From grammars and automata to algebras and coalgebras

Peter Padawitz

Technical University of Dortmund, Germany

April 2, 2013

Abstract. The increasing application of notions and results from category theory, especially from algebra and coalgebra, has revealed that any formal software or hardware model is constructor- or destructor-based, a *white-box* or a *black-box* model. A highly-structured system may involve both constructor- and destructor-based components. The two model classes and the respective ways of developing them and reasoning about them are dual to each other. Roughly said, algebras generalize the modeling with context-free grammars, word languages and structural induction, while coalgebras generalize the modeling with automata, Kripke structures, streams, process trees and all other state- or object-oriented formalisms. We summarize the basic concepts of co/algebra and illustrate them at a couple of signatures including those used in language or compiler construction like regular expressions or acceptors.

1 Introduction

More than forty years of research on formal system modeling led to the distinction between algebraic models on the one hand and coalgebraic ones on the other. The former describes a system in terms of the synthesis of its components by means of object-building operators (constructors). The latter models a system in terms of the analysis of its components by means of object-modifying, -decomposing or -measuring operators (destructors). The traditional presentation of a class of *algebraic* models is a context-free grammar that provides a concrete syntax of a set of constructors, whereas a class of *coalgebraic* models is traditionally given by all automata, transition systems or Kripke structures with the same *behavior*. Their respective state transition or labeling functions yield a set of the destructors. But not *any* member of such a class of models admits the application of powerful methods to operate on and reason about it. Among the members of an algebraic class it is the *initial* one, among those of a coalgebraic class it is the *final* one that the modeling should aim at. Initial algebras enable recursion and induction. Final coalgebras enable corecursion and coinduction.

Twenty years ago algebraic modeling was mainly algebraic specification and thus initial and free algebras were the main objects of interest [?,18, 15, 10], although *hidden algebra* and *final semantics* approaches [22, 31, 17, 28, 48, 49] already tended to the coalgebraic view (mostly in terms of greatest quotients of initial models). But first the dual concepts of category and fixpoint theory paved the way to the principles and methods current algebraic system modeling is based upon.

Here we use a slim syntax of types and many-sorted signatures, expressive enough for describing most models one meets in practice, but avoiding new guises for well-established categorical concepts. For instance and in contrast to previous *hierarchical* approaches (including our own), we keep off the explicit distinction of primitive or base sorts and a fixed base algebra because such entities are already captured by the constant functors among the types of a signature. Section 2 presents the syntax and semantics of the domains for co/algebraic models of constructive resp. destructive signatures. Section 3 draws the connection from signatures to functor co/algebras and provides initial and final model constructions. Roughly said, the latter are least resp. greatest fixpoints of the respective functor. Section 4 presents fundamental concepts and rules dealing with the extension, abstraction or restriction of and the logical reasoning about co/algebraic models. Again we are faced with least and greatest fixpoints, here with the relational ones co/inductive proofs are based upon. Moreover, each element of a – usually coalgebraic – class of infinite structures can be modeled as the *unique* fixpoint of the function derived from a set of guarded equations.

We assume that the reader is somewhat familiar with the notions of a category, a functor, a diagram, a co/cone, a co/limit, a natural transformation and an adjunction. Given a category $\mathcal{K}$, the target object of a colimit resp. source object of a limit of the empty diagram $\emptyset \to \mathcal{K}$ is called **initial** resp. **final in** $\mathcal{K}$. We remind of the uniqueness of co/limits modulo or up to isomorphism. Hence initial or final objects are also unique up to isomorphism.

Set denotes the category of sets with functions as morphisms. Given an index set I, $\prod_{s\in I} A_i$ and $\coprod_{s\in I} A_i$ denote the product resp. coproduct (= disjoint union) of sets A_i, $i \in I$. For all $n > 1$, $A_1 \times \ldots \times A_n = \prod_{s\in\{1,\ldots,n\}} A_i$ and $A_1 + \ldots + A_n = \coprod_{s\in\{1,\ldots,n\}} A_i$. For all $i \in I$, $\pi_i : \prod_{s\in I} A_i \to A_i$ and $\iota_i : A_i \to \coprod_{s\in I} A_i$ denote the i-th projection resp. injection: For all $a = (a_i)_{i\in I} \in \prod_{s\in I} A_i$, $i \in I$ and $b \in A_i$, $\pi_i(a) = a_i$ and $\iota_i(b) = (b, i)$. Given functions $f_i : A \to A_i$ and $g_i : A_i \to A$ for all $i \in I$, $\langle f_i\rangle_{i\in I} : A \to \prod_{s\in I} A_i$ and $[g_i]_{i\in I} : \coprod_{s\in I} A_i \to A$ denote the product resp. coproduct extension of $\{f_i\}_{i\in I}$: For all $a \in A$, $i \in I$, $b \in A_i$ and $n > 1$, $\langle f_i\rangle_{i\in I}(a) = (f_i(a))_{i\in I}$, $[g_i](b, i) = g_i(b)$, $\prod_{s\in I} f_i = \langle f_i \circ \pi_i\rangle$, $f_1 \times \ldots \times f_n = \prod_{s\in\{1,\ldots,n\}} f_i$, $\coprod_{s\in I} g_i = [\iota_i \circ g_i]$ and $g_1 + \ldots + g_n = \coprod_{s\in\{1,\ldots,n\}} g_i$.

1 denotes the singleton $\{*\}$. 2 denotes the two-element set $\{0, 1\}$. The elements of 2 are regarded as truth values. Let A be a set. $id_A : A \to A$ denotes the identity on A. $A^* = \{a \in A^n \mid n \in \mathbb{N}\}$, $\mathcal{P}_{fin}(A) = \{f : A \to 2 \mid |f^{-1}(1)| < \omega\}$ and $\mathcal{B}_{fin}(A) = \{f : A \to \mathbb{N} \mid |f^{-1}(\mathbb{N} \setminus \{0\})| < \omega\}$ denote the sets of finite words, sets resp. multisets of elements of A.

2 Many-sorted signatures and their algebras

Let S be a set of **sorts**. An S**-sorted** or S**-indexed set** is a family $A = \{A_s \mid s \in S\}$ of sets. An S**-sorted subset** of A, written as $B \subseteq A$, is an S-sorted set with $A \subseteq B$ for all $s \in S$. Given S-sorted sets $A_1, \ldots, A_n$, an S**-sorted relation**

$r \subseteq A_1 \times \ldots \times A_n$ is an S-sorted set with $r_s \subseteq A_{1,s} \times \ldots \times A_{n,s}$ for all $s \in S$. Given S-sorted sets A, B, an **S-sorted function** $f : A \to B$ is an S-sorted set such that for all $s \in S$, f_s is a function from A_s to B_s. Set^S denotes the category of S-sorted sets as objects and S-sorted functions as morphisms.

$\mathbb{T}(S)$ denotes the inductively defined set of (bounded) **types over** S:

$$\begin{array}{lll} S \subseteq \mathbb{T}(S), & & \\ X \in Set & \Rightarrow & X \in \mathbb{T}(S), \\ e_1, \ldots, e_n \in \mathbb{T}(S) & \Rightarrow & e_1 \times \ldots \times e_n, e_1 + \ldots + e_n \in \mathbb{T}(S), \\ e \in \mathbb{T}(S) & \Rightarrow & word(e), bag(e), set(e) \in \mathbb{T}(S), \\ X \in Set \wedge e \in S & \Rightarrow & e^X \in \mathbb{T}(S). \end{array}$$

We regard $e \in \mathbb{T}(S)$ as a finite tree: Each inner node of e is labelled with a *type constructor* ($\times$, $+$, *list*, *bag*, *set* or $\_^X$ for some $X \in Set$) and each leaf is labelled with an element of S. A set is a **base set** of e if it occurs in e.

$e \in \mathbb{T}(S)$ is **polynomial** if e does not contain *set*. $\mathbb{PT}(S)$ denotes the set of polynomial types over S.

The meaning of $e \in \mathbb{T}(S)$ is a functor $F_e : Set^S \to Set$ that is inductively defined as follows (also called **predicate lifting**; see [25, 26]): Let A, B be S-sorted sets, $h : A \to B$ be an S-sorted function, $s \in S$, $X \in Set$, $e, e_1, \ldots, e_n \in \mathbb{T}(S)$, $a_1, \ldots, a_n \in F_e(A)$, $f \in \mathcal{B}_{fin}(F_e(A))$, $g \in \mathcal{P}_{fin}(F_e(A))$, $b \in F_e(B)$ and $g' : X \to F_e(A)$.

$$\begin{array}{l} F_s(A) = A_s,\ F_s(h) = h_s,\ F_X(A) = X,\ F_X(h) = id_X, \\ F_{e_1 \times \ldots \times e_n}(A) = F_{e_1}(A) \times \ldots \times F_{e_n}(A),\ \ F_{e_1 \times \ldots \times e_n}(h) = F_{e_1}(h) \times \ldots \times F_{e_n}(h), \\ F_{e_1 + \ldots + e_n}(A) = F_{e_1}(A) + \ldots + F_{e_n}(A),\ \ F_{e_1 + \ldots + e_n}(h) = F_{e_1}(h) + \ldots + F_{e_n}(h), \\ F_{word(e)}(A) = F_e(A)^*,\ \ F_{word(e)}(h)(a_1, \ldots, a_n) = (F_e(h)(a_1), \ldots, F_e(h)(a_n)), \\ F_{bag(e)}(A) = \mathcal{B}_{fin}(F_e(A)), \\ F_{bag(e)}(h)(f)(b) = \sum\{f(a) \mid a \in F_e(A),\ F_e(h)(a) = b\}, \\ F_{set(e)}(A) = \mathcal{P}_{fin}(F_e(A)), \\ F_{set(e)}(h)(g)(b) = \bigvee\{g(a) \mid a \in F_e(A),\ F_e(h)(a) = b\}, \\ F_{e^X}(A) = F_e(A)^X,\ \ F_{e^X}(h)(g') = F_e(h) \circ g'. \end{array}$$

We often write A_e for the set $F_e(A)$ and h_e for the function $F_e(h)$. Each function $E : S \to \mathbb{T}(S)$ induces an endofunctor $F_E : Set^S \to Set^S$: For all $s \in S$, $F_E(A)(s) = F_{E(s)}(A)$ and $F_E(h)(s) = F_{E(s)}(h)$.

Given S-sorted sets A, B and an S-sorted relation $r \in A \times B$, the **relation lifting** $Rel_e(r) \subseteq A_e \times B_e$ of r is inductively defined as follows (analogously to

[25, 26]): Let $s \in S$, $e, e_1, \dots, e_n \in \mathbb{T}(S)$ and $X \in Set$.

$$\begin{aligned}
&Rel_s(r) = r_s,\ Rel_X(r) = \langle id_X, id_X\rangle(X),\\
&Rel_{e_1\times\dots\times e_n}(r) = \{((a_1,\dots,a_n),(b_1,\dots,b_n)) \mid \forall\, 1 \le i \le n : (a_i,b_i) \in Rel_{e_i}(r)\},\\
&Rel_{e_1+\dots+e_n}(r) = \{((a,i),(b,i)) \mid (a,b) \in Rel_{e_i}(r),\ 1 \le i \le n\},\\
&Rel_{word(e)}(r) = \{(a_1\dots a_n, b_1\dots b_n) \mid \forall\, 1 \le i \le n : (a_i,b_i) \in Rel_e(r),\ n \in \mathbb{N}\},\\
&Rel_{bag(e)}(r) = \{(f,g) \mid \exists\, p : supp(f) \stackrel{\sim}{\to} supp(g) :\\
&\qquad\qquad \forall\, a \in supp(f) : f(a) = g(p(a)) \wedge (a, p(a)) \in Rel_e(r)\},\\
&Rel_{set(e)}(r) = \{(C,D) \mid \forall\, c \in C\ \exists\, d \in D : (c,d) \in Rel_e(r)\ \wedge\\
&\qquad\qquad \forall\, d \in D\ \exists\, c \in C : (c,d) \in Rel_e(r),\\
&Rel_{e^X}(r) = \{(f,g) \mid \forall\, x \in X : \langle f,g\rangle(x) \in Rel_e(r)\}.
\end{aligned}$$

We often write r_e for the relation $Rel_e(r)$.

A **signature** $\Sigma = (S, F, P)$ consists of a finite set S (of sorts), a finite $\mathbb{T}(S)^2$-sorted set F of **function symbols** and a finite $\mathbb{T}(S)$-sorted set P of **predicate symbols**. $f \in F_{(e,e')}$ is written as $f : e \to e' \in F$. $dom(f) = e$ is the **domain** of f, $ran(f) = e'$ is the **range** of f. $p \in P_e$ is written as $p : e \in P$. $f : e \to e'$ is an e'**-constructor** if $e' \in S$. f is an e**-destructor** if $e \in S$. Σ is **constructive** resp. **destructive** if F consists of constructors resp. destructors. Σ is **polynomial** if for all $f : e \to e' \in F$, e' is polynomial. A set is a **base set** of Σ if it occurs in the domain or range of a function or predicate symbol.

Example 2.1 Here are some constructive signatures without predicate symbols. Let X and Y be sets.

1. *Nat* (natural numbers) $S = \{nat\}$, $F = \{0 : 1 \to nat,\ succ : nat \to nat\}$.
2. *Reg*(X) (regular expressions over X) $S = \{reg\}$,
$$\begin{aligned}F = \{\ &\emptyset,\ \epsilon : 1 \to reg,\ \_ : X \to reg,\\ &\_|\_,\ \_\cdot\_ : reg \times reg \to reg,\ star : reg \to reg\ \}.\end{aligned}$$
3. *List*(X) (finite sequences of elements of X) $S = \{list\}$,
$F = \{nil : 1 \to list,\ cons : X \times list \to list\}$.
4. *Tree*(X, Y) (finitely branching trees of finite depth with node labels from X and edge labels from Y) $S = \{tree, trees\}$,
$$\begin{aligned}F = \{\ &join : X \times trees \to tree,\ nil : 1 \to trees,\\ &cons : Y \times tree \times trees \to trees\ \}.\end{aligned}$$
5. *BagTree*(X, Y) (finitely branching unordered trees of finite depth with node labels from X and edge labels from Y)
$S = \{tree\}$, $F = \{join : X \times bag(Y \times tree) \to tree\}$.
6. *FDTree*(X, Y) (finitely or infinitely branching trees of finite depth with node labels from X and edge labels from Y)
$S = \{tree\}$, $F = \{join : X \times ((Y \times tree)^{\mathbb{N}} + word(Y \times tree)) \to tree\}$. ❏

Example 2.2 Here are some destructive signatures without predicate symbols. Let X and Y be sets.

1. *coNat* (natural numbers with infinity) $S = \{nat\}$,
$F = \{pred : nat \to 1 + nat\}$.

2. $coList(X)$ (finite or infinite sequences of elements of X; $coList(1) \simeq coNat$) $S = \{list\}$, $F = \{split : list \to 1 + (X \times list)\}$.
3. $DetAut(X, Y)$ (deterministic Moore automata with input set X and output set Y) $S = \{state\}$, $F = \{\delta : state \to state^X,\ \beta : state \to Y\}$.
4. $NDAut(X, Y)$ (non-deterministic Moore automata; image finite labelled transition systems) $S = \{state\}$, $F = \{\delta : state \to set(state)^X,\ \beta : state \to Y\}$.
5. $coTree(X, Y)$ (finitely or infinitely branching trees of finite or infinite depth with node labels from X and edge labels from Y) $S = \{tree, trees\}$,

$$F = \{\ root : tree \to X,\ subtrees : tree \to trees,\\ split : trees \to 1 + (Y \times tree \times trees)\ \}.$$

6. $FBTree(X, Y)$ (finitely branching trees of finite or infinite depth with node labels from X and edge labels from Y) $S = \{tree\}$,
$F = \{root : tree \to X,\ subtrees : tree \to word(Y \times tree)\}$. ❑

Let $\Sigma = (S, F, P)$ be a signature. A **Σ-algebra** A consists of an S-sorted set, the **carrier** of A, also denoted by A, for each $f : e \to e' \in F$, a function $f^A : A_e \to A_{e'}$, and for each $p : e \in P$, a relation $p^A \subseteq A_e$.

Let A and B be Σ-algebras, $h : A \to B$ be an S-sorted function and $f : e \to e' \in F$. h is **compatible with** f if $h_{e'} \circ f^A = f^B \circ h_e$. h is a **Σ-homomorphism** if for all $f \in F$, h is compatible with f and for all $p : e \in P$, $h_e(p^A) \subseteq p^B$. h is **predicate preserving** if the converse holds true as well, i.e., for all $p : e \in P$, $p^B \subseteq h_e(p^A)$. A **Σ-isomorphism** is a bijective and predicate preserving Σ-homomorphism. Alg_Σ denotes the category of Σ-algebras and Σ-homomorphisms.

A signature $\Sigma' = (S', F', P')$ is a **subsignature** of Σ if $S' \subseteq S$, $F' \subseteq F$ and $P' \subseteq P$. Let A be a Σ-algebra and $h : A \to B$ be a Σ-homomorphism. The **Σ'-reducts** $A|_{\Sigma'}$ and $h|_\Sigma$ of A resp. h are the Σ'-algebra resp. Σ'-homomorphism defined as follows:

- For all $s \in S'$, $(A|_{\Sigma'})_s = A_s$ and $(h|_{\Sigma'})_s = h_s$.
- For all $f \in F' \cup P'$, $f^{A|_{\Sigma'}} = f^A$.

Σ'-reducts yield the **reduct functor** or **forgetful functor** $\_|_{\Sigma'}$ from Alg_Σ to $Alg_{\Sigma'}$.

A constructive signature $\Sigma = (S, F, P)$ **admits terms** if for all $f \in F$ there are $e_1, \dots, e_n \in S \cup Set$ with $dom(f) = e_1 \times \dots \times e_n$. If Σ admits terms, then the Σ-algebra T_Σ of (ground) **Σ-terms** is defined inductively as follows:

- For all $s \in S$, $f : e \to s \in F$ and $t \in T_{\Sigma,e}$, $f^{T_\Sigma}(t) = ft \in T_{\Sigma,s}$.

If a Σ-term is regarded as a tree, each inner node is labelled with some $f \in F$, while each leaf is labelled with an element of a base set of Σ. The interpretation of P in T_Σ is not fixed. Any such interpretation would be an S-sorted set of term relations, usually called a *Herbrand structure*. Constructive signatures that admit terms can be presented as context-free grammars:

A **context-free grammar** $G = (S, Z, BS, R)$ consists of finite sets S of **sorts** (also called nonterminals), Z of **terminals**, BS of **base sets** and $R \subseteq$

$S \times (S \cup Z \cup BS)^*$ of **rules**. The constructive signature $\Sigma(G) = (S, F, \emptyset)$ with

$$F = \{f_r : e_1 \times \ldots \times e_n \rightarrow s \mid {r = (s, w_0e_1w_1 \ldots e_nw_n) \in R, \atop w_0, \ldots, w_n \in Z^*,\ e_1, \ldots, e_n \in S \cup BS}\}$$

is called the **abstract syntax** of G (see [?], Section 3.1; [46], Section 3). $\Sigma(G)$-terms are usually called **syntax trees** of G.

Example 2.3 The regular expressions over X form the *reg*-carrier of the $Reg(X)$-algebra $T_{Reg(X)}$ of $Reg(X)$-terms.

The usual interpretation of regular expressions over X as languages (= sets of words) over X yields the $Reg(X)$-algebra *Lang*: $Lang_{reg} = \mathcal{P}(X^*)$. For all $x \in X$ and $L, L' \in \mathcal{P}(X^*)$,

$$\begin{array}{l} \emptyset^{Lang} = \emptyset,\ \epsilon^{Lang} = \{\epsilon\},\ \_^{Lang}(x) = \{x\}, \\ L|^{Lang}L' = L \cup L',\ L \cdot^{Lang} L' = \{vw \mid v \in L,\ w \in L'\}, \\ star^{Lang}(L) = \{w_1 \ldots w_n \mid n \in \mathbb{N},\ \forall\, 1 \leq i \leq n : w_i \in L\}. \end{array}$$

The $Reg(X)$-Algebra *Bool* interprets the regular operators as Boolean functions: $Bool_{reg} = 2$. For all $x \in X$ and $b, b' \in 2$,

$$\begin{array}{l} \emptyset^{Bool} = 0,\ \epsilon^{Bool} = 1,\ \_^{Bool}(x) = 0, \\ b|^{Bool}b' = b \vee b',\ b \cdot^{Bool} b' = b \wedge b',\ star^{Bool}(b) = 1. \quad ❑ \end{array}$$

Let $\Sigma = (S, F, P)$ be a signature and A be a Σ-algebra. An S-sorted subset *inv* of A is a **Σ-invariant** or **-subalgebra** of A if *inv* is **compatible** with all $f : e \rightarrow e' \in F$, i.e. $f^A(inv_e) \subseteq inv_{e'}$. $inc : inv \rightarrow A$ denotes the injective S-sorted **inclusion function** that maps a to a. *inv* can be extended to a Σ-algebra: For all $f : e \rightarrow e' \in F$, $f^{inv} = f^A \circ inc_e$, and for all $r : e \in R$, $r^{inv} = r^A \cap inv_e$. Given an S-sorted subset B of A, the least Σ-invariant including B is denoted by $\langle B \rangle$.

An S-sorted relation $\sim \subseteq A^2$ is a **Σ-congruence** if $\sim$ is **compatible** with all $f : e \rightarrow e' \in F$, i.e. $(f^A \times f^A)(\sim_e) \subseteq \sim_{e'}$. $\sim^{eq}$ denotes the equivalence closure of $\sim$. $A_\sim$ denotes the Σ-algebra that agrees with A except for the interpretation of all $r : e \in R$: $r^{A_\sim} = \{b \in A_e \mid \exists\, a \in r^A : a \sim^{eq} b\}$. $A/\sim$ denotes the S-sorted quotient set $\{[a]_\sim \mid a \in A\}$ where $[a]_\sim = \{b \in A \mid a \sim^{eq} b\}$. $nat_\sim : A \rightarrow A/\sim$ denotes the surjective S-sorted **natural function** that maps $a \in A$ to $[a]_\sim$. $A/\sim$ can be extended to a Σ-algebra: For all $f : e \rightarrow e' \in F$, $f^{A/\sim} \circ nat_{\sim,e} = nat_{\sim,e'} \circ f^A$. For all $r : e \in R$, $r^{A/\sim} = \{nat_{\sim,e}(a) \mid a \in r^{A_\sim}\}$.

Let $h : A \rightarrow B$ be an S-sorted function. The S-sorted subset $img(h) = \{h(a) \mid a \in A\}$ of B is called the **image** of h. The S-sorted relation $ker(h) = \{(a, b \in A^2) \mid h(a) = h(b)\}$ is called the **kernel** of h.

Proposition 2.4 (1) *inc* and $nat_\sim$ are Σ-homomorphisms. $h : A \rightarrow B$ is surjective iff $img(h) = B$. h is injective iff $ker(h) = \langle id_A, id_A \rangle(A)$.

(2) A is a Σ-algebra and h is a Σ-homomorphism iff $ker(h)$ is a Σ-congruence. B is a Σ-algebra and h is a Σ-homomorphism iff $img(h)$ is a Σ-invariant. ❑

Homomorphism Theorem 2.5 h is a Σ-homomorphism iff there is a unique surjective Σ-homomorphism $h' : A \rightarrow img(h)$ with $inc \circ h' = h$ iff there is a unique injective Σ-homomorphism $h' : A/ker(h) \to B$ with $h' \circ nat_{ker(h)} = h$. ❑

Example 2.6 Given a *behavior* function $f : X^* \to Y$, the *minimal realization* of f coincides with the invariant $\langle f \rangle$ of the following $DetAut(X,Y)$-algebra $MinAut$: $MinAut_{state} = (X^* \to Y)$ and for all $f : X^* \to Y$ and $x \in X$, $\delta^{MinAut}(f)(x) = \lambda w.f(xw)$ and $\beta^{MinAut}(f) = f(\epsilon)$.

Let $Y = 2$. Then behaviors $f : X^* \to Y$ coincide with languages over X, i.e. subsets L of X^*, and $\langle L \rangle$ is $DetAut(X,2)$-isomorphic to the minimal acceptor of L with $\{L \subseteq X^* \mid \epsilon \in L\}$ as the set of final states. Hence the *state*-carrier of $MinAut$ agrees with the *reg*-carrier of $Lang$ (see Ex. 2.3). $T_{Reg(X)}$ also provides acceptors of regular languages, i.e., $T = T_{Reg(X)}$ is a $DetAut(X,2)$-algebra. Its transition function $\delta^T : T \to T^X$ is called a *derivative* function. It has been shown that for all regular expressions R, $\langle R \rangle \subseteq T_{Reg(X)}$ has only finitely many states ([14], Thm. 4.3 (a); [42], Section 5; [27], Lemma 8). If combined with coinductive proofs of state equivalence (see Section 4), the stepwise construction of the least invariant $\langle R \rangle$ of $T_{Reg(X)}$ can be lifted to a stepwise construction of the least invariant $\langle L(R) \rangle$ of $MinAut = Lang$ (= minimal acceptor of $L(R)$), thus avoiding the traditional detour via powerset automata and their minimization (see [45], Section 4).

Let $X = 1$. Then $MinAut$ is $DetAut(1,Y)$-isomorphic to the algebra of **streams over** Y: $MinAut_{state} = Y^{1^*} \cong Y^{\mathbb{N}}$. For all $s \in Y^{\mathbb{N}}$, $\beta(s) = s(0)$ and $\delta(s)(*) = \lambda n.s(n+1)$.

Let $X = 2$. Then $MinAut$ represents the set of **infinite binary trees** with node labels from Y: $MinAut_{state} = X^{2^*}$. For all $t \in X^{2^*}$ and $b \in 2$, $\beta(t) = s(\epsilon)$, $\delta(t)(b) = \lambda w.t(bw)$. ❑

3 Σ-algebras and F-algebras

Let $\mathcal{K}$ be a category and F be an endofunctor on $\mathcal{K}$.

An F**-algebra** or F**-dynamics** is a $\mathcal{K}$-morphism $\alpha : F(A) \to A$. Alg_F denotes the category whose objects are the F-algebras and whose morphisms from $\alpha : F(A) \to A$ to $\beta : F(B) \to B$ are the $\mathcal{K}$-morphisms $h : A \to B$ with $h \circ \alpha = \beta \circ F(h)$. Hence α is initial in Alg_F if for all F-algebras β there is unique Alg_F-morphism h from α to β. h is **defined by recursion** and called a **catamorphism**.

An F**-coalgebra** or F**-codynamics** is a $\mathcal{K}$-morphism $\alpha : A \to F(A)$. $coAlg_F$ denotes the category whose objects are the F-coalgebras and whose morphisms from $\alpha : A \to F(A)$ to $\beta : B \to F(B)$ are the $\mathcal{K}$-morphisms $h : A \to B$ with $F(h) \circ \alpha = \beta \circ h$. Hence α is final in $coAlg_F$ if for all F-coalgebras β there is unique $coAlg_F$-morphism h from β to α. h is **defined by corecursion** and called an **anamorphism**.

Theorem 3.1 ([29], Lemma 2.2; [11], Prop. 5.12; [8], Section 2; [41], Thm. 9.1) Initial F-algebras and final F-coalgebras are isomorphisms in $\mathcal{K}$. ❑

In other words, the object A of an initial F-algebra $\alpha : F(A) \to A$ or a final F-coalgebra $\alpha : A \to F(A)$ is a *fixpoint* of F, i.e., A *solves* the equation $F(A) = A$.

Let $\Sigma = (S, F, P)$ be a signature. Σ induces an endofunctor H_Σ on Set^S (notation follows [2]): For all S-sorted sets and functions A and $s \in S$,

$$H_\Sigma(A)_s = \begin{cases} \coprod_{f:e\to s\in F} A_e & \text{if } \Sigma \text{ is constructive,} \\ \prod_{f:s\to e\in F} A_e & \text{if } \Sigma \text{ is destructive.} \end{cases}$$

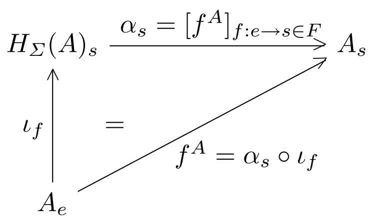
Example 3.2 (see Exs. 2.1 and 2.2) Let A be an S-sorted set.

$$\begin{aligned} H_{Nat}(A)_{nat} &= H_{coNat}(A)_{nat} = 1 + A_{nat}, \\ H_{List(X)}(A)_{list} &= H_{coList(X)}(A)_{list} = 1 + (X \times A_{list}), \\ H_{Reg(X)}(A)_{reg} &= 1 + 1 + X + A_{reg}^2 + A_{reg}^2 + A_{reg}, \end{aligned}$$

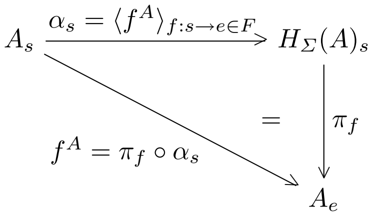
$$\begin{aligned} H_{DetAut(X,Y)}(A)_{state} &= A_{state}^X \times Y, \\ H_{NDAut(X,Y)}(A)_{state} &= \mathcal{P}_{fin}(A_{state})^X \times Y, \\ H_{Tree(X,Y)}(A)_{tree} &= H_{coTree(X,Y)}(A)_{tree} = X \times A_{trees}, \\ H_{Tree(X,Y)}(A)_{trees} &= H_{coTree(X,Y)}(A)_{trees} = 1 + (X \times A_{tree} \times A_{trees}), \\ H_{BagTree(X,Y)}(A)_{tree} &= Y \times \mathcal{B}_{fin}(X \times A_{tree}), \\ H_{FDTree(X,Y)}(A)_{tree} &= Y \times ((X \times A_{tree})^{\mathbb{N}} + (X \times A_{tree})^*), \\ H_{FBTree(X,Y)}(A)_{tree} &= Y \times (X \times A_{tree})^*. \end{aligned}$$

❏

Given a constructive signature Σ, an H_Σ-algebra $H_\Sigma(A) \xrightarrow{\alpha} A$ is an S-sorted function and uniquely corresponds to a Σ-algebra A: For all $s \in S$ and $f : e \to s \in F$,



Given a destructive signature Σ, an H_Σ-coalgebra $A \xrightarrow{\alpha} H_\Sigma(A)$ is an S-sorted function and uniquely corresponds to a Σ-algebra A: For all $s \in S$ and $f : s \to e \in F$,



α_s combines all s-constructors resp. -destructors into a single one.

An **ascending ω-chain** is a diagram sending the index category $\{n \to n+1 \mid n \in \mathbb{N}\}$ to $\mathcal{K}$. $\mathcal{K}$ is **ω-cocomplete** if the empty diagram and all ascending ω-chains have colimits. A **descending ω-chain** is a diagram sending the index category $\{n \leftarrow n+1 \mid n \in \mathbb{N}\}$ to $\mathcal{K}$. $\mathcal{K}$ is **ω-complete** if the empty diagram and all descending ω-chains have limits.

Let $\mathcal{K}$ and $\mathcal{L}$ be ω-cocomplete. A functor $F : \mathcal{K} \to \mathcal{L}$ is **ω-cocontinuous** if for all ascending ω-chains $\mathcal{D}$ and colimits $\{\mu_n : \mathcal{D}(n) \to C \mid n \in \mathbb{N}\}$ of $\mathcal{D}$, $\{F(\mu_n) \mid n \in \mathbb{N}\}$ is a colimit of $F \circ \mathcal{D}$.

Let $\mathcal{K}$ and $\mathcal{L}$ be ω-complete. A functor $F : \mathcal{K} \to \mathcal{L}$ is **ω-continuous** if for all descending ω-chains $\mathcal{D}$ and limits $\{\nu_n : C \to \mathcal{D}(n) \mid n \in \mathbb{N}\}$ of $\mathcal{D}$, $\{F(\nu_n) \mid n \in \mathbb{N}\}$ is a limit of $F \circ \mathcal{D}$.

Theorem 3.3 ([8], Section 2; [30], Thm. 2.1) (1) Let $\mathcal{K}$ be ω-cocomplete, $F : \mathcal{K} \to \mathcal{K}$ be an ω-cocontinuous functor, I be initial in $\mathcal{K}$, ini be the unique $\mathcal{K}$-morphism from I to $F(I)$ and A be the target of the colimit of the ascending ω-chain $\mathcal{D}$ defined as follows:

$$n \to n+1 \quad \mapsto \quad F^n(I) \stackrel{F^n(ini)}{\to} F^{n+1}(I).$$

Since F is ω-cocontinuous, $F(A)$ is the target of the colimit of $F \circ \mathcal{D}$. Hence there is a unique $\mathcal{K}$-morphism $ini'(F) : F(A) \to A$, which can be shown to be an initial F-algebra.

(2) Let $\mathcal{K}$ be ω-complete, $F : \mathcal{K} \to \mathcal{K}$ be an ω-continuous functor, T be final in $\mathcal{K}$, *fin* be the unique $\mathcal{K}$-morphism from $F(T)$ to T and A be the source of the limit of the descending ω-chain $\mathcal{D}$ defined as follows:

$$n \leftarrow n+1 \quad \mapsto \quad F^n(T) \stackrel{F^n(fin)}{\leftarrow} F^{n+1}(T).$$

Since F is ω-continuous, $F(A)$ is the source of the limit of $F \circ \mathcal{D}$. Hence there is a unique $\mathcal{K}$-morphism $fin' : A \to F(A)$, which can be shown to be a final F-coalgebra. ❏

Theorem 3.4 (folklore) Set^S is ω-complete and ω-cocomplete. ❏

For defining data types of trees with infinite outdegree we need the generalization of Thm. 3.3 (1) from ω to greater ordinals λ. $F : \mathcal{K} \to \mathcal{K}$ is called **λ-cocontinuous** if F preserves colimits of ascending λ-chains, i.e., diagrams sending the index category $\{n \to n+1 \mid n < \lambda\}$ of ordinals to $\mathcal{K}$.

Thms. 3.3 and 3.1 tell us that the ascending chain $\mathcal{D} = \{F^n(I)\}_{n<\omega}$ *converges in ω steps*, i.e. $F(colim(\mathcal{D})) \cong colim(\mathcal{D})$. For $\lambda > \omega$, we extend $\mathcal{D}$ to $\{F^n(I)\}_{n\leq\lambda}$ where for all limit ordinals $n \leq \lambda$, $F^n(I) =_{def} colim(\{F^i(I)\}_{i<n})$. $F^\lambda(I)$ is the initial F-algebra (This was originally shown by [1]; see also [2], Thm. 3.19, or [5], Cor. 4.1.5). By [5], Thm. 4.1.12, all signatures Σ with constructors of infinite arity less than λ, H_Σ is λ-cocontinuous.

Given index sets I and J, a functor $F : Set^I \to Set^J$ is **permutative** if for all $A \in Set^I$ and $j \in J$ there is $i \in I$ such that $F(A)_j = A_i$.

Theorem 3.5 For all polynomial types e over S, $F_e : Set \to Set$ is ω-continuous.

Let e be a type over S, κ be the cardinality of the greatest base set occurring in e as an exponent and λ be the first regular cardinal number $> \kappa$. F_E is λ-cocontinuous.

Proof. By [9], Thms. 1 and 4, or [12], Prop. 2.2 (1) and (2), permutative and constant functors are ω-continuous and -cocontinuous, ω-continuous or λ-cocontinuous functors are closed under coproducts, ω-continuous functors are closed under products (and thus under exponentiation; see [41], Thm. 10.1) and λ-cocontinuous functors are closed under finite products. By [12], Prop. 2.2 (3), ω-continuous or λ-cocontinuous functors are closed under quotients modulo finite equivalence relations. Since for all sets A, $A^* \cong \coprod_{n\in\mathbb{N}} A^n$ and $\mathcal{B}_{fin}(A) \cong \coprod_{n\in\mathbb{N}} A^n/{\sim_n}$ where $a \sim_n b$ iff a is a permutation of b, $\_^*$ and $\mathcal{B}_{fin}$ are ω-continuous and -cocontinuous. By [5], Ex. 2.2.13, $\mathcal{P}_{fin}$ is ω-cocontinuous. For a proof of the fact that $\mathcal{P}_{fin}$ is not ω-continuous, see [5], Ex. 2.3.11. Analogously to [5], Thm. 4.1.12, one may show that λ-cocontinuous functors are closed under exponentiation by exponents with a cardinality less than λ. Finally, ω-continuous or λ-cocontinuous functors are closed under sequential composition. Putting all this together, we conclude that for all $e \in \mathbb{PT}(S)$, $F_e : Set^S \to Set$ is ω-continuous, and for all $e \in \mathbb{T}(S)$, F_e is λ-cocontinuous. ❑

Define $E(\Sigma) : S \to \mathbb{T}(S)$ as follows: For all $s \in S$,

$$E(\Sigma)(s) = \begin{cases} dom(f_1) + \ldots + dom(f_n) \text{ if } \{f_1, \ldots, f_n\} = \{f \in F \mid ran(f) = s\} \\ \qquad\qquad\qquad\qquad\qquad \text{and } \Sigma \text{ is constructive,} \\ ran(f_1) \times \ldots \times ran(f_n) \;\; \text{if } \{f_1, \ldots, f_n\} = \{f \in F \mid dom(f) = s\} \\ \qquad\qquad\qquad\qquad\qquad \text{and } \Sigma \text{ is destructive.} \end{cases}$$

Obviously, the endofunctor $F_{E(\Sigma)}$ agrees with H_Σ. Hence by Thm. 3.5, if Σ is constructive, then there is an initial Σ-algebra, if Σ is destructive and polynomial, then there is a final Σ-algebra, and both algebras are represented by co/limits of ascending resp. descending ω-chains:

Theorem 3.6 Let Σ be a constructive signature. By Thm. 3.3 (1), the initial Σ-algebra A is a colimit of a chain of S-sorted sets. Hence the carriers of A look as follows: Let I be the S-sorted set with $I_s = \emptyset$ for all $s \in S$, ini be the unique S-sorted function from I to $H_\Sigma(I)$ and $\sim_s$ be the equivalence closure of $\{(a, H^n_\Sigma(ini)(a)) \mid a \in H^n_\Sigma(I)_s,\ n \in \mathbb{N}\}$. For all $s \in S$,

$$A_s = (\coprod_{n\in\mathbb{N}} H^n_\Sigma(I)_s)/{\sim_s} .$$

Let B be a Σ-algebra, β_0 be the unique S-sorted function from I to B and for all $n \in \mathbb{N}$ and $s \in S$, $\beta_{n+1,s} = [f^B \circ F_e(\beta_{n,s})]_{f:e\to s\in F} : H^{n+1}_\Sigma(I)_s \to B_s$. The unique Σ-homomorphism $fold^B : A \to B$ is the unique S-sorted function satisfying $fold^B \circ nat_\sim = [\beta_n]_{n\in\mathbb{N}}$. ❑

Theorem 3.7 If Σ admits terms, then T_Σ is an initial Σ-algebra and for all Σ-algebras A, $fold^A : T_\Sigma \to A$ agrees with **term evaluation in** A: For all $f : e \to s \in F$ and $t \in T_{\Sigma,e}$, $fold^A(ft) = f^A(fold^A_e(t))$. ❑

Let $G = (S, Z, BS, P)$ be a context-free grammar (see Section 2) and $Y = \cup_{X\in BS} X$. The following $\Sigma(G)$-algebra is called the **word algebra of** G: For all

$s \in S$, $Word(G)_s = Z^*$. For all $w_0, \ldots, w_n \in Z^*$, $e_1, \ldots, e_n \in S \cup BS$, $r = (s, w_0 s_1 w_1 \ldots s_n w_n) \in R$ and $v \in F_{e_1 \times \ldots \times e_n}(Word(G)) \subseteq (Z \cup Y)^n$, $f_r^{Word(G)}(v) = w_0 v_1 w_1 \ldots v_n w_n$. The **language** $L(G)$ **of** G is the S-sorted image of $T_{\Sigma(G)}$ under term evaluation in $Word(G)$: For all $s \in S$, $L(G)_s = \{fold^{Word(G)}(t) \mid t \in T_{\Sigma(G),s}\}$. $L(G)$ is also characterized as the least solution of the set $E(G)$ of equations between the left- and right-hand sides of the rules of G (with the non-terminals regarded as variables). If G is not left-recursive, then the solution is unique [40]. This provides a simple method of proving that a given language L agrees with $L(G)$: Just show that L solves $E(G)$.

Each parser for G can be presented as a function $parse_G : (Z \cup Y)^* \rightarrow M(T_{\Sigma(G)})$ where M is a *monadic functor* that embeds $T_{\Sigma(G)}$ into a larger set of possible results, including syntax errors and/or sets of syntax trees for realizing non-deterministic parsing [40]. The parser is correct if $parse_G \circ fold^{Word(G)} = \eta_{T_{\Sigma(G)}}$ (where η is the *unit* of M) and if all words of $(Z \cup Y)^* \setminus L(G)$ are mapped to error messages.

The most fascinating advantage of algebraic compiler construction is the fact that the same *generic* compiler can be used for translating $L(G)$ into an arbitrary target language formulated as a $\Sigma(G)$-algebra A. The respective instance $compile_G^A : (Z \cup Y)^* \rightarrow M(A)$ agrees with the composition $M(fold^A) \circ parse_G$. More efficiently than by first constructing a syntax tree and then evaluating it in A, $compile_G^A$ can be implemented as a slight modification of $parse_G$. Whenever the parser performs a reduction step w.r.t. a rule r of G by building the syntax tree $f_r(t_1, \ldots, t_n)$ from already constructed trees $t_1, \ldots, t_n$, the compiler derived from $parse_G$ applies the interpretation f_r in A to already computed elements $a_1, \ldots, a_n$ of A and thus returns the target object $f_r^A(a_1, \ldots, a_n)$ instead of the tree $f_r(t_1, \ldots, t_n)$. Syntax trees need not be constructed at all!

Expressing the target language of a compiler for G as a $\Sigma(G)$-algebra *Target* also provides a method for proving that the compiler is correct w.r.t. the semantics $Sem(G)$ and $Sem(Target)$ of G resp. *Target*. The correctness amounts to the commutativity of the following diagram:

$$\begin{array}{ccc} T_{\Sigma(G)} & \xrightarrow{fold^{Target}} & Target \\ {\scriptstyle fold^{Sem(G)}}\downarrow & (1) & \downarrow{\scriptstyle execute} \\ Sem(G) & \xrightarrow{encode} & Sem(Target) \end{array}$$

Of course, $Sem(G)$ has to be a $\Sigma(G)$-algebra. $Sem(Target)$, however, usually refers to a signature different from $\Sigma(G)$. But the interpretations of the constructors of $\Sigma(G)$ in *Target* can often be transferred easily to $Sem(Target)$ such that the interpreter *execute* becomes a $\Sigma(G)$-homomorphism: For all $p \in P$, $f_r^{Sem(Target)} \circ execute = execute \circ f_r^{Target}$ is a definition of $f_r^{Sem(Target)}$ iff the kernel of *execute* is compatible with f_r^{Target}. Finally, we need a homomorphism *encode* that mirrors the (term) compiler $fold^{Target}$ on the semantical level. Finally, if all four functions of (1) are $\Sigma(G)$-homomorphisms, then the initiality of $T_{\Sigma(G)}$ in $Alg_{\Sigma(G)}$ implies that the diagram commutes!

Algebraic approaches to formal languages and compiler design are not new. They have been applied sucessfully to various programming languages (see, e.g., [**?**,46, 33, 13, 47, 32, 37]). Hence it is quite surprising that they are more or less ignored in the currently hot area of document definition and query languages (XML and all that) – although structured data play a prominent rôle in such languages. Instead of associating these data with adequate co/algebraic types, XML theoreticians boil everything down to regular expressions, words and word recognizing automata.

Example 3.8 (cf. Exs. 2.1 and 2.4) $\mathbb{N}$ is an initial *Nat*-algebra: $0^{\mathbb{N}} = 0$ and for all $n \in \mathbb{N}$, $succ^{\mathbb{N}}(n) = n + 1$.

$T_{Reg(X)}$ is an initial $Reg(X)$-algebra. Hence $T_{Reg(X),reg}$ is the set of regular expressions over X. For all such expressions R, $fold^{Lang}(R)$ is the language of R and $fold^{Bool}(R)$ checks it for inclusion of the empty word.

For $\Sigma \in \{List(X), Tree(X,Y), BagTree(X,Y), FDTree(X,Y)\}$, the elements of the *list*- resp. *tree*-carrier of an initial Σ-algebra can be represented by the sequences resp. trees that Ex. 2.1 associates with Σ. ❏

We proceed with to the destructor analogue of Thm. 3.6:

Theorem 3.9 Let Σ be a polynomial destructive signature. By Thm. 3.3 (2), the final Σ-algebra A is a limit of a chain of S-sorted sets. Hence the carriers of A look as follows: Let T be the S-sorted set with $T_s = 1$ for all $s \in S$ and *fin* be the unique S-sorted function from $H_\Sigma(T)$ to T. For all $s \in S$,

$$A_s = \{a \in \prod_{n \in \mathbb{N}} H^n_\Sigma(T)_s \mid \forall\, n \in \mathbb{N} : a_n = H^n_\Sigma(fin)(a_{n+1})\}.$$

Let B be a Σ-algebra, β_0 be the unique S-sorted function from B to T and for all $n \in \mathbb{N}$ and $s \in S$, $\beta_{n+1,s} = \langle F_e(\beta_{n,s}) \circ f^A \rangle_{f:s \to e \in F} : A_s \to H^{n+1}_\Sigma(T)_s$. The unique Σ-homomorphism $unfold^B : B \to A$ is the unique S-sorted function satisfying $inc \circ unfold^B = \langle \beta_n \rangle_{n \in \mathbb{N}}$. ❏

Example 3.10 (cf. Exs. 2.2 and 2.6) $A = \mathbb{N} \cup \{\infty\}$ is a final *coNat*-algebra: For all $n \in A$,

$$pred^A(n) = \begin{cases} * & \text{if } n = 0, \\ n - 1 & if\ n > 0, \\ \infty & if\ n = \infty. \end{cases}$$

MinAut is a final $DetAut(X,Y)$-algebra, in particular, the $DetAut(1,Y)$-algebra of streams over Y is a final $DetAut(1,Y)$-algebra.

Since $T = T_{Reg(X)}$ and *Lang* are $DetAut(X,2)$-algebras, $fold^{Lang} : T \to Lang$ is a $DetAut(X,2)$-homomorphism (see [40], Section 12) and *Lang* is a final $DetAut(X,2)$-algebra, $fold^{Lang}$ coincides with $unfold^T$. This fact allows us to build a generic parser for all regular languages upon δ^T and β^T and to extend it to a generic parser for all context-free languages by simply incorporating the respective grammar rules (see [40], Sections 12 and 14).

For $\Sigma \in \{coList(X), coTree(X,Y), FBTree(X,Y)\}$, the elements of the *list*-resp. *tree*-carrier of a final Σ-algebra can be represented by the sequences resp. trees that Ex. 2.2 associates with Σ. ❏

The construction of *coNat*, *coList* and *coTree* from *Nat*, *List* resp. *Tree* is not accidental. Let $\Sigma = (S,F,P)$ be a constructive signature and A be a Σ-algebra. Σ induces a destructive signature $Co\Sigma$: Since for all $s \in S$, $H_\Sigma(A)_s = \coprod_{f:e\to s\in F} A_e$, and since by Thm. 3.1, the initial H_Σ-algebra $[f^A]_{f:e\to s\in F}$ is an isomorphism, ini^{-1} is both a H_Σ-algebra and a $H_{Co\Sigma}$-coalgebra where

$$Co\Sigma = (S, \{d_s : s \to \coprod_{f:e\to s\in F} e \mid s \in S\}, R).$$

The final $Co\Sigma$-algebra is a *completion* of the initial Σ-algebra (cf. [12], Thm. 3.2; [3], Prop. IV.2). Its carriers consist of finitely branching trees such that each node is labelled with a base set or a constructor of Σ (cf. [**?**], Section 4; [7], Section II.2):

Let BS be the set of base sets of Σ. The $(BS\cup S)$-sorted set CT_Σ of **Σ-trees** consists of all partial functions $t : \mathbb{N}^* \to F \cup (\cup BS)$ such that for all $B \in BS$, $CT_{\Sigma,B} = B$ and for all $s \in S$, $t \in CT_{\Sigma,s}$ iff for all $w \in \mathbb{N}^*$,

- $t(\epsilon) \in F \wedge ran(t(\epsilon)) = s$,
- if $dom(t(w)) = e_1 \times \ldots \times e_n \to s' \in F$, then for all $0 \le i < n$, $t(wi) \in e_{i+1}$ or $t(wi) \in F$ and $ran(t(wi)) = e_{i+1}$.

CT_Σ is both a Σ- and a $Co\Sigma$-algebra: For all $f : e \to s \in F$, $t \in CT_{\Sigma,e})$ and $w \in \mathbb{N}^*$,

$$f^{CT_\Sigma}(t)(w) = \begin{cases} f & \text{if } w = \epsilon, \\ \pi_i(t)(v) & \text{if } \exists\, i \in \mathbb{N}, v \in \mathbb{N}^* : w = iv. \end{cases}$$

For all $s \in S$ and $t \in CT_{\Sigma,s}$,

$$d_s^{CT_\Sigma}(t) = ((\lambda w.t(0w), \ldots, \lambda w.t((|dom(t(\epsilon))| - 1)w)), t(\epsilon)) \in \coprod_{f:e\to s\in F} CT_{\Sigma,e}.$$

Moreover, CT_Σ is an ω-complete partially ordered S-sorted set – provided that Σ is **pointed**, i.e., for all $s \in S$ there is $B \in BS$ such that Σ contains a function symbol $\bot_s : B \to s$. A Σ-algebra A is **ω-continuous** if its carriers are complete partial orders and if for all $f \in F$, f^A is ω-continuous. ωAlg_Σ denotes the subcategory of Alg_Σ that consists of all ω-continuous Σ-algebras as objects and all ω-continuous Σ-homomorphisms between them.

Theorem 3.11 CT_Σ is a final $Co\Sigma$-algebra. If Σ is pointed, then CT_Σ is initial in ωAlg_Σ.

Proof. Initiality follows from [**?**], Thm. 4.15, [12], Thm. 3.2, or [3], Prop. IV.2.

Let A be a $Co\Sigma$-algebra. An S-sorted function $h = unfold^A : A \to CT_\Sigma$ is defined as follows: For all $s \in S$, $a \in A_s$, $i \in \mathbb{N}$ and $w \in \mathbb{N}^*$, $d_s^A(a) = ((a_1,\ldots,a_n), f)$ implies

$$h(a)(\epsilon) = f,$$
$$h(a)(iw) = \begin{cases} h(a_i)(w) & \text{if } 0 \le i < |dom(f)|, \\ \text{undefined otherwise}, \end{cases}$$

in short: $h(a) = f(h(a_1), \dots, h(a_n))$. Let $s \in S$, $a \in A_s$ and $d_s^A(a) = ((a_1, \dots, a_n), f)$. Then

$$\begin{array}{l} d_s^{CT_\Sigma}(h(a)) = d_s^{CT_\Sigma}(f(h(a_1), \dots, h(a_n))) \\ = ((h(a_1), \dots, h(a_n)), f) = h((a_1, \dots, a_n), f) = h(d_s^A(a)). \end{array}$$

Hence h is a $co\Sigma$-homomorphism. Conversely, let $h' : A \to CT_\Sigma$ be a $co\Sigma$-homomorphism. Then

$$\begin{array}{l} d_s^{CT_\Sigma}(h'(a)) = h'(d_s^A(a)) = h'((a_1, \dots, a_n), f) = ((h'(a_1), \dots, h'(a_n)), f) \\ = d_s^{CT_\Sigma}(f(h'(a_1), \dots, h'(a_n))) \end{array}$$

and thus $h'(a) = f(h'(a_1), \dots, h'(a_n))$ because $d_s^{CT_\Sigma}$ is injective. We conclude that h' agrees with h. ❑

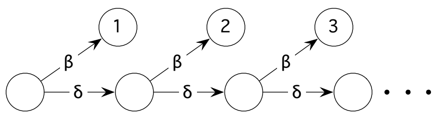
Another class of polynomial destructive signatures is obtained by dualizing constructive signatures that admit terms. A destructive signature $\Sigma = (S, F, P)$ **admits coterms** if for all $f \in F$ there are $e_1, \dots, e_n \in S \cup Set$ with $ran(f) = e_1 + \ldots + e_n$. If Σ admits terms, then the Σ-algebra coT_Σ of **Σ-coterms** is defined as follows:

- For all $s \in S$, $coT_{\Sigma,s}$ is the greatest set of finitely branching trees t of finite or infinite depth such that for all $f : s \to e_1 + \ldots + e_n \in F$, $n \in \mathbb{N}$, a unique arc a labelled with a pair (f, i), $1 \le i \le n$, emanates from the (unlabelled) root of t and either $e_i \in S$ and the target of a is in coT_{Σ,e_i} or e_i is a base set and the target of a is a leaf labelled with an element of e_i.
- For all $f : s \to e_1 + \ldots + e_n \in F$ and $t \in coT_{\Sigma,s}$, $f^{coT_\Sigma}(t)$ is the tree where the edge emanating from the root of t and labelled with (f, i) for some i points to.

Again, the interpretation of R in T_Σ is not fixed.

Theorem 3.12 If Σ admits coterms, then coT_Σ is an final Σ-algebra and for all Σ-algebras A, $unfold^A : A \to coT_\Sigma$ agrees with **coterm evaluation in** A: Let $s \in S$, $a \in A_s$, $f : s \to e \in F$ and $f^A(a) = (b_f, i_f)$. $\{(f, i_f) \mid f : s \to e \in F\}$ is the set of labels of the arcs emanating from the root of $unfold^A(a)$ and for all $f : s \to e \in F$, the outarc labelled with (f, i) points to $unfold^A(b_f)$. ❑

Example 3.13 (cf. Exs. 2.2 and 3.10) Since $DetAut(1, \mathbb{N})$ admits coterms (if $state^X$ is replaced by $state$), $coT_{DetAut(1,\mathbb{N})}$ is a final $DetAut(1, \mathbb{N})$-algebra. For instance, the stream $[1, 2, 3, \ldots]$ is represented in $coT_{DetAut(1,\mathbb{N})}$ as the following infinite tree:



We omitted the number component of the edge labels because it is always 1. ❑

Of course, the construction of a destructive signature from a constructive one can be reversed: Let $\Sigma = (S, F, P)$ be a destructive signature. Then

$$Co\Sigma = (S, \{c_s : \prod_{f:s \to e \in F} e \to s \mid s \in S\}, R)$$

is a constructive signature.

It remains to supply a construction for final Σ-algebras for *non-polynomial* destructive signatures where the range of some destructor involves the finite-set constructor *set*.

Given an S-sorted set M, a signature Σ is **M-bounded** if for all Σ-algebras A, $s \in S$ and $a \in A_s$, $|\langle a\rangle_s| \leq |M_s|$.

Example 3.14 (cf. Ex. 2.2) By [41], Ex. 6.8.2, or [20], Lemma 4.2, $H_{DetAut(X,Y)}$ is X^*-bounded: For all $DetAut(X,Y)$-algebras A and $a \in A_{state}$,

$$\langle st\rangle = \{\delta^{A*}(a)(w),\ w \in X^*\}$$

where $\delta^{A*}(a)(\epsilon) = st$ and $\delta^{A*}(a)(xw) = \delta^{A*}(\delta^A(a)(x))(w)$ for all $x \in X$ and $w \in X^*$. Hence $|\langle st\rangle| \leq |X^*|$. ❏

Example 3.15 (cf. Ex. 2.2) $H_{NDAut(X,Y)}$ is $(X^* \times \mathbb{N})$-bounded: For all $NDAut$-algebras A and $a \in A_{state}$, $\langle st\rangle = \cup\{\delta^{A*}(a)(w),\ w \in X^*\}$ where $a \in A_{state}$, $\delta^{A*}(a)(\epsilon) = \{st\}$ and $\delta^{A*}(a)(xw) = \cup\{\delta^{A*}(st')(w) \mid st' \in \delta^A(a)(x)\}$ for all $x \in X$ and $w \in X^*$. Since for all $a \in A_{state}$ and $x \in X$, $|\delta^A(a)(x)| \in \mathbb{N}$, $|\langle st\rangle| \leq |X^* \times \mathbb{N}|$. If $X = 1$, then $X^* \times \mathbb{N} \cong \mathbb{N}$ and thus $H_{NDAut(1,Y)}$ is $\mathbb{N}$-bounded (see [41], Ex. 6.8.1; [20], Section 5.1). ❏

Theorem 3.16 ([41], Thm. 10.6; [20], Cor. 4.9 and Section 5.1) All signatures are bounded. ❏

A destructive signature $\Sigma = (S, F, P)$ is **Moore-like** if there is an S-sorted set M such that for all $f : s \to e \in F$, $e = s^{M_s}$ or e is a base set. Then M is called the output of Σ.

Lemma 3.17 Let $\Sigma = (S, F, P)$ be a Moore-like signature with output M. Σ is polynomial and thus by Thm. 3.8, Alg_Σ has a final object A. If $|S| = 1$, then Σ agrees with $DetAut(M,Y)$ and thus $A \cong MinAut$ (see Exs. 2.6 and 3.9). Otherwise A can be constructed as a straightforward extension of $MinAut$ to several sorts: For all $s \in S$, $A_s = (M_s^* \to \prod_{g:s\to Z\in F} Z)$, and for all $f : s \to s^{M_s}$, $g : s \to Z \in F$ and $h \in A_s$, $f^A(h) = \lambda x.\lambda w.h(xw)$ and $g^A(h) = \pi_g(h(\epsilon))$.

A can be visualized as the S-sorted set of trees such that for all $s \in S$ and $h \in A_s$, the root r of h has $|M_s|$ outarcs, for all $g : s \to Z \in F$, r is labelled with $g^A(h)$, and for all $f : s \to s^{M_s}$ and $m \in M_s$, $f^A(h)(m) = \lambda w.h(mw)$ is the subtree of h where the m-th outarc of r points to. ❏

Theorem 3.18 Let M be an S-sorted set, $\Sigma = (S, F, P)$ be a destructive signature and $F' = \{f_s : s \to s^{M_s} \mid s \in S\} \cup \{f' : s \to M_e \mid f : s \to e \in F\}$. Of course, $\Sigma' = (S, F', R)$ is Moore-like. Let $\tau : H_{\Sigma'} \to H_\Sigma$ be the function defined as follows: For all S-sorted sets A, $f : s \to e \in F$ and $a \in H_{\Sigma'}(M)_s$, $\pi_f(\tau_{A,s}(a)) = F_e(\pi_{f_s}(a))(\pi_{f'}(a))$. τ is a surjective natural transformation.

Proof. The theorem is an adaption of [20], Thm. 4.7 (i)⇒(iv), and the definitions in its proof to our many-sorted syntax. ❏

Lemma 3.19 Let $\Sigma = (S, F, P)$ and $\Sigma' = (S, F', R)$ be destructive signatures, $\tau : H_{\Sigma'} \to H_\Sigma$ be a surjective natural transformation and A be final in $Alg_{\Sigma'}$. The following Σ-algebra B is **weakly final** (i.e., Σ-homomorphisms into B need not be unique): For all $s \in S$, $B_s = A_s$, and for all $f : s \to e \in F$, $f^B = \pi_f \circ \tau_{A,s} \circ \langle g_1, \ldots, g_n\rangle$ where $\{g_1, \ldots, g_n\} = \{g^A \mid g : s \to e \in F'\}$. $B/\!\sim$

final in Alg_Σ where $\sim$ is the greatest Σ-congruence on B, i.e. the union of all Σ-congruences on B.

Proof. The lemma is an adaption of [20], Lemma 2.3 (iv), and the definitions in its proof to our many-sorted syntax. ❑

Given an arbitrary destructive signature Σ, the previous results lead to a construction of the final Σ-algebra – provided that the bound is known:

Theorem 3.20 Let M be an S-sorted set, $\Sigma = (S, F, P)$ be a destructive signature and the Σ-algebra C be defined as follows: For all $s \in S$, $C_s = (M_s^* \rightarrow \prod_{f:s\rightarrow e\in F} M_e)$, and for all $f : s \rightarrow e \in F$ and $h \in C_s$, $f^C(h) = F_e(\lambda x.\lambda w.h(xw))(\pi_f(h(\epsilon)))$. $C/\!\sim$ is final in Alg_Σ where $\sim$ is the greatest Σ-congruence on C.

Proof. Let $F' = \{f' : s \rightarrow M_e \mid f : s \rightarrow e \in F\}$ and $\Sigma' = (S, F' \cup \{f_s : s \rightarrow s^{M_s} \mid s \in S\}, P)$. By Thm. 3.18, $\tau : H_{\Sigma'} \rightarrow H_\Sigma$ with $\pi_f(\tau_{A,s}(a)) = F_e(\pi_{f_s}(a))(\pi_{f'}(a))$ for all $A \in Set^S$, $f : s \rightarrow e \in F$ and $a \in H_{\Sigma'}(A)_s$ is a surjective natural transformation. Since Σ' is Moore-like, Lemma 3.17 implies that the following Σ'-algebra A is final: For all $s \in S$, $A_s = (M_s^* \rightarrow \prod_{f':s\rightarrow Z\in F'} Z = \prod_{f:s\rightarrow e\in F} M_e)$, and for all $h \in A_s$ and $f' : s \rightarrow Z \in F'$, $f_s^A(h) = \lambda m.\lambda w.h(mw)$ and $f'^A(h) = \pi_{f'}(h(\epsilon))$. By Lemma 3.19, the following Σ-algebra B is weakly final: For all $s \in S$, $B_s = A_s$, and for all $f : s \rightarrow e \in F$, $f^B = \pi_f \circ \tau_{A,s} \circ \langle g_1, \ldots, g_n\rangle$ where $\{g_1, \ldots, g_n\} = \{g^A \mid g : s \rightarrow e \in F'\}$. Hence for all $f : s \rightarrow e \in F$ and $h \in B_s$,

$$\begin{aligned} f^B(h) &= \pi_f(\tau_{A,s}(\langle g_1, \ldots, g_n\rangle(h))) = \pi_f(\tau_{A,s}(g_1(h), \ldots, g_n(h))) \\ &= F_e(\pi_{f_s}(g_1(h), \ldots, g_n(h)))(\pi_{f'}(g_1(h), \ldots, g_n(h))) = F_e(f_s^A(h))(f'^A(h)) \\ &= F_e(\lambda x.\lambda w.h(xw))(\pi_{f'}(h(\epsilon))) = F_e(\lambda x.\lambda w.h(xw))(\pi_f(h(\epsilon))) = f^C(h). \end{aligned}$$

We conclude that $C = B$ is weakly final. Hence by Lemma MOOREFIN, $C/\!\sim\ = B/\!\sim$ is final in Alg_Σ. ❑

Example 3.21 Let $M = M_{state} = X^* \times \mathbb{N}$, $Z_1 = F_{set(state)^X}(M) = \mathcal{P}_{fin}(M)^X$ and $Z_2 = F_Y(M) = Y$. By Ex. 3.15, $H_{NDAut(X,Y)}$ is M-bounded. Hence by Thm. 3.20, the following $NDAut(X, Y)$-algebra C is weakly final: $C_{state} = (M^* \rightarrow Z_1 \times Z_2)$ and for all $h \in C_{state}$ and $x \in X$, $h(\epsilon) = (g, y)$ implies

$$\begin{aligned} \delta^C(h)(x) &= F_{set(state)^X}(\lambda m.\lambda w.h(mw))(\pi_\delta(h(\epsilon)))(x) \\ &= F_{set(state)^X}(\lambda m.\lambda w.h(mw))(g)(x) = F_{set(state)}(\lambda m.\lambda w.h(mw))(g(x)) \\ &= \{F_{state}(\lambda m.\lambda w.h(mw))(m) \mid m \in g(x)\} \\ &= \{\lambda m.\lambda w.h(mw))(m) \mid m \in g(x)\} = \{\lambda w.h(mw) \mid m \in g(x)\}, \\ \beta^C(h) &= F_Y(\lambda x.\lambda w.h(xw))(\pi_\beta(h(\epsilon))) = F_Y(\lambda x.\lambda w.h(xw))(y) = id_Y(y) = y. \end{aligned}$$

Moroever, $C/\!\sim$ is a final Σ-algebra where $\sim$ is the greatest Σ-congruence on C, i.e. the union of all binary relations on C such that for all $h, h' \in C_{state}$,

$$h \sim h' \text{ implies } \delta^C(h) \sim_{set(state)^X} \delta^C(h') \ \wedge\ \beta^C(h) \sim_Y \beta^C(h'),$$

i.e., for all $x \in X$, $h \sim h'$, $h(\epsilon) = (g, y)$ and $h'(\epsilon) = (g', y')$ imply

$$\begin{aligned} &\forall\, m \in g(x)\ \exists n \in g'(x) : \lambda w.h(mw) \sim \lambda w.h'(nw) \ \wedge \\ &\forall\, n \in g'(x)\ \exists m \in g(x) : \lambda w.h(mw) \sim \lambda w.h'(nw) \ \wedge\ y = y'. \end{aligned}$$

Let $\Sigma' = (\{state\}, \{\delta : state \to Z_1, \beta : state \to Z_2, f : state \to state^M\}, \emptyset)$. By the proof of Thm. 3.20, C is constructed from the Σ'-algebra A such that $A_{state} = C_{state}$ and for all $h \in A_{state}$, $f^A_{state}(h) = \lambda m.\lambda w.h(mw)$ and $\langle \delta^A, \beta^A \rangle(h) = h(\epsilon) \in Z_1 \times Z_2$. Since Σ' is Moore-like, Lemma 3.17 implies that C_{state} can be visualized as the set of trees h such that the root r of h has $|M|$ outarcs, r is labelled with $h(\epsilon)$ and for all $m \in M$, $\lambda w.h(mw)$ is the subtree of h where the m-th outarc of r points to. See [21], Section 5, for a description of $C/\!\sim$ in the case $X = Y = 1$. ❏

4 Co/induction, abstraction, restriction, extension and co/recursion

After having shown in the previous sections how to build the domains of many-sorted initial or final models, let us turn to their analysis (by co/induction), the definition of functions on their domains (by co/recursion), their extension by further constructors resp. destructors, the factoring (abstraction) of initial models and the restriction of final one.

The dual operations of the last two, i.e., restriction of an initial model or abstraction of a final model, are impossible because an initial Σ-algebra has no Σ-invariants besides itself and a final Σ-algebra has no congruences besides the diagonal ([44], Thm. 4.3):

Lemma 4.1 (see Section 2) Let Σ be a constructive signature. (1) For all Σ-algebras A, $img(fold^A)$ is the least Σ-invariant of A. (2) If A is initial, then A is the only Σ-invariant of A.

Let Σ be a destructive signature. (3) For all Σ-algebras A, $ker(unfold^A)$ is the greatest Σ-congruence on A. (4) If A is final, then $\langle id_A, id_A \rangle(A)$ is the only Σ-congruence on A.

Proof of (2) and (4). Let Σ be constructive, A be initial and inv be a Σ-invariant of A. Then $inc \circ fold^{inv} = id_A$. Hence $inc \circ fold^{inv}$ and thus inc are surjective. We conclude that inv and A are Σ-isomorphic.

Let Σ be destructive, A be final and $\sim$ be a Σ-congruence on A. Then $unfold^{A/\sim} \circ nat_\sim = id_A$. Hence $unfold^{A/\sim} \circ nat_\sim$ and thus $nat_\sim$ are injective. We conclude that A and $A/\!\sim$ are Σ-isomorphic. ❏

By Lemma 4.1 (2) and (4), algebraic co/induction is sound:

Algebraic Induction. Let $\Sigma = (S, F, P)$ be a constructive signature, A be an initial Σ-algebra and $R \subseteq A$. $R = A$ iff $inv \subseteq R$ for some Σ-invariant inv of A. ❏

Algebraic Coinduction. Let $\Sigma = (S, F, P)$ be a destructive signature, A be a final Σ-algebra and $R \subseteq A^2$. $R \subseteq \langle id_A, id_A \rangle(A)$ iff $R \subseteq\sim$ for some Σ-congruence $\sim$ on A. ❏

In practice, an inductive proof of $R = A$ starts with $inv := R$ and stepwise decreases inv as long as inv is not an invariant. In terms of the formula φ that represents inv, each modification of inv is a conjunctive extension – usually called a *generalization* – of φ. The goal $R = A$ means that A satisfies φ.

Dually, a coinductive proof of $R = \langle id_A, id_A\rangle(A)$ starts with $\sim := R$ and stepwise increases $\sim$ as long as $\sim$ is not a congruence. In terms of the formula φ that represents $\sim$, each modification of $\sim$ is a disjunctive extension of φ. The goal $R = \langle id_A, id_A\rangle(A)$ means that A satisfies the equations given by φ.

Example 4.2 (see Exs. 2.6 and 3.10) Let A be a $DetAut(X,2)$-algebra. $\sim\subseteq A^2$ is a $DetAut(X,2)$-congruence iff for all $a, b \in A_{state}$ and $x \in X$, $a \sim b$ implies $\delta^A(a)(x) \sim \delta^A(b)(x)$ and $\beta^A(a)(x) = \beta^A(b)(x)$. Since the algebra $T = T_{Reg(X)}$ of regular expressions and the algebra $Lang$ of languages over X is a final $DetAut(X,2)$-algebra, $Lang$ is final and $unfold^T$ agrees with $fold^{Lang}$, two regular expressions R, R' have the same language (= image under $fold^{Lang}$) iff for some $w \in X^*$, the regular expressions $\delta^{T*}(R)(w)$ and $\delta^{T*}(R')(w)$ (see Ex. 3.14) have the same language (since, e.g., they are rewritable into each other by applying basic properties of regular operators). It is easy to see how this way of proving language equality can also be used for constructing the minimal acceptor $\langle L\rangle$ of the language L of a regular expression. ❑

Algebraic co/induction is a special case of predicate co/induction that applies to arbitrary Σ-algebra A and least resp. greatest interpretations of predicates of Σ in A. For ensuring that such interpretations exist, the predicates must be axiomatized in terms of *co/Horn clauses* [34, 35, 38, 39]:

Let $\Sigma = (S, F, P)$ be a signature and A be a Σ-algebra. A **Σ-formula** φ is a well-typed first-order formula built up from logical operators, symbols of $F \cup R$, liftings thereof (see Section 2) and elements of a fixed $\mathbb{T}(S)$-sorted set Var of variables. The interpretation φ^A of φ in A is the set of $\mathbb{T}(S)$-sorted *valuations* $f : Var \to A$ that satisfy φ. The interpretation $t^A : A^{Var} \to A$ of a term t occurring in φ is the $\mathbb{T}(S)$-sorted function that takes a valuation f and evaluates t in A under f. (For lack of space, we omit formal definitions here.) A Σ-formula $\varphi \Leftarrow \psi$ resp. $\varphi \Rightarrow \psi$ is a **Σ-Horn clause** resp. **Σ-co-Horn clause** if φ is an atom(ic formula) and ψ is negation-free.

Let $\Sigma = (S, F, P)$ be a signature, $\Sigma' = (S, F, \emptyset)$, C be a Σ'-algebra and $Alg_{\Sigma,C}$ be the category of all Σ-algebras A with $A|_{\Sigma'} = C$. $Alg_{\Sigma,C}$ is a *complete lattice*: For all $A, B \in Alg_{\Sigma,C}$, $A \le B \Leftrightarrow \forall\, p \in P : p^A \subseteq p^B$. For all $\mathcal{A} \subseteq Alg_{\Sigma,C}$ and $p : e \in P$, $p^\bot = \emptyset$, $p^\top = A_e$, $p^{\sqcup(\mathcal{A})} = \bigcup_{A\in\mathcal{A}} r^A$ and $p^{\sqcap(\mathcal{A})} = \bigcap_{A\in\mathcal{A}} r^A$. Let $\Phi : Alg_{\Sigma,C} \to Alg_{\Sigma,C}$ be a monotone function. $A \in Alg_{\Sigma,C}$ is **Φ-closed** if $\Phi(A) \le A$. A is **Φ-dense** if $A \le \Phi(A)$. The well-known fixpoint theorem of Knaster and Tarski provides fixpoints of Φ:

Theorem 4.3 $lfp(\Phi) = \sqcap\{A \in Alg_{\Sigma,C} \mid A \text{ is } \Phi\text{-dense}\}$ is the least and $gfp(\Phi) = \sqcup\{A \in Alg_{\Sigma,C} \mid A \text{ is } \Phi\text{-closed}\}$ is the greatest fixpoint of Φ. ❑

Obviously, for all negation-free formulas φ and $A, B \in Alg_{\Sigma,C}$, $A \le B$ implies $\varphi^A \subseteq \varphi^B$. A set AX of Σ-formulas that consists of only Horn clauses or only co-Horn clauses induces a monotone function $\Phi : Alg_{\Sigma,C} \to Alg_{\Sigma,C}$: For all $A \in Alg_{\Sigma,C}$ and $p : e \in P$, $p^\Phi(A) = \{(t^A(f) \mid f \in \varphi^A,\ r(t) \Leftarrow \varphi \in AX\}$ if AX consists of Horn clauses and $p^\Phi(A) = A_e \setminus \{(t^A(f) \mid f \in A^{Var} \setminus \varphi^A,\ p(t) \Rightarrow \varphi \in AX\}$ if AX consists of co-Horn clauses. Hence by Thm. 4.3, Φ has a least fixpoint $lfp(\Sigma, C, AX) = lfp(\Phi)$ and a greatest fixpoint $gfp(\Sigma, C, AX) = gfp(\Phi)$. In other words, lfp and gfp are the least resp. greatest $A \in Alg_{\Sigma,C}$ that satisfy AX, or, if we regard the predicate symbols in AX as variables, then $\{p^{lfp} \mid p \in P\}$ is the least and $\{p^{gfp} \mid p \in P\}$ is the greatest solution of AX in P. This implies immediately that predicate co/induction is sound:

Predicate Induction Let AX be a set of Horn clauses. $lfp = lfp(\Sigma, C, AX)$ satisfies $p(x) \Rightarrow \psi(x)$ iff there is a formula $\psi'(x)$ such that for all $p(t) \Leftarrow \varphi \in AX$, lfp satisfies $p(t) \Leftarrow \varphi'$ where φ' is obtained from φ by replacing all occurrences of atoms $p(u)$ with $\psi(u) \wedge \psi'(u)$. ❏

Predicate Coinduction Let AX be a set of co-Horn clauses. $gfp = gfp(\Sigma, C, AX)$ satisfies $p(x) \Leftarrow \psi(x)$ iff there is a formula $\psi'(x)$ such that for all $p(t) \Rightarrow \varphi \in AX$, gfp satisfies $r(t) \Rightarrow \varphi'$ where φ' is obtained from φ by replacing all occurrences of atoms $p(u)$ with $\psi(u) \vee \psi'(u)$. ❏

$Alg_{\Sigma,AX}$ denotes the category of all Σ-algebras that satisfy AX. Co/Horn clause syntax admits four ways of axiomatizing invariants resp. congruences and thus restricting resp. factoring initial or final models: Let $\Sigma = (S, F, P)$ be a signature and $\Sigma' = (S, F, \emptyset)$.

Theorem 4.4 (abstractions) [36] For all $s \in S$, let $r : s \times s \in P$ and C be initial in $Alg_{\Sigma'}$.

(1) Suppose that AX is a set of Horn clauses such that for all $A \in Alg_{\Sigma,AX}$, r^A is a Σ-congruence. Let $lfp = lfp(\Sigma, C, AX)$. If AX meets certain syntactical restrictions, then the quotient of lfp by r^{lfp} is initial in the category $\mathcal{K}$ of all algebras of $Alg_{\Sigma,C}$ that satisfy AX and interpret $r : s \times s$ as $\langle id, id\rangle(C_s)$.

(2) Suppose that AX is a set of co-Horn clauses such that for all $A \in Alg_{\Sigma,AX}$, r^A is a Σ-congruence. Let $gfp = gfp(\Sigma, C, AX)$. If AX meets certain syntactical restrictions, then the quotient of gfp by r^{gfp} is final in the category of all F-*reachable* algebras of $\mathcal{K}$ (see below). r^{gfp} coincides with the final semantics [28, 48, 49] deal with. ❏

Theorem 4.5 (restrictions) [36] For all $s \in S$, let $r : s \in P$ and C be final in $Alg_{\Sigma'}$.

(1) Suppose that AX is a set of co-Horn clauses such that for all $A \in Alg_{\Sigma,AX}$, r^A is a Σ-invariant. Let $gfp = gfp(\Sigma, C, AX)$. If AX meets certain syntactical restrictions, then r^{gfp} is final in the category $\mathcal{K}$ of all algebras of $Alg_{\Sigma,C}$ that satisfy AX and interpret $r : s$ as C_s.

(2) Suppose that AX is a set of Horn clauses such that for all $A \in Alg_{\Sigma,AX}$, r^A is a Σ-invariant. Let $lfp = lfp(\Sigma, C, AX)$. If AX meets certain syntactical restrictions, then r^{lfp} is initial in the category of all F-*observable* algebras of $\mathcal{K}$ (see below). ❏

Given a signature $\Sigma = (S, F, P)$ and a set AX of Σ-formulas, $Alg_{\Sigma,AX}$ denotes the full subcategory of Alg_Σ whose objects satisfy (all formulas of) AX. Let $\Sigma' = (S', F', P')$ be a subsignature of Σ, AX be a set Σ-formulas, $AX' \subseteq AX$ be a set Σ'-formulas, A be a Σ-algebra and $B = A|_{\Sigma'}$.

Let Σ be constructive and $\mu\Sigma$ and $\mu\Sigma'$ be initial in $Alg_{\Sigma,AX}$ resp. $Alg_{\Sigma',AX'}$. A is **F'-reachable** or -generated if $fold^B : \mu\Sigma' \to B$ is surjective. A is **F'-consistent** if $fold^B$ is injective. (Σ, AX) is a **conservative extension** of (Σ', AX') if $\mu\Sigma$ is F'-reachable and F'-consistent, i.e. if $\mu\Sigma|_{\Sigma'}$ and $\mu\Sigma'$ are isomorphic.

Let Σ be destructive and $\nu\Sigma'$ and $\nu\Sigma$ be final in $Alg_{\Sigma,AX}$ resp. $Alg_{\Sigma',AX'}$. A is F'**-observable** or -cogenerated if $unfold^B : B \to \nu\Sigma'$ is injective. A is F'**-complete** if $unfold^B$ is surjective. (Σ, AX) is a **conservative extension** of (Σ', AX') if $\nu\Sigma$ is F'-observable and F'-complete, i.e. $\nu\Sigma|_{\Sigma'}$ and $\nu\Sigma'$ are isomorphic.

Proposition 4.6 [36] Let Σ be constructive. If A is initial in $Alg_{\Sigma,AX}$, then A is F'-reachable iff $img(fold^B)$ is a Σ-invariant. If $\mu\Sigma'$ can be extended to an algebra of (Σ, AX)-algebra, then (Σ, AX) is a conservative extension of (Σ', AX').

Let Σ be destructive. If A is final in $Alg_{\Sigma,AX}$, then A is F'-observable iff $ker(unfold^B)$ is a Σ-congruence. If $\nu\Sigma'$ can be extended to a (Σ, AX)-algebra, then (Σ, AX) is a conservative extension of (Σ', AX'). ❑

Conservative extensions add constructors or destructors to a signature without changing the carrier of the initial resp. final model. Each other functions can be axiomatized in terms of co/recursive equations, which means that there is a Σ-algebra A such that f agrees with $fold^A$ resp. $unfold^A$. By Prop. 2.4 (2), this holds true iff f is simply an S-sorted function whose kernel resp. image is compatible with F. (The use Prop. 2.4 (2) for co/recursive definitions on initial resp. final co/algebras was first suggested by [16], Thm. 4.2 resp. 5.2.) However, as constructors and destructors usually are not (components of) S-sorted functions, the domain or range of f is seldom a single sort $s \in S$, but a composed type $e \in \mathbb{T}(S)$. Hence we follow [23] and start out from a category $\mathcal{K}$ and an adjunction between $\mathcal{K}$ and Set^S such that f can be described as a $\mathcal{K}$-morphism, while $fold^A$ resp. $unfold^A$ comes up as the unique Set^S-extension of f that the adjunction generates:

Let $\Sigma = (S, F, P)$ be a constructive signature, $\mathcal{K} = \prod_{s\in S} \mathcal{K}_s$ be a product category and $(L : Set^S \to \mathcal{K}, G : \mathcal{K} \to Set^S, \eta, \epsilon)$ be an adjunction. A $\mathcal{K}$-morphism $f : L(\mu\Sigma) \to B$ is Σ**-recursive** if the kernel of the Set^S-extension $f^\# : \mu\Sigma \to G(B)$ of f is compatible with F.

Let $\Sigma = (S, F, P)$ be a destructive signature, $\mathcal{K} = \prod_{s\in S} \mathcal{K}_s$ be a product category and $(L : \mathcal{K} \to Set^S, G : Set^S \to \mathcal{K}, \eta, \epsilon)$ be an adjunction. A $\mathcal{K}$-morphism $f : A \to G(\nu\Sigma)$ is Σ**-corecursive** if the image of the Set^S-extension $f^* : L(A) \to \nu\Sigma$ of f is compatible with F.

Example 4.7 The factorial function $fact : \mathbb{N} \to \mathbb{N}$ is usually axiomatized by the following equations involving the constructors $0 : 1 \to nat$ and $succ : nat \to nat$ of Nat (see Ex. 2.1):

$$fact(0) = 1, \quad fact(n+1) = fact(n) * (n+1).$$

Since by Ex. 3.8, $\mathbb{N}$ is an initial Nat-algebra, we may show that $fact$ is Nat-recursive. This cannot be concluded from the above equations because the variable n occurs at a non-argument position. Hence we add the identity on $\mathbb{N}$ and show that the desired property for $fact$ and id simultaneously. The corresponding equations read as follows:

$$\langle fact, id\rangle(0) = (1, 0), \quad \langle fact, id\rangle(n+1) = (fact(n) * (id(n)+1), id(n)+1).$$

We choose the product adjunction

$$((\_,\_) : Set \to Set^2, \times : Set^2 \to Set, \lambda A.\langle id_A, id_A\rangle, (\pi_1, \pi_2)).$$

The latter equations imply that the kernel of the Set-extension $(fact, id)^\# = \langle fact, id\rangle : \mathbb{N} \to \mathbb{N}^2$ of $(fact, id) : (\mathbb{N}, \mathbb{N}) \to (\mathbb{N}, \mathbb{N})$ is compatible with 0 and $succ$. Hence $(fact, id)$ is Nat-recursive and by Prop. 2.4 (2), $\langle fact, id\rangle$ is a Nat-homomorphism, in particular, $\mathbb{N}^2$ is a Nat-algebra: $0^{\mathbb{N}^2} = (1, 0)$ and $succ^{\mathbb{N}^2} = \lambda(m, n).(m * (n + 1), n + 1)$. Hence $fold^{\mathbb{N}^2} = (fact, id)^\#$. ❏

Example 4.8 The streams $\overline{01} = [0, 1, 0, 1, \ldots]$ and $\overline{10} = [1, 0, 1, 0, \ldots]$ can be axiomatized by the following equations involving the destructors $\delta : state \to state$ and $\beta : state \to 2$ of $DetAut(1, 2)$ (see Ex. 2.2):

$$\langle \delta, \beta\rangle(\overline{01}) = (\overline{10}, 0), \qquad \langle \delta, \beta\rangle(\overline{10}) = (\overline{01}, 1) \tag{1}$$

Since by Ex. 3.10, $2^{\mathbb{N}}$ is a final $DetAut(1, 2)$-algebra, we may show that $(\overline{01}, \overline{10})$ is $DetAut(1, 2)$-corecursive. We choose the coproduct adjunction

$$(+ : Set^2 \to Set, (\_,\_) : Set \to Set^2, (\iota_1, \iota_2), \lambda A.[id_A, id_A]).$$

The above equations imply that the image of the Set-extension $(\overline{01}, \overline{10})^* = [\overline{01}, \overline{10}] : 2 \to 2^{\mathbb{N}}$ of $(\overline{01}, \overline{10}) : (1, 1) \to (2^{\mathbb{N}}, 2^{\mathbb{N}})$ is compatible with δ and β. Hence $(\overline{01}, \overline{10})$ is $DetAut(1, 2)$-corecursive and by Prop. 2.4 (2), $[\overline{01}, \overline{10}]$ is a $DetAut(1, 2)$-homomorphism, in particular, 2 is a $DetAut(1, 2)$-algebra: $\delta^2(0) = 1$, $\delta^2(1) = 0$ and $\beta^2 = id_2$. Hence $unfold^2 = (\overline{01}, \overline{10})^*$.

Since for all sets 2, $DetAut(1, 2)$ admits coterms, $DetAut(1, 2)$ induces the constructive signature $CoDetAut(1, 2) = (\{state\}, \{cons : 2 \times state \to state\}, \emptyset)$ that admits terms (see Section 3). It does not matter that *initial* $CoDetAut(1, 2)$-algebras are empty. Here we only use the syntax of $CoDetAut(1, 2)$: The streams $\overline{01}$ and $\overline{10}$ can be axiomatized by equations involving $cons$:

$$\overline{01} = cons(0, \overline{10}), \qquad \overline{10} = cons(1, \overline{01}). \tag{2}$$

The definition of $\overline{01}$ and $\overline{10}$ derived from (1) provides a *solution* of (2) where $\overline{01}$ and $\overline{10}$ are regarded as variables. Conversely, each solution (a, b) of (2) has the unique Set-extension $[a, b]$, which is a $DetAut(1, 2)$-homomorphism into a final $DetAut(1, 2)$-algebra and thus unique. Hence (2) has a unique solution! ❏

The last observation can be generalized to the following result obtained in several ways and on many levels of abstraction (see, e.g., [**?**], Thm. 5.2; [4], Thm. 3.3): Given a constructive signature Σ that admits terms, *ideal* or *guarded* Σ-equations like (2) have unique solutions in CT_Σ (see Section 3). Via this result, coalgebra has even found its way into functional programming (see, e.g. [43, 24]).

References

1. J. Adámek, *Free algebras and automata realizations in the language of categories*, Commentat. Math Univers. Carolinae 15 (1974) 589-602

2. J. Adámek, *Introduction to Coalgebra*, Theory and Applications of Categories 14 (2005) 157-199
3. J. Adámek, *Final coalgebras are ideal completions of initial algebras*, Journal of Logic and Computation 12 (2002) 217-242
4. P. Aczel, J. Adámek, J. Velebil, *A Coalgebraic View of Infinite Trees and Iteration*, Proc. Coalgebraic Methods in Computer Science, Elsevier ENTCS 44 (2001) 1-26
5. J. Adámek, S. Milius, L.S. Moss, *Initial algebras and terminal coalgebras: a survey*, draft of Feb. 7, 2011, TU Braunschweig
6. J. Adámek, H.–E. Porst, *From varieties of algebras to covarieties of coalgebras*, Proc. Coalgebraic Methods in Computer Science, Elsevier ENTCS 44 (2001) 27–46
7. J. Adámek, H.–E. Porst, *On Tree Coalgebras and Coalgebra Presentations*, Theoretical Computer Science 311 (2004) 257-283
8. M.A. Arbib, *Free dynamics and algebraic semantics*, Proc. Fundamentals of Computation Theory, Springer LNCS 56 (1977) 212-227
9. M.A. Arbib, E.G. Manes, *Parametrized Data Types Do Not Need Highly Constrained Parameters*, Information and Control 52 (1982) 139-158
10. E. Astesiano, H.-J. Kreowski, B. Krieg-Brückner, eds., *Algebraic Foundations of Systems Specification*, IFIP State-of-the-Art Report, Springer 1999
11. M. Barr, *Coequalizers and Free Triples*, Math. Zeitschrift 116 (1970) 307-322
12. M. Barr, *Terminal coalgebras in well-founded set theory*, Theoretical Computer Science 114 (1993) 299-315
13. M.G.J. van den Brand, J. Heering, P. Klint, P.A. Olivier, *Compiling Rewrite Systems: The ASF+SDF Compiler*, ACM TOPLAS 24 (2002)
14. J.A. Brzozowski, *Derivatives of regular expressions*, Journal ACM 11 (1964) 481–494
15. H. Ehrig, B. Mahr, *Fundamentals of Algebraic Specification 1*, Springer 1985
16. J. Gibbons, G. Hutton, Th. Altenkirch, *When is a function a fold or an unfold?*, Proc. Coalgebraic Methods in Computer Science, Elsevier ENTCS 44 (2001) 146-159
17. J. Goguen, G. Malcolm, *A Hidden Agenda*, Theoretical Computer Science 245 (2000) 55-101
18. J.A. Goguen, J.W. Thatcher, E.G. Wagner, *An Initial Algebra Approach to the Specification, Correctness and Implementation of Abstract Data Types*, in: R. Yeh, ed., Current Trends in Programming Methodology 4, Prentice-Hall (1978) 80-149
19. J.A. Goguen, J.W. Thatcher, E.G. Wagner, J.B. Wright, *Initial Algebra Semantics and Continuous Algebras*, J. ACM 24 (1977) 68-95
20. H.P. Gumm, T. Schröder, *Coalgebras of bounded type*, Math. Structures in Computer Science 12 (2002),565-578
21. H.P. Gumm, *Universelle Coalgebra*, in: Th. Ihringer, *Allgemeine Algebra*, Heldermann Verlag 2003
22. J. Guttag, E. Horowitz, D.R. Musser, *Abstract Data Types and Software Validation*, Communications of the ACM 21 (1978) 1048-1064
23. R. Hinze, *Adjoint Folds and Unfolds*, Proc. Mathematics of Program Construction 2010, Springer LNCS 6120, 195–228,
24. R. Hinze, *Reasoning about Codata*, Proc. CEFP 2009, Springer LNCS 6299 (2010) 42-93
25. B. Jacobs, *Invariants, Bisimulations and the Correctness of Coalgebraic Refinements*, Proc. Algebraic Methodology and Software Technology, Springer LNCS 1349 (1997) 276-291
26. B. Jacobs, *Introduction to Coalgebra*, Radboud University Nijmegen 2005

27. B. Jacobs, *A Bialgebraic Review of Deterministic Automata, Regular Expressions and Languages*, in: K. Futatsugi et al. (eds.), Goguen Festschrift, Springer LNCS 4060 (2006) 375–404
28. S. Kamin, *Final Data Type Specifications: A New Data Type Specification Method*, ACM TOPLAS 5 (1983) 97-123
29. J. Lambek, *A fixpoint theorem for complete categories*, Math. Zeitschrift 103 (1968) 151-161
30. D.J. Lehmann, M.B. Smyth, *Algebraic Specification of Data Types: A Synthetic Approach*, Math. Systems Theory 14 (1981) 97-139
31. J. Meseguer, J.A. Goguen, *Initiality, Induction and Computability*, in: M. Nivat, J. Reynolds, eds., Algebraic Methods in Semantics, Cambridge University Press (1985) 459-541
32. J. Meseguer, G. Rosu, *The Rewriting Logic Semantics Project*, Theoretical Computer Science 373 (2007)
33. E.A. van der Meulen, *Deriving incremental implementations from algebraic specifications*, Proc. 2nd AMAST, Springer (1992) 277-286
34. P. Padawitz, *Proof in Flat Specifications*, in: E. Astesiano, H.-J. Kreowski, B. Krieg-Brückner, eds., *Algebraic Foundations of Systems Specification*, IFIP State-of-the-Art Report, Springer (1999) 321-384
35. P. Padawitz, *Swinging Types = Functions + Relations + Transition Systems*, Theoretical Computer Science 243 (2000) 93-165
36. P. Padawitz, *Dialgebraic Specification and Modeling*, slides, TU Dortmund 2011, fldit-www.cs.tu-dortmund.de/∼peter/DialgSlides.pdf
37. P. Padawitz, *Algebraic compilers and their implementation in Haskell*, Sierra Nevada IFIP WG 1.3 meeting (January 14-18th, 2008)
38. P. Padawitz, *Algebraic Model Checking*, in: F. Drewes, A. Habel, B. Hoffmann, D. Plump, eds., Manipulation of Graphs, Algebras and Pictures, Electronic Communications of the EASST 26 (2010); extended slides: fldit-www.cs.tu-dortmund.de/∼peter/CTL.pdf
39. P. Padawitz, *Expander2 as a Prover and Rewriter*, fldit-www.cs.tu-dortmund.de/∼peter/Prover.pdf
40. P. Padawitz, *Übersetzerbau*, course notes, TU Dortmund 2010, fldit-www.cs.tu-dortmund.de/∼peter/CbauFolien.pdf
41. J.J.M.M. Rutten, *Universal Coalgebra: A Theory of Systems*, Theoretical Computer Science 249 (2000) 3-80
42. J.J.M.M. Rutten, *Automata and coinduction (an exercise in coalgebra)*, Proc. CONCUR '98, Springer LNCS 1466 (1998) 194–218
43. J.J.M.M. Rutten, *Behavioural differential equations: a coinductive calculus of streams, automata, and power series*, Theoretical Computer Science 308 (2003) 1-53
44. J.J.M.M. Rutten, D. Turi, *Initial Algebra and Final Coalgebra Semantics for Concurrency*, Report CS-R9409, CWI, Amsterdam 1994
45. K. Sen, G. Rosu, *Generating Optimal Monitors for Extended Regular Expressions*, Proc. Runtime Verification 2003, Elsevier ENTCS 89 (2003) 226-245
46. J.W. Thatcher, E.G. Wagner, J.B. Wright, *More on Advice on Structuring Compilers and Proving Them Correct*, Theoretical Computer Science 15 (1981) 223-249
47. E. Visser, *Program Transformation with Stratego/XT: Rules, Strategies, Tools, and Systems*, in: C. Lengauer et al., eds., Domain-Specific Program Generation, Springer LNCS 3016 (2004)
48. M. Wand, *Final algebra semantics and data type extension*, J. Comp. Syst. Sci. 19 (1979) 27-44

49. M. Wirsing, *Algebraic Specification*, in: J. van Leeuwen, ed., Handbook of Theoretical Computer Science, Elsevier (1990) 675-788