sigma(X)(s)$ for all $op : s_1 \ldots s_n \to s$ in OP and all n-tuples $(t_1, \ldots, t_n) \in T_\Sigma(X)(s_1) \times \ldots \times T_\Sigma(X)(s_n)$.

Term algebras

The construction step $(t_1, \ldots, t_n) \mapsto op\langle t_1, \ldots, t_n\rangle$ can be seen as the application of an operation in a "syntactic" Σ-algebra: For any **variable system** X we obtain a **Σ-term algebra over X**:

- $\mathcal{T}_\Sigma(X)(S) = T_\Sigma(X)(S)$ for all $s \in S$.
- $\mathcal{T}_\Sigma(X)(op)(t_1, \ldots, t_n) = op\langle t_1, \ldots, t_n\rangle \in T_\Sigma(X)(s)$ for all $op : s_1 \ldots s_n \to s$ in OP and all n-tuples $(t_1, \ldots, t_n) \in \mathcal{T}_\Sigma(X)(w) = \mathcal{T}_\Sigma(X)(s_1) \times \ldots \times \mathcal{T}_\Sigma(X)(s_n)$.

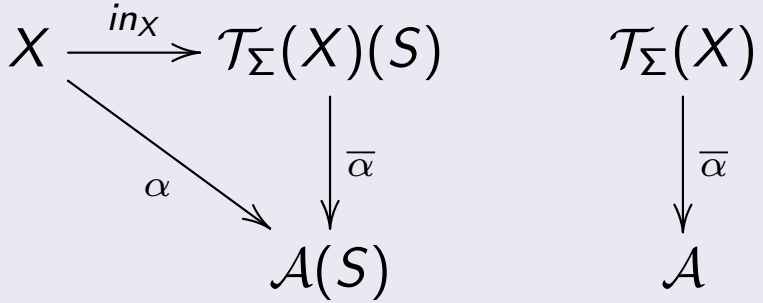
Initial and free Algebras

What properties are characterizing the elements in $\mathcal{T}_\Sigma(X)$?

Each $t \in \mathcal{T}_\Sigma(X)(s)$, $s \in S$ is **uniquely constructed** in **finite many steps** by means of the operations from $\mathcal{T}_\Sigma(X)(OP)$. Besides its role as a "data item" in the Σ-algebra $\mathcal{T}_\Sigma(X)$ a term t represents, at the same time, its own unique construction recipe/history.

What properties characterize the term algebra $\mathcal{T}_\Sigma(X)$ as a whole?

The property "uniquely finitely constructible" is reflected by the following **universal property** of $\mathcal{T}_\Sigma(X)$: For any Σ-algebra $\mathcal{A}$ and any variable assignment $\alpha = (\alpha(s) : X(s) \to \mathcal{A}(s) \mid s \in S)$ there exists a unique Σ-homomorphism $\overline{\alpha} : \mathcal{T}_\Sigma(X) \to \mathcal{A}$ with $in_X ; \overline{\alpha} = \alpha$.



Isomorphic free algebras

Fact

The universal property determines the **free Σ-algebra** $\mathcal{T}_\Sigma(X)$ uniquely up to isomorphism.

Practical use

The definition of (polymorphic) data types in functional programming is based on this fact!

Example - Lists

We consider Σ with $S = \{Elem, List\}$ and two (constructor) operations $nil : \rightarrow List$, $add : Elem\ List \rightarrow List$.
For each set (od data) D the term algebra $\mathcal{T}_\Sigma(D)$ is isomorphic to the Σ-algebra $\mathcal{L}$ with $\mathcal{L}(Elem) = D$, $\mathcal{L}(List) = D^*$, $\mathcal{L}(nil)() = \ulcorner\urcorner$ and $\mathcal{L}(add)(d, \ulcorner d_1, \ldots, d_n \urcorner) = d :: \ulcorner d_1, \ldots, d_n \urcorner = \ulcorner d, d_1, \ldots, d_n \urcorner$.

Selectors

Selectors as the inverse of constructors

That each element e in a free Σ-algebra has a unique construction means that e has unique direct "subelements", that we can access.
In case of term algebras we can access subelements by means of pattern matching.
To describe the access to subelements for arbitrary free algebras we can introduce for each $op : s_1 \dots s_n \to s$ in OP with $n \geq 1$, n partial **selectors** $sel_i^{op} : s \to s_i$ where we require $sel_i^{op}\langle op\langle x_1, \dots, x_n\rangle\rangle = x_i$ for all $1 \leq i \leq n$ and $op\langle sel_1^{op}\langle y\rangle, \dots, sel_n^{op}\langle y\rangle\rangle = y$.

Selectors for Lists

In case of lists we have $\mathcal{L}(add) : D \times D^* \to D^*$, and usually the partial operations $\mathcal{L}(sel_1^{add}) : D^* \to D$ and $\mathcal{L}(sel_2^{add}) : D^* \to D^*$ are denoted by *head* and *tail*, respectively.

Selectors as Coalgebra

Selectors more abstractly considered

The relation between constructors and selectors in free algebras can be described more summary and abstractly: In case of lists we can collect the two constructor operations $\ulcorner\urcorner$ and :: into one single map by means of case distinction where $1 = \{()\}$,

$$1 + D \times D^* \xrightarrow{[\ulcorner\urcorner,::]} D^*$$

That each element in a free Σ-algebra has unique direct "subelements", means then nothing but that the map $[\ulcorner\urcorner, ::]$ is bijective. The corresponding inverse map can be denoted (a bit incorrect) by

$$D^* \xrightarrow{\langle head, tail \rangle} 1 + D \times D^*$$

And this inverse map is a coalgebra!!!

(Unsorted) Coalgebras

'Signatures' for Coalgebras

We consider and expression $F(Z)$ that contains beside the set identifier Z the following set operations:

- the singleton set 1,
- optional parameter sets D,
- sum (disjoint union) $\_ + \_$,
- product $\_ \times \_$, and
- function space $[\_ \to \_]$.

Example: List selectors

Here we have the 'signature' $L(Z) = 1 + D \times Z$ with parameter D.

F-coalgebras

An F-coalgebra (C, γ) is given by a set C and a map $\gamma : C \to F(C)$.

Coalgebra - Example List selectors

Question

What kind of structures/systems are described by $(1 + D \times \_)$-coalgebras?

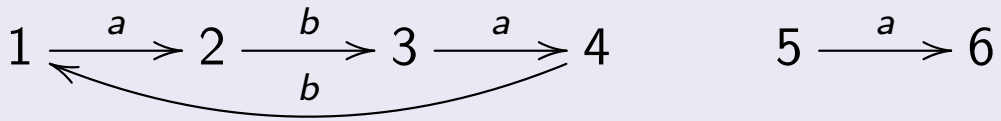
$$A \xrightarrow{\quad\gamma\quad} 1 + D \times A$$

Answer

Partial deterministic automata with output!

Example of a $(1 + D \times \_)$-coalgebra

Parameter $D = \{a, b\}$, set of states $A = \{1, 2, 3, 4, 5, 6\}$ and state transitions $\gamma(1) = (a, 2)$, $\gamma(2) = (b, 3)$, $\gamma(3) = (a, 4)$, $\gamma(4) = (b, 1)$, $\gamma(5) = (a, 6)$, $\gamma(6) = ()$,

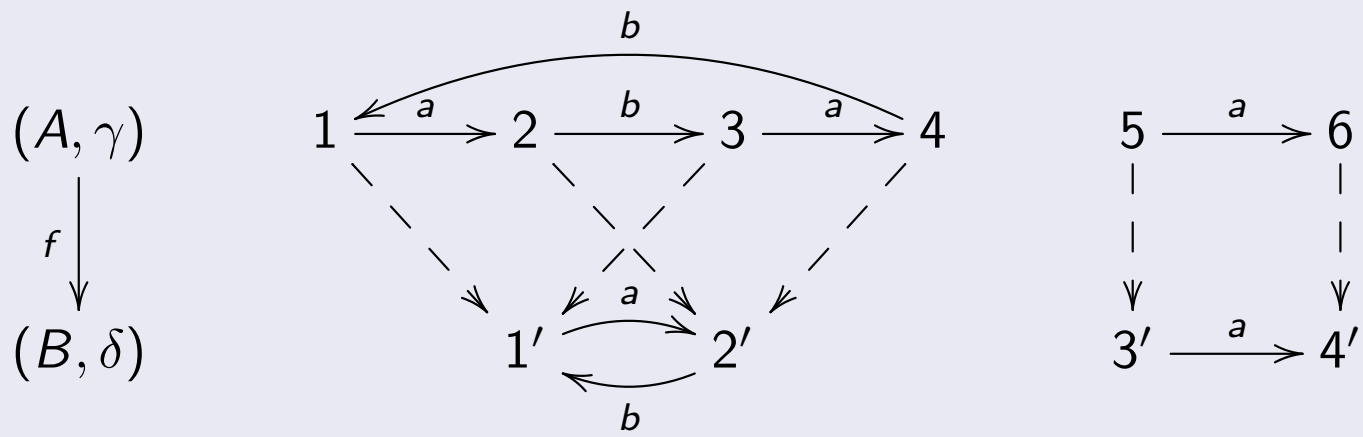


Morphisms between coalgebras

A morphismus $f : (A, \gamma) \to (B, \delta)$ between two F-coalgebras is a map $f : A \to B$ such that $\gamma; F(f) = f; \delta$

$$\begin{array}{ccc} A & \xrightarrow{\gamma} & F(A) \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle F(f)} \\ B & \xrightarrow{\delta} & F(B) \end{array} \qquad \begin{array}{ccc} A & \xrightarrow{\gamma} & 1 + D \times A \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle id_1 + id_D \times f} \\ B & \xrightarrow{\delta} & 1 + D \times B \end{array}$$

Example: Morphism between L-coalgebras



Coalgebras and the duality Limit-Colimit

Colimits

Dual to algebras all colimit constructions are simple, i.e., based on the corresponding constructions for sets nad maps.

- initial F-coalgebra and all sums of F-coalgebras
- F-subcoalgebras closed under union
- quotients of F-coalgebras

Limits

Limit constructions are, in general, difficult or may not exist:

- F-subcoalgebras
- products of F-coalgebras
- final F-coalgebras

Final $(1 + D \times \_)$-coalgebra

Question

Does there exit a final $(1 + D \times \_)$-coalgebra, and, if yes, how does it look like?

Answer

A final $(1 + D \times \_)$-coalgebra exists and the carrier is the set $D^\infty = D^* + D^{\mathbb{N}}$ of all finite and infinite sequences of elements from D.

$$D^\infty \xrightarrow{\langle head, tail \rangle} 1 + D \times D^\infty$$

Final coalgebras are also algebras

Analogously to initial algebras, the structure map of a final F-coalgebra is bijective, and the corresponding inverse map is then an F-algebra. Roughly said, "process algebras" try to describe final coalgebras by means of this algebraic structure.

Final Morphisms

Question

What does the unique morphism from a coalgebra (A, γ) into the final coalgebra?

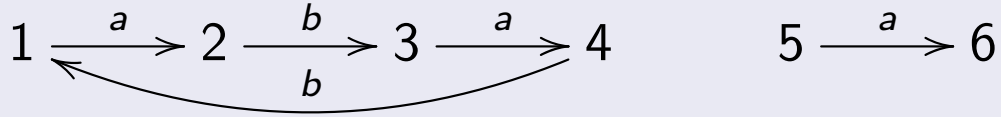
Answer

It assigns to each system state $a \in A$ a **process**, i.e., on often infinite mathematical object that describes the behaviour that one can observe if the system starts to run in this state a.

Final Morphisms

Example: Final morphism for a $(1 + D \times \_)$-coalgebra

For our example $(1 + D \times \_)$-coalgebra



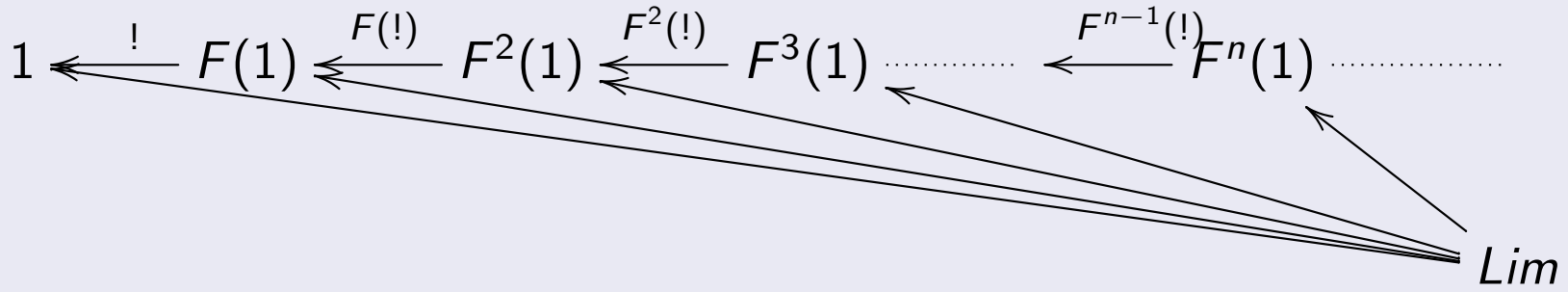
the final morphism $! : (A, \gamma) \to (D^\infty, \langle head, tail \rangle)$ produces the following processes

$$
\begin{aligned}
!(1) &= ababbabab \ldots \cong (\varepsilon, a, ab, aba, abab, ababa, \ldots) \\
!(2) &= bababbaba \ldots \\
!(3) &= ababbabab \ldots \\
!(4) &= bababbaba \ldots \\
!(5) &= a \ldots \cong (\varepsilon, a, a, a, a, a, \ldots) \\
!(6) &= \varepsilon \ldots \cong (\varepsilon, \varepsilon, \varepsilon, \varepsilon, \varepsilon, \varepsilon, \ldots)
\end{aligned}
$$

Categorical construction of final F-coalgebren

Construction of final F-coalgebras

We start with 1 and obtain by iterative application of the construction F an increasing chain of sets where $F^n(1)$ represents the set of all observations we can potentially make if a system of kind F has made exactly n transition steps.



Then we take the limit *Lim* of this chain. The elements of *Lim* are infinite sequences of finite observations, i.e., finite approximations of infinite behaviour. Note, that we don't need "internal" orders and their completions!!!

Other examples of coalgebras

For what expression F the final F-coalgebra contains only the infinite sequences $D^{\mathbb{N}}$? **Antwort:** $F(Z) = D \times Z$

Partial Deterministice Automata with Input

Algebraically, these are systems of the kind

$$A \times D \xrightarrow{next} 1 + A$$

Question: Can these systems be considered as coalgebras?
Answer: Yes!! Those kind of systems can be described equivalently described by

$$A \xrightarrow{\lambda next} [D \to 1 + A]$$

The elements of the corresponding final coalgebras are exactly the "deterministic CSP processes", i.e., all prefix closed subsets of D^* (finite and infinite trees).

Duality Term-Process

Structure

Terms, i.e., the elements of initial algebra, represent finitely generated/constructed data.
Initial algebras are uniquely characterized by universal properties of constructors.

Behaviour

Processes, i.e., the elements of final coalgebras, represent the possibly infinite behaviour of systems.
Final coalgebras are uniquely characterized by universal properties of selektors/observations.

Thanks !!!