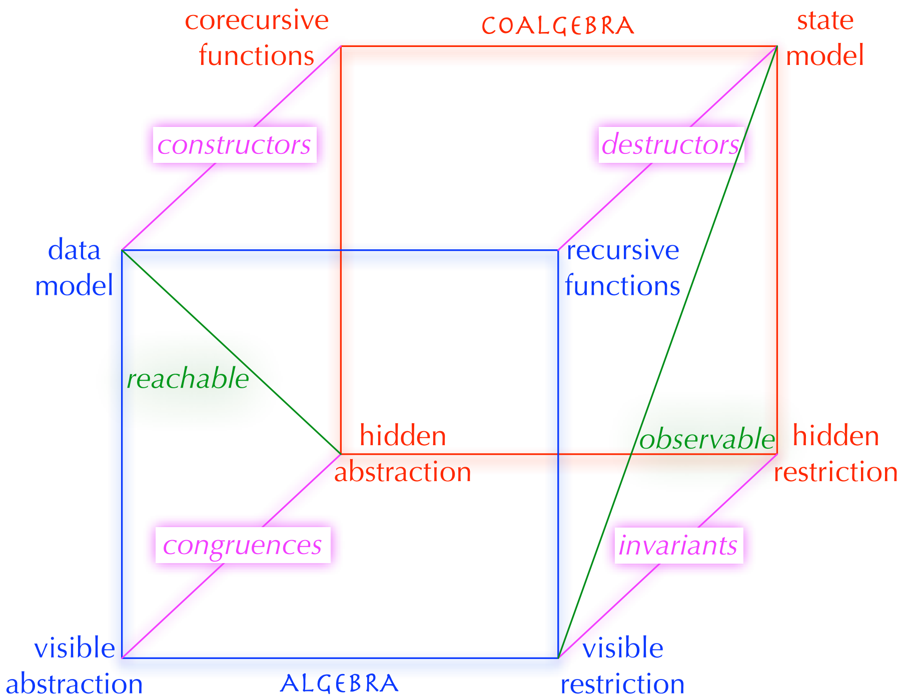
Dialgebraic Specification and Modeling

Peter Padawitz
http://padawitz.de
University of Dortmund, Germany
January 12, 2024



Abstract

Dialgebraic specifications combine algebraic with coalgebraic ones. We present a uniform syntax, semantics and proof system for chains of signatures and axioms such that presentations of *visible data types* may alternate with those of *hidden state types*. Each element in the chain matches a design pattern that reflects some least/initial or greatest/final model construction over the respective predecessor in the chain. We sort out twelve such design patterns. Six of them lead to least models, the other six to greatest models. Each construction of the first group has its dual in the second group. All categories used in this approach are classes of structures with many-sorted carrier sets. The model constructions could be generalized to other categories, but this is not a goal of our approach. On the contrary, we aim at applications in software (and, maybe, also hardware) design and will show that for describing and solving problems in this area one need not go beyond categories of sets. Consequently, a fairly simple, though sufficiently powerful, syntax of dialgebraic specifications that builds upon classical algebraic notations as much as possible is crucial here. It captures the main constructions of both universal (co)algebra and relational fixpoint semantics and thereby extends "Lawvere-style" algebraic theories from product to arbitrary polynomial types and modal logics from one- to many-sorted Kripke frames.

Contents

1 Introduction

??? Swinging types provide a one-tiered approach to the axiomatic specification of systems: static and dynamic components as well as structural and behavioral aspects are treated within the same (many-sorted) logic. A swinging type (ST) as defined in [89] is non-hierarchical. Standard hidden constructors are the tupling operators for product sorts and injections for sum sorts. In this paper, we drop the implicit assumption of [89] that only countable, term-generated data domains are specified in terms of swinging types.

The elements of the standard models are easy to conceive: those of the Herbrand SP-model are the ground Σ-terms. If SP is functional, the initial SP-model is isomorphic to the *normal form model* $NF(SP)$ whose elements are the ground Σ-normal forms, i.e., the Σ-terms consisting of constructors. The final SP-model is isomorphic to the quotient of $NF(SP)$ that identifies behaviorally equivalent normal forms.

If SP is the domain completion of a coalgebraic swinging type (see Section 4.2), most nullary constructors are (names of) elements of a final coalgebra and thus each of them denotes the "behavior" of an object in a certain state (see Section 4.1). In this case, non-nullary constructors describe state modifiers, in object-oriented terminology: methods. In other cases, they just build up static data structures.

Hence a normal form may represent

- the static structure of an object *or*
- the history of an object *or*
- a set of attribute values *or*
- a composition of substates *or*
- possible actions on an object *or*
- an "infinite" data structure *or*
- an element of another uncountable domain.

The most common hidden constructors are injections into sum sorts. Sum sorts are crucial for integrating coalgebraic specifications into STs because are based on *destructors* and the most interesting destructors map into sum sorts (see Sections 4 and 5). Hidden constructors can be regarded as the algebraic counterpart to destructors whose nature is coalgebraic. As injections into sum sorts are implicit constructors, so are projections into the factors of a product sort implicit destructors. Other common destructors are the *apply* operators that occur implicitly in axioms for higher-order defined functions (cf., e.g., Example 14.2).

Section 3 summarizes the crucial notions and results from category theory and universal algebra that justify the initial algebra/final coalgebra semantics of data types. We focus on the category Set^S of S-sorted sets. Some connections to swinging types are already drawn in this section. The new approach, however, is presented in Sections 4 and 5 and can be followed even if one has not read all details of Section 3.

Section 4 introduces notions of coalgebraic signatures and specifications, called *cosignatures* and *cospecifications*, respectively. A cosignature Δ starts out from a swinging signature $vis\Sigma$ as the syntactical basis for hidden sorts and destructors for them. Given a $vis\Sigma$-structure C, a (Δ, C)-coalgebra is merely a Δ-algebra whose $vis\Sigma$-reduct agrees with C. Δ induces *coterms* and *contexts*. Each element of the final (Δ, C)-coalgebra is a sequence of functions each of which interprets a context. A cospecification CSP adds to Δ and C *auxiliary functions*, *cofunctions*, *copredicates*, *inductive axiomatizations* for the auxiliary functions, *coinductive axiomatizations* for the cofunctions, co-Horn axioms for the copredicates and *assertions*. The final CSP-model , $Fin(CSP)$, is the subcoalgebra of the final (Δ, C)-coalgebra consisting of all data satisfying the assertions.those elements that

In Section 5, swinging types are combined with cospecifications. A swinging type SP becomes *dialgebraic* if SP is extended by a cospecification CSP such that SP and CSP have the same visible subsignature, the same hidden sorts, the same destructors for these sorts and the same axioms for the auxiliary functions and copredicates of CSP. Hence cofunctions and assertions are the only actual contributions of a cospecification to a dialgebraic ST. Semantically, however, there is a difference between a dialgebraic type CST and an algebraic one even if the cospecification CSP that is part of CST lacks assertions and cofunctions. While the standard model of an algebraic ST is an—always countable—quotient of a Herbrand model, the standard model of CST should be given by $Fin(CSP)$ and thus may have uncountable carriers (see above). Fortunately, the main result of Section 5 (Thm. 18.5) tells us that, under certain weak conditions, $Fin(CSP)$ is isomorphic to the standard model of an algebraic ST, called the *domain completion* of CST. By replacing CST with its domain completion, we may keep to the hierarchical notion of a swinging type, both syntactically and semantically.

2 Set-theoretical preliminaries

At first, we recall some basic notions of set theory and relation algebra. Let S be a set. $id_S : S \to S$ denotes the identity on S.

Given a **product** $A = \prod_{i \in I} A_i =_{def} \{(a_i)_{i \in I} \mid \forall i \in I : a_i \in A_i\}$ and $i \in I$, a_i denotes the i-th component of $a \in A$ and $\pi_i : A \to A_1$ denotes the **i-th projection** from A, i.e., for all $\pi_i(a) = a_i$. Moreover, for all $a, b \in A$,

$$\langle a, b\rangle \in \prod_{i \in I} A_i^2 \quad =_{def} \quad (a_i, b_i)_{i \in I}.$$

If I is a singleton, say $I = \{k\}$, we write k instead of I and A_k instead of A. A function $f : C \to A$ is well-defined iff for all $i \in I$, $\pi_i \circ f$ is well-defined. Given functions $f_i : C \to A_i$, $i \in I$, the **product extension** $\langle f_i\rangle_{i \in I} : C \to A$ of $\{f_i \mid i \in I\}$ is defined as follows: For all $c \in C$, $\langle f_i\rangle_{i \in I}(c) = (f_i(c))_{i \in I}$.

Relational update. Let $R \subseteq A = \prod_{i \in I} A_i$, $k \in I$, $a \in A$ and $b \in A_k$ $a[b/k] \in A$ and $R[b/k] \subseteq A$ are defined as follows:

$$\begin{aligned} (a[b/k])_i &= \begin{cases} b & \text{if } i = k \\ a_i & \text{otherwise} \end{cases} \\ R[b/k] &= \{a[b/k] \mid a \in R\}. \end{aligned}$$

Analogously, let $R \subseteq A = \coprod_{i \in I} A_i$, $k \in I$, $(a, i) \in A$ and $b \in A_k$. $(a, i)[b/k] \in A$, $i \in I$ and $R[b/k] \subseteq A$ are defined as follows:

$$\begin{aligned} ((a, i)[b/k]) &= \begin{cases} (b, i) & \text{if } i = k \\ (a, i) & \text{otherwise} \end{cases} \\ R[b/k] &= \{a[b/k] \mid a \in R\}. \end{aligned}$$

Relational product. Let $R \subseteq \prod_{i \in I} A_i$ and $R \subseteq \prod_{i \in J} A_i$ such that $I \cap J = \emptyset$.

$$R \times R' \quad =_{def} \quad \{(a_i)_{i \in I} \in \prod_{i \in I \cup J} A_i \mid (a_i)_{i \in I} \in R,\ (a_i)_{i \in J} \in R'\}.$$

Given a **sum** or **coproduct** $A = \coprod_{i \in I} A_i =_{def} \{(a, i) \mid i \in I,\ a \in A_i\}$ and $i \in I$, $\iota_i : A_i \to A$ denotes the **i-th injection** into A, i.e., $\iota_i(a) = (a, i)$. If I is a singleton, say $I = \{k\}$, we write k instead of I and A_k instead of A. A function $f : A \to C$ is well-defined iff for all $i \in I$, $f \circ \iota_i$ is well-defined. Given functions $f_i : A_i \to C$,

$i \in I$, the **sum extension** $[f_i]_{i\in I} : A \to C$ of $\{f_i \mid i \in I\}$ is defined as follows: For all $i \in I$ and $a \in A_i$, $[f_i]_{i\in I}(a) = f_i(a)$.

For all $f : \prod_{i\in I} A_i \to A$, $k \in I$ and $g : B \to A_k$, $f \circ_k g =_{def} f \circ \langle g_i\rangle_{i\in I} : \prod_{i\in I} B_i \to A$ where

$$g_i = \begin{cases} g & \text{if } i = k \\ \pi_i & \text{otherwise} \end{cases} \qquad B_i = \begin{cases} B & \text{if } i = k \\ A_i & \text{otherwise} \end{cases}$$

For all $f : A \to \coprod_{i\in I} A_i$, $k \in I$ and $g : A_k \to B$, $g \circ^k f =_{def} [g_i]_{i\in I} \circ f : A \to \coprod_{i\in I} B_i$ where

$$g_i = \begin{cases} g & \text{if } i = k \\ \iota_i & \text{otherwise} \end{cases} \qquad B_i = \begin{cases} B & \text{if } i = k \\ A_i & \text{otherwise} \end{cases} \quad ❏$$

For all functions $f_i : A_i \to B_i$, $i \in I$,

$$\prod_{i\in I} f_i =_{def} \langle f_i \circ \pi_i\rangle_{i\in I} : \prod_{i\in I} A_i \to \prod_{i\in I} B_i,$$
$$\coprod_{i\in I} f_i =_{def} [\iota_i \circ f_i]_{i\in I} : \coprod_{i\in I} A_i \to \coprod_{i\in I} B_i.$$

For all sets or functions $A_1, \dots, A_n$, $\prod_{i=1}^n A_i$ and $\coprod_{i=1}^n A_i$ are also written as $A_1 \times \dots \times A_n$ and $A_1 + \dots + A_n$, respectively. Some notations used above will re-appear in the following syntax of types, terms and formulas.

3 Types, terms and formulas, and their interpretations

Definition 3.1 (types) Let S be a finite set of **sorts** (type constants) and X be a finite set of type variables. The set $\mathbb{T}_S(X)$ of (polynomial) **types** over S and X are the sets of expressions generated by the following rules:

$\mathbb{T}_S(X)$

sorts and type variables

$$\frac{}{s}\ s \in S \cup \{1\} \qquad \frac{}{x}\ x \in X$$

product and sum

$$\frac{e_1, \dots, e_n}{e_1 \times \dots \times e_n} \qquad \frac{e_1, \dots, e_n}{e_1 + \dots + e_n}$$

powers

$$\frac{e}{s \to e}\ s \in S$$

constructive and destructive types over $Y \subseteq X$

$$\frac{}{\coprod_{i=1}^n \prod_{j=1}^{n_i} e_{ij}}\ \forall\, 1 \le i \le n : \forall\, 1 \le j \le n_i : e_{ij} \in S \cup Y$$

$$\frac{}{\prod_{i=1}^n (s_i \to \coprod_{j=1}^{n_i} e_{ij})}\ \forall\, 1 \le i \le n : s_i \in S \wedge \forall\, 1 \le j \le n_i : e_{ij} \in S \cup Y$$

recursive and corecursive types

$$\frac{e_1, \dots, e_n}{\mu x_1 \dots x_n.(e_1, \dots, e_n)}\ e_1, \dots, e_n \text{ are constructive over } \{x_1, \dots, x_n\} \subseteq X$$

$$\frac{e_1, \dots, e_n}{\nu x_1 \dots x_n.(e_1, \dots, e_n)}\ e_1, \dots, e_n \text{ are destructive over } \{x_1, \dots, x_n\} \subseteq X$$

$\times$ and $+$ are regarded as associative operators. $\times$ binds stronger than $+$, $+$ stronger than $\to$.

var(e) denotes the set of free type variables of e. Given $e, e' \in \mathbb{T}_S(X)$, expressions of the form $e \to e'$ are called **(first-order) functional types** over S and X. $\mathbb{F}_S(X)$ denotes the set of functional types over S. A (functional) type is **ground** or **monomorphic** if it does not contain type variables. The set of ground (functional) types over S is denoted by $\mathbb{T}_S$ and $\mathbb{F}_S$, respectively.

e is a **subtype of** e' if $e = e'$ or there are types $e_1, \ldots, e_n, s$ such that

- $e = e_1 \times \cdots \times e_n$ and e_i is a subtype of e', or
- $e = e_i$ and $e_1 + \cdots + e_n$ is a subtype of e', or
- $e = s \to e_1$, $e' = s \to e_2$ and e_1 is a subtype of e_2. ❏

If there is $s \in \mathbb{T}(S)$ such that $s_i = s$ for all $1 \leq i \leq n$, $e_1 \times \cdots \times e_n$ is also written as s^n.

Sum types are particularly useful for specifying partial functions, exceptions and inheritance. In contrast to subsorting approaches [30] sum typing keeps to the syntax and semantics of many-sorted logic. Sum typing is also a way of avoiding the classical theory of complete partial orders (cpos) for modeling recursive functions, similarly to partially-additive semantics [70] that also uses sums terms. The coalgebraic interpretation of λ-definable functions [39] is another way of handling recursive functions without employing cpos.

Definition 3.2 (sorted sets and functions) Let Set be the category of (small) sets and functions between sets. Let S be a set. Then Set^S denotes the **category of S-sorted sets** $A = \{s^A \mid s \in S\}$ **and S-sorted functions** $f : A \to B = \{s^f : s^A \to s^B \mid s \in S\}$.

Given $s \in S$, Δ_s^A denotes the **diagonal** relation on s^A, i.e., $\Delta_s^A = \{(a, a) \mid a \in s^A\}$. An S-sorted set B is an **S-sorted subset** of A, written as $B \subseteq A$, if for all $s \in S$, s^B is a subset of s^A. Consequently, A and B are equal iff A and B are sortwise equal:

$$A = B \quad \Longleftrightarrow_{def} \quad \forall s \in S : s^A = s^B. \text{ ❏}$$

Definition 3.3 (interpretation of types by functors) Each type $e \in \mathbb{T}_S(X)$ and each S-sorted set A yield a functor $F_{e,A} : Set^X \to Set$ that is defined as follows: Let $B, C \in Set^X$ and $f : B \to C$ be an X-sorted function.

- For all $s \in S$, $F_{s,A}(B) = s^A$ and $F_{s,A}(f) = id_{s^A}$.
- $F_{1,A}(B) = \{()\}$ and $F_{1,A}(f) = id_{\{()\}}$.
- For all $x \in X$, $F_{x,A}(B) = x^B$ and $F_{x,A}(f) = x^f$.
- $F_{e_1 \times \cdots \times e_n, A}(B) = F_{e_1,A}(B) \times \ldots \times F_{e_n,A}(B)$ and $F_{e_1 \times \cdots \times e_n, A}(f) = F_{e_1,A}(f) \times \ldots \times F_{e_n,A}(f)$.
- $F_{e_1 + \cdots + e_n, A}(B) = F_{e_1,A}(B) + \cdots + F_{e_n,A}(B)$ and $F_{e_1 + \cdots + e_n, A}(f) = F_{e_1,A}(f) + \cdots + F_{e_n,A}(f)$.
- $F_{s \to e, A}(B) = [s^A \to F_{e,A}(B)]$ and for all $g : s^A \to F_e(B)$, $F_{s \to e,A}(f)(g) = F_{e,A}(f) \circ g : s^A \to F_{e,A}(C)$.
- Let $Y = \{x_1, \ldots, x_n\}$ and $F : Set^Y \to Set^Y$ be defined by $x_i^{F(D)} = F_{e_i,A}(D)$ for all $1 \leq i \leq n$ and $D \in Set^Y$. Then $F_{\mu x_1 \ldots x_n.(e_1,\ldots,e_n),A}(B) = Ini(Alg_F)$, $F_{\mu x_1 \ldots x_n.(e_1,\ldots,e_n),A}(f) = id_{Ini(Alg_F)}$, $F_{\nu x_1 \ldots x_n.(e_1,\ldots,e_n),A}(B) = Fin(Coalg_F)$ and $F_{\nu x_1 \ldots x_n.(e_1,\ldots,e_n),A}(f) = id_{Fin(Coalg_F)}$. ❏

A straightforward structural induction shows that F_e is indeed a functor, i.e., the follwing diagram $(*)$

commutes:

$$
\begin{tikzcd}
\bullet \arrow[r, dash] \arrow[dd, dash, "g\circ f"'] & B \arrow[r, mapsto, "F_{e,A}"] \arrow[d, "f"'] & B' \arrow[r, dash] \arrow[d, "F_{e,A}(f)"] & \bullet \arrow[dd, dash, "F_{e,A}(g\circ f)"] \\
{} \arrow[r, phantom, "="] & C \arrow[r, mapsto, "F_{e,A}"] \arrow[d, "g"'] & C' \arrow[d, "F_{e,A}(g)"] \arrow[r, phantom, "(*)"] & {} \\
\bullet \arrow[r] & D \arrow[r, mapsto, "F_{e,A}"] & D' & \bullet \arrow[l]
\end{tikzcd}
$$

Definition 3.4 (substitution of type variables by types) Let X be a finite set of type variables. A function $\sigma : X \to \mathbb{T}_S(X)$ extends to a function $\sigma^* : \mathbb{T}_S(X) \to \mathbb{T}_S(X)$ as follows:

- For all $x \in X$, $\sigma^*(x) = \sigma(x)$.
- For all $s \in S \cup \{1\}$, $\sigma^*(s) = s$.
- For all $e_1, \dots, e_n \in \mathbb{T}_S(X)$,
  $\sigma^*(e_1 \times \dots \times e_n) = \sigma^*(e_1) \times \ldots \times \sigma^*(e_n)$ and $\sigma^*(e_1 + \dots + e_n) = \sigma^*(e_1) + \dots + \sigma^*(e_n)$.
- For all $s \in S$ and $e \in \mathbb{T}_S(X)$, $\sigma^*(s \to e) = s \to \sigma^*(e)$.
- $\sigma^*(\mu x_1 \dots x_n.(e_1, \dots, e_n) = \mu x_1 \dots x_n.(\tau^*(e_1), \dots, \tau^*(e_n))$ where $\tau : X \to \mathbb{T}_S(X)$ is defined by
  $\tau(x) = \begin{cases} x_i & \text{if } x = x_i \text{ for some } 1 \le i \le n \\ \sigma(x) & \text{otherwise.} \end{cases}$ ❏

Given $e \in \mathbb{T}_S(X)$ and $A \in Set^S$, the functor property of $F_{e,A}$ implies the following

Proposition 3.5 *Let e be a type over S and X, $\sigma : X \to \mathbb{T}_S$, $A \in Set^S$ and $B \in Set^X$. Then*

$$F_{\sigma^*(e),A}(B) = F_{e,A}(\sigma(B))$$

where for all $x \in X$, $x^{\sigma(B)} =_{def} \begin{cases} F_{\sigma(x),A}(B) & \text{if } x \in var(e), \\ x^B & \text{otherwise.} \end{cases}$ ❏

Definition 3.6 (signatures, structures and homomorphisms) Let X be a set of type variables. A **signature** $\Sigma = (S, Op, Rel)$ **over** X consists of a set S of sorts, an S^2-sorted set Op of function symbols and an S^+-sorted set Rel of relation symbols. We write $f : s_1 \to s_2 \in \Sigma$ instead of $f \in (s_1, s_2)^{Op}$ and $r : w \in \Sigma$ instead of $r \in w^{Rel}$. $dom(f) = s_1$ and $ran(f) = s_2$ are called the **domain** resp. **range** of f.

For all $s \in S$, Rel implicitly includes the s**-equality** $=_s$: ss and the s**-membership** $\in_s$: s. A relation is **logical** if it is neither a membership nor an equality.

A signature $\Sigma' = (S', Op', Rel')$ is a **subsignature** of Σ if $S' \subseteq S$, $Op' \subseteq Op$ and $Rel' \subseteq Rel$.

A Σ**-structure** A consists of an S-sorted set, the **carrier** of A, also denoted by A, for all $f : s_1 \to s_2 \in \Sigma$, a function $f^A : s_1^A \to s_2^A$ and for all $r : s \in \Sigma$, a relation $r^A \subseteq s^A$. A is a Σ**-structure with equality** if for all $s \in S$, $=_s^A = \Delta_s^A$. A is a Σ**-structure with membership** if for all $s \in S$, $\in_s^A = s^A$.

Given Σ-structures A and B, an S-sorted function $h : A \to B$ is a Σ**-homomorphism** if for all $f : s_1 \to s_2 \in \Sigma$, $s_2^h \circ f^A = f^B \circ s_1^h$. If, in addition, f is surjective or injective, f is a Σ**-epimorphism** or a Σ**-monomorphism**, respectively. A Σ-homomorphism $h : A \to B$ is a Σ**-isomorphism** if there is a Σ-homomorphism $g : B \to A$ such that $g \circ h = id_A$ and $h \circ g = id_B$. Then we write $A \cong B$ and say that A and B are Σ**-isomorphic**.

$Mod(\Sigma)$ denotes the category of Σ-structures and Σ-homomorphisms. $Mod_{EM}(\Sigma)$ denotes the full subcategory of $Mod(\Sigma)$ whose objects are Σ-structures with equality and membership.[1]

Given a subsignature $\Sigma' = (S', Op', Rel')$ of Σ and a Σ'-structure A, $Mod(\Sigma, A)$ denotes the category of all **Σ-structures over** A, i.e., all Σ-structures B and Σ-homomorphisms h such that for all $s \in S'$, $s^B = s^A$ and $s^h = id_{s^A}$. $Mod_{EM}(\Sigma, A) =_{def} Mod(\Sigma, A) \cap Mod_{EM}(\Sigma)$. ❏

A Σ-homomorphism is a Σ-isomorphism iff it is bijective.

Definition 3.7 *(terms)* Let $\Sigma = (S, Op, Rel)$ be a signature over X. $Id_S(X) =_{def} \{id_e \mid e \in \mathbb{T}_S(X)\}$. The $\mathbb{F}_S(X)$-sorted set $T_\Sigma(X)$ of **Σ-terms over** X is the set of expressions generated by following rules:

$T_\Sigma(X)$

functions of Σ and identities

$$\frac{}{op : s_1 \to s_2}\; op \in Op \qquad \frac{}{id_e : e \to e}\; e \in \mathbb{T}_S(X)$$

product and sum

$$\frac{t_1 : e \to e_1, \dots, t_n : e \to e_n}{\langle t_1, \dots, t_n\rangle : e \to e_1 \times \dots \times e_n} \qquad \frac{t_1 : e_1 \to e, \dots, t_n : e_n \to e}{[t_1, \dots, t_n] : e_1 + \dots + e_n \to e}$$

projections and injections

$$\frac{}{\pi_i : e_1 \times \dots \times e_n \to e_i} \qquad \frac{}{\iota_i : e_i \to e_1 + \dots + e_n} \qquad 1 \le i \le n$$

rooting and branching

$$\frac{t : e \to e_1,\ t_1 : e_1 \to e_2}{t_1 \circ t : e \to e_2} \qquad \frac{t_2 : e \to e_1,\ t : e_1 \to e_2}{t \circ t_2 : e \to e_2} \qquad t \notin Id_S(X), \quad \begin{array}{l} t_1 \in Op \cup \Sigma\iota \cup \Sigma join, \\ t_2 \in Op \cup \Sigma\pi \cup \Sigma fork \end{array}$$

sub- and supertyping

$$\frac{t : e_1 \to e}{t \times e_2 : e_1 \times e_2 \to e} \qquad \frac{t : e \to e_1}{t + e_2 : e \to e_1 + e_2}$$

left and right distribution

$$\frac{}{dist_L : e \times (e_1 + \dots + e_n) \to e \times e_1 + \dots + e \times e_n}$$

$$\frac{}{dist_R : (e_1 + \dots + e_n) \times e \to e_1 \times e + \dots + e_n \times e}$$

abstraction and application

$$\frac{t : e_1 \times \ldots \times e_n \to e}{\lambda i.t : e_1 \times \ldots \times e_{i-1} \times e_{i+1} \ldots \times e_n \to (e_i \to e)} \qquad \frac{}{\$: (s \to e) \times s \to e} \qquad \begin{array}{l} s, e_i \in S, \\ 1 \le i \le n \end{array}$$

fork and join

$$\frac{\varphi : e}{fork(\varphi) : e \to e + e} \qquad \frac{\varphi : e}{join(\varphi) : e \times e \to e} \qquad \varphi \in Form_\Sigma(X) \text{ (see Def. 3.12)}$$

$\Sigma\pi$, $\Sigma\iota$, $\Sigma fork$ and $\Sigma join$ denote the sets of projections, injections, forks and joins, respectively. A term is **ground** or **monomorphic** if it does not contain type variables. The set of ground Σ-terms is denoted by T_Σ. If $\mathbb{T}_S(X)$ is restricted to either products or sums, T_Σ is an **algebraic theory** in the sense of [64, 109, 23, 107].

Given a Σ-term $t : e \to e'$, $dom(t) =_{def} e$ and $ran(t) =_{def} e'$ are called the **domain** resp. **range** of t.

We sometimes omit $\circ$ and write $t\langle t_1, \dots, t_n\rangle$ and $[t_1, \dots, t_n]t$ instead of $t \circ \langle t_1, \dots, t_n\rangle$ and $[t_1, \dots, t_n] \circ t$, respectively. If $dom(t)$ is a binary product, $t\langle u, v\rangle$ is sometimes written as $u\ t\ v$.

[1] A subcategory $\mathcal{D}$ of a category $\mathcal{C}$ is *full* if all $\mathcal{C}$-morphisms between $\mathcal{D}$-objects are also $\mathcal{D}$-morphisms.

Given terms $t_1 : e_1 \to e_1', \ldots, t_n : e_n \to e_n'$, $t_1 \times \cdots \times t_n$ and $t_1 + \cdots + t_n$ denote the product $\langle t_1 \circ \pi_1, \ldots, t_n \circ \pi_n \rangle$ and sum $[\iota_1 \circ t_1, \ldots, \iota_n \circ t_n]$, respectively.

$u \in T_\Sigma(X)$ is a **subterm** of $t \in T_\Sigma(X)$ if $t = u$ or there are $f \in Op$, $t_1, \ldots, t_n \in T_\Sigma(X)$ and $1 \leq i \leq n$ such that

- $t = f \circ \langle t_1, \ldots, t_n \rangle$ and u is a subterm of t_i or
- $t = [t_1, \ldots, t_n] \circ f$, $u = [u_1, \ldots, u_n] \circ f$, u_i is a subterm of t_i and $u_k = t_k$ for all $1 \leq k \leq n$ with $k \neq i$.

$u \in T_\Sigma(X)$ is a **superterm** of $t \in T_\Sigma(X)$ if $t = u$ or there are $f \in F$, $t_1, \ldots, t_n \in T_\Sigma(X)$ and $1 \leq i \leq n$ such that

- $t = [t_1, \ldots, t_n] \circ f$ and u is a superterm of t_i or
- $t = f \circ \langle t_1, \ldots, t_n \rangle$, $u = f \circ \langle u_1, \ldots, u_n \rangle$, u_i is a superterm of t_i and $u_k = t_k$ for all $1 \leq k \leq n$ with $k \neq i$.❏

It is easy to see that for each Σ-term $t : e \to e_1 \times \cdots \times e_n$ there are $t_1, \ldots, t_n \in T_\Sigma(X)$ such that $t = \langle t_1, \ldots, t_n \rangle$, while for each Σ-term $t : e_1 + \cdots + e_n \to e$ there are $t_1, \ldots, t_n \in T_\Sigma(X)$ such that $t = [t_1, \ldots, t_n]$.

Terms built up of products or sums can be visualized as trees growing downwards or upwards, respectively:

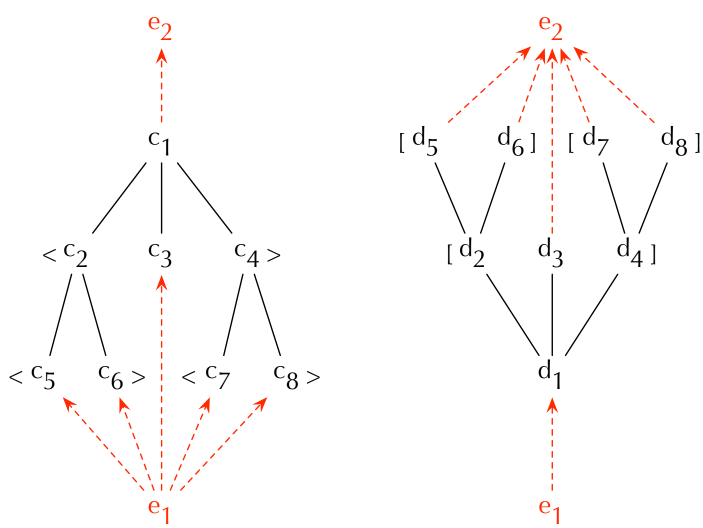


Figure 1. *Tree representations of the terms* $c_1\langle c_2\langle c_5, c_6\rangle, c_3, c_4\langle c_7, c_8\rangle\rangle : e_1 \to e_2$ *and* $[[d_5, d_6]d_2, d_3, [d_7, d_8]d_4]d_1 : e_1 \to e_2$.

As types denote functors (see Def. 3.2), terms denote natural transformations:

Definition 3.8 (interpretation of terms by natural transformations) Each $t : e_1 \to e_2 \in T_\Sigma(X)$ and each $A \in Mod(\Sigma)$ yield a natural transformation $T_{t,A} : F_{e_1,A} \to F_{e_2,A}$, i.e., a set

$$\{T_{t,A}(B) : F_{e_1,A}(B) \to F_{e_2,A}(B) \mid B \in Set^X\}$$

of functions such that for all $B, C \in Set^X$ and X-sorted functions $f : B \to C$, the following diagram commutes:

$$\begin{tikzcd}
F_{e_1,A}(B) \arrow[r, "T_{t,A}(B)"] \arrow[d, "F_{e_1,A}(f)"'] & F_{e_2,A}(B) \arrow[d, "F_{e_2,A}(f)"] \\
F_{e_1,A}(C) \arrow[r, "T_{t,A}(C)"] & F_{e_2,A}(C)
\end{tikzcd}$$

$T_{t,A}$ is defined as follows: Let $B, C \in Set^X$ and $f : B \to C$ be an X-sorted function. The set-theoretical operators on the right-hand sides of the following equations are defined in section 2.

- For all $op : s_1 \to s_2 \in \Sigma$, $T_{op,A}(B) = op^A$.
- For all $e \in \mathbb{T}_S(X)$, $T_{id_e,A}(B) = id_{F_{e,A}(B)}$.
- For all $t_1 : e \to e_1, \dots, t_n : e \to e_n \in T_\Sigma(X)$,
  $T_{\langle e_1,\dots,e_n\rangle,A}(B) = \langle T_{e_1,A}(B), \dots, T_{e_n,A}(B)\rangle$ and $T_{[e_1,\dots,e_n],A}(B) = [T_{e_1,A}(B), \dots, T_{e_n,A}(B)]$.
- For all $e_1, \dots, e_n \in \mathbb{T}_S(X)$ and $1 \le i \le n$, $T_{\pi_i,A}(B) = \pi_i$ and $T_{\iota_i,A}(B) = \iota_i$.
- For all $t_1 : e_1 \to e_2, t_2 : e_2 \to e_3$, $T_{t_2 \circ t_1,A}(B) = T_{t_2,A}(B) \circ T_{t_1,A}(B)$.
- For all $t : e \to e_1$, $e_2 \in \mathbb{T}_S(X)$ and $(c, d) \in F_{e\times e_2,A}(B)$, $T_{t\times e_2,A}(B)(c, d) = T_{t,A}(B)(c)$.
- For all $t : e \to e_1$, $e_2 \in \mathbb{T}_S(X)$ and $c \in F_{e,A}(B)$, $T_{t+e_2,A}(B)(c) = T_{t,A}(B)(c)$.
- For all $e, e_1, \dots, e_n \in \mathbb{T}_S(X)$, $1 \le i \le n$ and $(c, \iota_i(d)) \in F_{e\times(e_1+\cdots+e_n),A}(B)$, $T_{dist_L,A}(B)(c, \iota_i(d)) = \iota_i(c, d)$.
- For all $e, e_1, \dots, e_n \in \mathbb{T}_S(X)$, $1 \le i \le n$ and $(\iota_i(c), d) \in F_{e\times(e_1+\cdots+e_n),A}(B)$, $T_{dist_R,A}(B)(\iota_i(c), d) = \iota_i(c, d)$.
- For all $t : e_1 \times \ldots \times e_n \to e$ with $e_i \in S$, $f : e_i^A \to (F_{e_1\times\ldots\times e_{i-1}\times e_{i+1}\ldots\times e_n,A}(B) \to F_{e,A}(B))$ and $(c_1, \dots, c_n) \in F_{e_1\times\ldots\times e_n,A}(B)$, $T_{\lambda i.t,A}(B)(c_1, \dots, c_{i-1}, c_{i+1}, \dots, c_n)(c_i) = T_{t,A}(B)(c_1, \dots, c_n)$.
- For all $s \in S$, $e \in \mathbb{T}_S(X)$, $f : s^A \to F_{e,A}(B)$ and $a \in s^A$, $T_{\$,A}(B)(f, a) = f(a)$.
- For all $\varphi : e \in Form_\Sigma$ and $a \in F_{e,A}(B)$, $T_{fork(\varphi),A}(B)(a) = \begin{cases} (a,1) & \text{if } a \in \varphi^A, \\ (a,2) & \text{if } a \notin \varphi^A. \end{cases}$
- For all $\varphi : e \in Form_\Sigma$ and $(a, b) \in F_{e\times e,A}(B)$, $T_{join(\varphi),A}(B)(a, b) = \begin{cases} a & \text{if } a \in \varphi^A, \\ b & \text{if } a \notin \varphi^A. \end{cases}$ ❏

Definition 3.9 Let $\Sigma = (S, Op, Rel)$ be a signature over X and $Y = \{x_1, \dots, x_n\}$.

For all $1 \le i \le n$, let $e_i = \coprod_{j=1}^{k_i} e_{ij}$ be a constructive type over Y and $C_i = \{\iota_{ij} : e_{ij} \to e_i \mid 1 \le j \le k_i\}$ be the set of injections into e_i. Then the signature

$$\Sigma' \;=\; (S \cup Y, Op \cup C_1 \cup \cdots \cup C_n, Rel)$$

over $X \setminus Y$ is called a **constructor signature** with base signature Σ.

For all $1 \le i \le n$, let $e_i = \prod_{j=1}^{k_i} e_{ij}$ be a destructive type over Y and $D_i = \{\pi_{ij} : e_i \to e_{ij} \mid 1 \le j \le k_i\}$ be the set of projections from e_i. Then the signature

$$\Sigma' \;=\; (S \cup Y, Op \cup D_1 \cup \cdots \cup D_n, Rel)$$

over $X \setminus Y$ is called a **destructor signature** with base signature Σ. ❏

???? Σ is **algebraic** if for all $f : e \to e' \in Op$ and $r : e \in Rel$, e and e' are products of sorts.

Definition 3.10 (term composition) Let $\Sigma = (S, Op, Rel)$ be a signature over X. The **term category** $\mathcal{T}_\Sigma(X)$ has all types over S and X as objects and all Σ-terms over X as morphisms. The composition $\odot$ of $\mathcal{T}_\Sigma(X)$-morphisms is defined inductively on the structure of Σ-terms:

- For all $t : e_1 \to e_2 \in T_\Sigma(X)$, $id_{e_2} \odot t = t \odot id_{e_1} = t$.
- For all $\langle t_1, \dots, t_n\rangle \in T_\Sigma(X)$ and $1 \le i \le n$, $\pi_i \odot \langle t_1, \dots, t_n\rangle = t_i$.
- For all $[t_1, \dots, t_n] \in T_\Sigma(X)$ and $1 \le i \le n$, $[t_1, \dots, t_n] \odot \iota_i = t_i$.
- For all $t : e \to e_1 \in T_\Sigma(X) \setminus Id_S(X)$ and $t_1 : e_1 \to e_2 \in Op \cup \Sigma\iota \cup \Sigma join$, $t_1 \odot t = t_1 \circ t$.
- For all $t_2 : e \to e_1 \in Op \cup \Sigma\pi \cup \Sigma fork$ and $t : e_1 \to e_2 \in T_\Sigma(X) \setminus Id_S(X)$, $t \odot t_2 = t \circ t_2$.
- For all $t : e_1 \to e \in T_\Sigma(X)$, $e_2 \in \mathbb{T}_S(X)$ and $t' : e \to e_3 \in T_\Sigma(X)$, $t' \odot (t \times e_2) = (t' \odot t) \times e_2$.

- For all $t : e \to e_1 \in T_\Sigma(X)$, $e_2 \in \mathbb{T}_S(X)$ and $t' : e_3 \to e \in T_\Sigma$, $(t + e_2) \odot t' = (t \odot t') + e_2$.
- For all $t : e \to e_1 \in T_\Sigma(X)$ and $\langle t_1, \dots, t_n\rangle : e_1 \to e_2 \in T_\Sigma(X)$, $\langle t_1, \dots, t_n\rangle \odot t = \langle t_1 \odot t, \dots, t_n \odot t\rangle$.
- For all $[t_1, \dots, t_n] : e \to e_1 \in T_\Sigma(X)$ and $t : e_1 \to e_2 \in T_\Sigma(X)$, $t \odot [t_1, \dots, t_n] = [t \odot t_1, \dots, t \odot t_n]$.
  ****
- For all $t : s \to s' \in T_\Sigma$ and $\varphi : s' \in \mathit{Form}_\Sigma$, $\mathit{fork}(\varphi) \odot t = (t + t) \odot \mathit{fork}(\varphi \odot t)$ (see Def. 3.12).
- For all $\varphi : s \in \mathit{Form}_\Sigma$ and $t : s \to s' \in T_\Sigma$, $t \odot \mathit{join}(\varphi) = \mathit{join}(t \odot \varphi) \odot (t \times t)$ (see Def. 3.12).

For all $t : \prod_{i\in I} s_i \to s \in T_\Sigma$, $k \in I$ and $u : s' \to s_k \in T_\Sigma$, $t \odot_k u =_{def} t \odot \langle u_i\rangle_{i\in I} : \prod_{i\in I} s'_i \to s$ where

$$u_i = \begin{cases} u \odot \pi_i & \text{if } i = k \\ \pi_i & \text{otherwise} \end{cases} \qquad s'_i = \begin{cases} s' & \text{if } i = k \\ s_i & \text{otherwise} \end{cases}$$

For all $t : s \to \coprod_{i\in I} s_i \in T_\Sigma$, $k \in I$ and $u : s_k \to s' \in T_\Sigma$, $u \odot^k t =_{def} [u_i]_{i\in I} \odot t : s \to \coprod_{i\in I} s'_i$ where

$$u_i = \begin{cases} \iota_i \odot u & \text{if } i = k \\ \iota_i & \text{otherwise} \end{cases} \qquad s'_i = \begin{cases} s' & \text{if } i = k \\ s_i & \text{otherwise} \end{cases} \quad ❏$$

Since the indices of projections replace the logical variables used in applicative logical syntax, we sometimes write i instead of π_i. Conversely, a term t in applicative syntax can be turned into a (purely functional) Σ-term by regarding the set X of variables occurring in t as a set of indices and replacing each $x : s_x \in X$ with the projection $\pi_x : \prod_{i\in I} s_i \to s_x$. For instance, let $I = \{x : s_x, y : s_y, z : s_z\}$. Then the applicative term $f(x, g(y, x), z) : s$ becomes the Σ-term

$$f \circ \langle \pi_x, g \circ \langle \pi_y, \pi_x\rangle, \pi_z\rangle \;:\; s_x \times s_y \times s_z \to s.$$

T_Σ covers both applicative terms and substitutions into such terms. For instance, the substitution $\sigma = \{t_x/x, t_y/y, t_z/z\}$ that maps variables x, y, z to terms $t : \mathit{dom}_x \to s_x, u : \mathit{dom}_y \to s_y, v : \mathit{dom}_z \to s_z$, respectively, represents the Σ-term $\vec{t} = t \times u \times v$. Hence the instance $f(t, g(u, t), v)$ of $f(x, g(y, x), z)$ by σ becomes the composition $f(x, g(y, x), z) \odot \vec{t}$ that represents a single term of type $\mathit{dom}_x \times \mathit{dom}_y \times \mathit{dom}_z \to s$:

$$\begin{aligned} f(x, g(y, x), z) \odot \vec{t} &= (f \circ \langle \pi_x, g \circ \langle \pi_y, \pi_x), \pi_z\rangle) \odot \vec{t} = f \circ \langle \pi_x \odot \vec{t}, g \circ \langle \pi_y, \pi_x\rangle \odot \vec{t}, \pi_z \odot \vec{t}\rangle \\ &= f(t, g \circ \langle \pi_y \circ \vec{t}, \pi_x \circ \vec{t}\rangle, v) = f(t, g \circ \langle u, t\rangle, v) = f(t, g(u, t), v). \end{aligned}$$

To sum up this translation, let T_s be the set of applicative terms of sort s and $T = \{T_s\}_{s\in S}$. The following function $\mathit{comp} : T \to T_\Sigma$ turns each applicative term t with variables $x_1 : s_1, \dots, x_m : s_m$ and constants $c_1 : s'_1, \dots, c_n : s'_n$ into a Σ-term $\mathit{comp}(t) : s_1 \times \dots \times s_n \times 1 \to s$:

- For all $1 \le i \le m$, $\mathit{comp}(x_i) = \pi_i$.
- For all $1 \le i \le n$, $\mathit{comp}(c_i) = c_i \circ \pi_{n+1}$.
- For all $k > 0$, $1 \le i \le k$, functions $f : s_1 \times \dots \times s_k \to s \in \Sigma$ and $t_i \in T_{s_i}$,
  $\mathit{comp}(f(t_1, \dots, t_k)) = f \circ \langle \mathit{comp}(t_1), \dots, \mathit{comp}(t_k)\rangle$.

Definition 3.11 *(substitution of sorts by terms)* Let $\Sigma = (S_0, S, F, R)$ be a signature. A function $\mathit{sub} : S \to T_\Sigma$ extends to a function $\mathit{sub}^* : \mathbb{T}_S \to T_\Sigma$ as follows:

- For all $s \in S$, $\mathit{sub}^*(s) = \mathit{sub}(s)$.
- For all $\{s_i\}_{i\in I} \subseteq \mathbb{T}_S$, $\mathit{sub}^*(\prod_{i\in I} s_i) = \prod_{i\in I}(\mathit{sub}^*(s_i))$ and $\mathit{sub}^*(\coprod_{i\in I} s_i) = \coprod_{i\in I}(\mathit{sub}^*(s_i))$.

As terms represent functions, formulas represent relations:

Definition 3.12 *(formulas)* Let $\Sigma = (S_0, S, F, R)$ be a signature and I, J be nonempty sets. The $\mathbb{T}_S$-sorted set $Form_\Sigma$ of **Σ-formulas** is the least set of typed expressions φ generated by the following rules:

relations of Σ and truth values

$$\frac{}{r : s}\quad r : s \in R \qquad\qquad \frac{}{True : \coprod_{s\in S} s} \qquad\qquad \frac{}{False : \coprod_{s\in S} s}$$

Σ-atoms and -coatoms

$$\frac{t : s \to s'}{r \circ t : s} \qquad \frac{t : s' \to s}{t \circ r : s} \qquad r : s' \in R,\ t \in T_\Sigma \setminus \{id_s\}$$

negation, conjunction and disjunction

$$\frac{\varphi : s}{\neg\varphi : s} \qquad \frac{\varphi : s,\quad \psi : s}{\varphi \wedge \psi : s} \qquad \frac{\varphi : s,\quad \psi : s}{\varphi \vee \psi : s}$$

quantification

$$\frac{\varphi : \prod_{i\in I} s_i}{\forall k\varphi : \prod_{i\in I} s_i} \qquad \frac{\varphi : \prod_{i\in I} s_i}{\exists k\varphi : \prod_{i\in I} s_i} \qquad k \in I$$

sub- and supertyping

$$\frac{\varphi : s_1}{\varphi \times s_2 : s_1 \times s_2} \qquad \frac{\varphi : s_1}{\varphi + s_2 : s_1 + s_2}$$

We sometimes omit $\circ$ and write $r\langle t_1, \dots, t_n\rangle$ and $[t_1, \dots, t_n]q$ instead of $r \circ \langle t_1, \dots, t_n\rangle$ and $[t_1, \dots, t_n] \circ q$, respectively. If dom_r is a binary product, $r\langle t, u\rangle$ may be written as $t\ r\ u$. Moreover, $\varphi \Rightarrow \psi$ and $\psi \Leftarrow \varphi$ stand for $\neg\varphi \vee \psi$ and $\varphi \Leftrightarrow \psi$ stands for $(\varphi \Rightarrow \psi) \wedge (\psi \Rightarrow \varphi)$.

Given a Σ-formula $\varphi : s$, $dom_\varphi =_{def} s$ is called the **domain** of φ. φ is a **ground formula** if $s = 1$.

The component formulas φ_i of a conjunction $\varphi = \varphi_1 \wedge \dots \wedge \varphi_n$ or disjunction $\psi = \varphi_1 \vee \dots \vee \varphi_n$ are called **factors** resp. **summands** of φ.

Given a Σ-atom p, p and $\neg p$ are called a **Σ-literals**. If r is logical, then p is **logical**. A Σ-formula $\equiv \circ\langle t, u\rangle$ is called a **Σ-equation** and usually written as $t \equiv u$. A Σ-formula $all \circ t$ is called a **Σ-membership** and usually written as $all(t)$.

Given a Σ-atom $p : s$ and a Σ-formula $\varphi : s$, $p \Leftarrow \varphi$ and $p \Rightarrow \varphi$ are called a **Horn** resp. **co-Horn clause over** Σ.[2] Given terms $t : s$ and $u : s$, a Horn clause $t \equiv u \Leftarrow \varphi$ is also called a **conditional equation**.

A Σ-formula φ is **normalized** if φ consists of literals, quantifiers and conjunction or disjunction symbols. Given a set R' of relations, a normalized Σ-formula φ is **R-positive** if for all literals $\neg r \circ t$ of φ, $r \notin R$. A Horn clause $p \Leftarrow \varphi$ and a co-Horn clause $p \Rightarrow \varphi$ is **R-positive** if φ is R-positive.

$r \circ t \Leftarrow \varphi$ or $r \circ t \Rightarrow \varphi$ is called a Horn resp. co-Horn clause **for** r and all sets of relations that include r. Given a constructur f, $f \circ t \equiv u \Leftarrow \varphi$ is called a Horn clause **for** f and all sets of functions that include f. Given a destructur f, $t \circ f \equiv u \Leftarrow \varphi$ is called a Horn clause **for** f and all sets of functions that include f.

A Σ-formula φ is **restricted** if

- for all subformulas $\forall k\psi$ of φ, $s_k \in S_0$ or $\neg all_{s_k} \circ \pi_k$ is a summand of ψ,
- for all subformulas $\exists k\psi$ of φ, $s_k \in S_0$ or $all_{s_k} \circ \pi_k$ is a factor of ψ.

[2]Since φ is not confined to finite conjunctions of atoms, our notion of a Horn clause deviates from the classical one. It also does not coincide with the notion of a *hereditary Harrop formula* [80]. Premises with universal quantifiers, which do not occur in classical Horn clauses, are allowed both in Harrop formulas and in Horn clauses as defined here. But Harrop formulas impose further restrictions on their premises.

A Σ-formula φ is **implicational** if φ can be constructed by applying the above rules except for negation, disjunction and existential quantification and by the following additional rule:

simple implication	
$\dfrac{\psi : s, \quad \varphi : s}{\psi \Rightarrow \varphi : s}$	ψ is a conjunction of universally quantified atoms

❏

A Horn clause $p \Leftarrow \mathit{True}_s$ is identified with p. A co-Horn clause $p \Rightarrow \mathit{False}_s$ is identified with $\neg p$.

Definition 3.13 *(term rewriting)* Let $\Sigma = (S_0, S, F, R)$ be a signature and E be a set of Σ-equations. $\longrightarrow_E$ denotes the least binary relation on T_Σ such that

- for all $t \equiv u \in E$, $t \longrightarrow_E u$,
- for all and $v : \prod_{i\in I} s_i \to s \in T_\Sigma$, $k \in I$ and $t, u : s' \to s_k \in T_\Sigma$, $t \longrightarrow_E u$ implies $v \odot_k t \longrightarrow_E v \odot_k u$.
- for all and $t, u : \prod_{i\in I} s_i \to s \in T_\Sigma$, $k \in I$ and $v : s' \to s_k \in T_\Sigma$, $t \longrightarrow_E u$ implies $t \odot_k v \longrightarrow_E u \odot_k v$.
- for all and $v : s \to \coprod_{i\in I} s_i \in T_\Sigma$, $k \in I$ and $t, u : s_k \to s' \in T_\Sigma$, $t \longrightarrow_E u$ implies $t \odot^k v \longrightarrow_E u \odot^k v$.
- for all and $t, u : s \to \coprod_{i\in I} s_i \in T_\Sigma$, $k \in I$ and $v : s_k \to s \in T_\Sigma$, $t \longrightarrow_E u$ implies $v \odot^k t \longrightarrow_E v \odot^k u$.

$\stackrel{*}{\longrightarrow}_E$ and $\stackrel{*}{\longleftrightarrow}_E$ denote the reflexive-transitive resp. equivalence closure of $\longrightarrow_E$. ❏

Definition 3.14 *(signature morphism)* Let $\Sigma = (S_0, S, F, R)$ and $\Sigma' = (S_0', S', F', R')$ be signatures. A **signature morphism** $\sigma : \Sigma \to \Sigma'$ consists of a function from S to $\mathbb{T}_{S'}$ and S-sorted functions $\{\sigma_s : F_s \to T_{\Sigma,\sigma(s)}\}_{s\in S}$ and $\{\sigma_s : R_s \to \mathit{Form}_{\Sigma,\sigma(s)}\}_{s\in S}$. The **domain of** σ, $dom(\sigma)$, is the set of symbols of Σ such that $\sigma(s) \neq s$.

It is obvious how σ extends to a function from $\mathbb{T}_S$ to $\mathbb{T}_{S'}$ and $\mathbb{T}_S$-sorted functions $\{\sigma_s : T_{\Sigma,s} \to T_{\Sigma,\sigma(s)}\}_{s\in\mathbb{T}_S}$ and $\{\sigma_s : \mathit{Form}_{\Sigma,s} \to \mathit{Form}_{\Sigma,\sigma(s)}\}_{s\in\mathbb{T}_S}$. Given a Σ-term or -formula t, $\sigma(t)$ is also witten as $t[\sigma(s)/s | s \in dom(\sigma)]$.

For all $s \in S \cup F$, $\sigma^*(s) =_{def} \sigma(s)$ and for all $r \in R$, $\sigma^*(r) =_{def} \bigvee_{i\in\mathbb{N}} \sigma^i(r)$. ❏

Definition 3.15 *(formula-term composition)* Let the assumptions of Def. 3.12 hold true. Formula-term composition is a function $\odot : \mathit{Form}_\Sigma \times T_\Sigma \to \mathit{Form}_\Sigma$ that extends the composition of morphisms in $\mathcal{T}_S$ (see Def. 3.10) inductively on the structure of Σ-formulas:

- For all $t : s \to s' \in T_\Sigma \setminus \{id_s\}$ and $r : s' \in R$, $r \odot t = r \circ t$.
- For all $r : s' \in R$ and $t : s' \to s \in T_\Sigma \setminus \{id_s\}$, $t \odot r = t \circ r$.
- For all $t : s \to s', t' : s' \to s'' \in T_\Sigma$ and $\varphi : s'' \in \mathit{Form}_\Sigma$, $(\varphi \circ t) \odot t' = \varphi \odot (t \odot t')$.
- For all $\varphi : s \in \mathit{Form}_\Sigma$ and $t : s \to s', t' : s' \to s'' \in T_\Sigma$, $t' \odot (t \odot \varphi) = (t' \odot t) \odot \varphi$.
- For all $\varphi : s \in \mathit{Form}_\Sigma$ and $t : s' \to s \in T_\Sigma$, $(\neg\varphi) \odot t = \neg(\varphi \odot t)$.
- For all $\varphi : s, \psi : s \in \mathit{Form}_\Sigma$ and $t : s' \to s \in T_\Sigma$,
  $(\varphi \wedge \psi) \odot t = (\varphi \odot t) \wedge (\psi \odot t)$ and $(\varphi \vee \psi) \odot t = (\varphi \odot t) \vee (\psi \odot t)$.
- For all $\varphi : s, \psi : s \in \mathit{Form}_\Sigma$ and $t : s \to s' \in T_\Sigma$,
  $t \odot (\varphi \wedge \psi) = (t \odot \varphi) \wedge (t \odot \psi)$ and $t \odot (\varphi \vee \psi) = (t \odot \varphi) \vee (t \odot \psi)$.
- For all $t : s \to s' \in T_\Sigma$, $\varphi : s' \in \mathit{Form}_\Sigma$ and $k \in I$ such that π_k does not occur in t,
  $(\forall k\varphi) \odot t = \forall k(\varphi \odot t)$ and $(\exists k\varphi) \odot t = \exists k(\varphi \odot t)$.
- For all $\varphi : s \in \mathit{Form}_\Sigma$, $t : s \to s' \in T_\Sigma$ and $k \in I$ such that ι_k does not occur in t,
  $t \odot \forall k\varphi = \forall k(t \odot \varphi)$ and $t \odot \exists k\varphi = \exists k(t \odot \varphi)$.
- For all $t : s_1 \to s \in T_\Sigma$, $s_2 \in \mathbb{T}_S$ and $\varphi : s \in \mathit{Form}_\Sigma$, $\varphi \odot t \times s_2 = (\varphi \odot t) \times s_2$.
- For all $t : s \to s_1 \in T_\Sigma$, $s_2 \in \mathbb{T}_S$ and $\varphi : s \in \mathit{Form}_\Sigma$, $t + s_2 \odot \varphi = (t \odot \varphi) + s_2$.

For all $\varphi : \prod_{i\in I} s_i \in Form_\Sigma$, $k \in I$ and $t : s \to s_k \in T_\Sigma$, $\varphi \odot_k t =_{def} \varphi \odot \langle t_i\rangle_{i\in I} : \prod_{i\in I} s'_i \to s$ where

$$t_i = \begin{cases} t \odot \pi_i & \text{if } i = k \\ \pi_i & \text{otherwise} \end{cases} \qquad s'_i = \begin{cases} s & \text{if } i = k \\ s_i & \text{otherwise} \end{cases}$$

For all $\varphi : \coprod_{i\in I} s_i \in Form_\Sigma$, $k \in I$ and $t : s_k \to s \in T_\Sigma$, $t \odot^k \varphi =_{def} [t_i]_{i\in I} \odot \varphi : \coprod_{i\in I} s'_i$ where

$$t_i = \begin{cases} \iota_i \odot t & \text{if } i = k \\ \iota_i & \text{otherwise} \end{cases} \qquad s'_i = \begin{cases} s & \text{if } i = k \\ s_i & \text{otherwise} \end{cases}$$

For all $\varphi : \prod_{i\in I} s_i \in Form_\Sigma$, $k \in I$ and $t : s \to s_k \in T_\Sigma$, $\varphi \odot_k t =_{def} \varphi \odot (t \odot_k id_s)$.
For all $\varphi : \coprod_{i\in I} s_i \in Form_\Sigma$, $k \in I$ and $t : s_k \to s \in T_\Sigma$, $t \odot^k \varphi =_{def} (t \odot^k id) \odot \varphi$. ❑

For all $s \in S$, let F be the set of usual first-order formulas over Σ. The function $comp : T \to T_\Sigma$ given in section 2 extends to a function $comp : F \to Form_\Sigma$ that turns each first-order formula φ with free variables $x_1 : s_1, \dots, x_n : s_n$ into a Σ-formula $comp(\varphi) : s_1 \times \cdots \times s_n \times 1$:

- $comp(True) = True$ and $comp(False) = False$.
- For all $k > 0$, $1 \le i \le k$, relations $r : s_1 \times \cdots \times s_k \in \Sigma$ and $t_i \in T_{s_i}$, $comp(r(t_1, \dots, t_k)) = r \circ \langle comp(t_1), \dots, comp(t_k)\rangle$.
- For all $\varphi \in F$, $comp(\neg\varphi) = \neg comp(\varphi)$.
- For all $\varphi, \psi \in F$, $comp(\varphi \wedge \psi) = comp(\varphi) \wedge comp(\psi)$.
- For all $\varphi \in F$ and $1 \le i \le n$, $comp(\forall x_i \varphi) = \forall i\ comp(\varphi)$.

4 Semantics of specifications

Definition 4.1 *(generator, observer, complete axiomatization)* Let $\Sigma = (S_0, S, F, R)$ be a signature and $S_1 = S \setminus S_0$. The S-sorted sets Gen_Σ of **Σ-generators** and Obs_Σ of **Σ-observers** are defined as follows (see Def. 3.7):

- For all $s \in S_0$, $Gen_{\Sigma,s} = Obs_{\Sigma,s} = \{id_s\}$.
- For all $s \in S_1$, $Gen_{\Sigma,s}$ consists of all Σ-terms $t : dom \to s$ built up of S_1-constructors and **variables**, i.e., Σ-projections occurring in leaves of the tree representation of t.
- For all $s \in S_1$, $Obs_{\Sigma,s}$ consists of all Σ-terms $t : s \to ran$ built up of S_1-destructors and **covariables**, i.e., Σ-injections that occur only in leaves of the tree representation of t.

$t \in Gen_\Sigma$ is a **maximal generator** if $dom_t \in \mathbb{T}_{S_0}$. $t \in Obs_\Sigma$ is a **maximal observer** if $ran_t \in \mathbb{T}_{S_0}$. $MGen_\Sigma$ and $MObs_\Sigma$ denote the S-sorted sets of maximal Σ-generators and Σ-observers, respectively. ❑

Since generators do not involve sums $[t_i]_{i\in I}$ of terms and observers do not involve products $\langle t_i\rangle_{i\in I}$ of terms, the domain of each subterm of a generator agrees with generator's domain, while the range of each superterm of an observer agrees with the observer's range. Still, generators may involve sum types as observers may involve product types. For instance, some proper (!) subtype of a constructor's domains may be a sum, or some proper subtype of a destructor's range may be a product—and often has to be (see Example 15.2).

As products of terms are crucial for building up generators, so are sums of terms for building up observers. In many previous papers on coalgebraic specification, sums do not play the prominent rôle they seem to have here. The simple reason is that the sample signatures used in those papers lack destructors with sum ranges. However, in practice, such destructors emerge as quickly as constructors with product domains do (see, e.g., Examples 14.2 and 15.2). Only Corina Cîrstea [15] pays the special attention to sums of terms that they deserve.

Her notion of a **coterm** almost agrees with our notion of an observer. As we compile variables of applicative terms into projections, so, dually, Cîrstea calls the injections at the leaves of observers **covariables**.

CoCASL [83] does not allow sums in the range of destructors. For instance, CoCASL would replace the destructor $ht : clist(s) \to 1 + s \times clist(s)$ of COLIST (see Example 15.2) by two partial (!) destructors $head : clist(s) \to s$ and $tail : clist(s) \to clist(s)$ (see [83], Fig. 4). However, it is well-known that the involvement of partial functions makes most algebraic and logic reasoning very complicated. Sum types are a better choice: they cover partiality *and* stay within usual algebraic frameworks.

Since generators lack sums and observers lack products, these two kinds of terms admit a simple graphical representation (see Figs. 1 and 2). Signatures all of whose functions involving non-primitive sorts are either constructors or destructors admit initial resp. final models (see Section 5). They are called algebraic resp. coalgebraic. The term "dialgebraic" is usually associated with functions that are neither constructors nor destructors, i.e., both their domains and their ranges are complex types. They do not initial or final models (see [99]) and thus cannot be treated inductively or coinductively. Besides this proof-theoretical drawback the question is whether purely dialgebraic functions are needed in a framework for specifying data types. The motivating example of [99] is hardly convincing: the function *split* that takes a list L of numbers and a number n and returns the lists of elements of L that are smaller resp. greater than n framework is simply the product $\langle f, g\rangle$ of a function f that returns the smaller elements and a function g that returns the greater elements. Dually, a function of type $\coprod_{i\in I} s_i \to s$ is the sum of functions $f_i : s_i \to s$, $i \in I$. The third kind of purely dialgebraic functions has types of the form $\prod_{i\in I} s_i \to \coprod_{i\in I} s'_i$. To make them algebraic (or coalgebraic) one may introduce a new sort s for $\coprod_{i\in I} s_i$ (or $\prod_{i\in I} s'_i$) with the injections ι_i, $i \in I$, as constructors or the projections π_i, $i \in I$, as destructors.

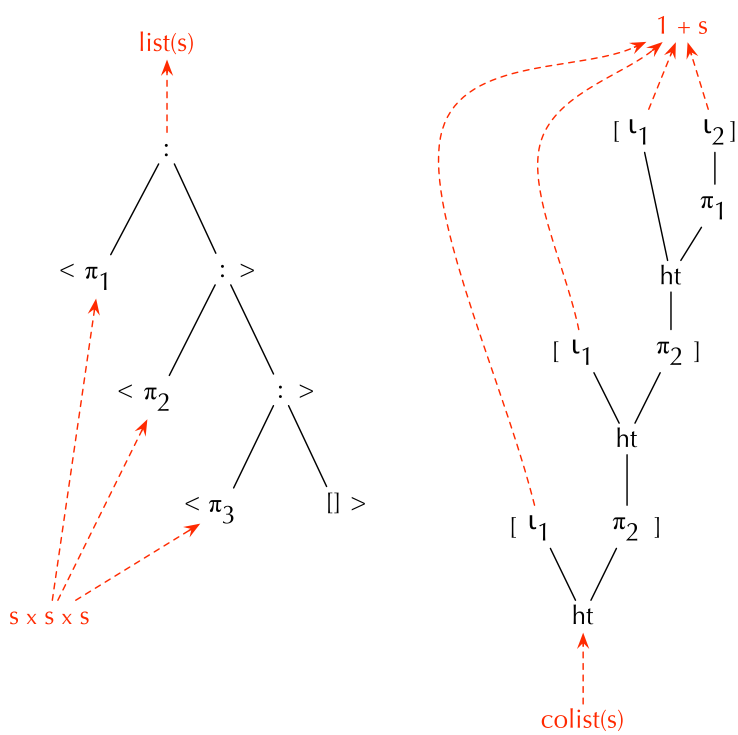


Figure 2. *The LIST-generator* $: \circ\langle\pi_1, : \circ\langle\pi_2 : \circ\langle\pi_3, [] \circ \pi_4\rangle\rangle\rangle : s \times s \times s \times 1 \to list(s)$ *that represents lists with exactly three elements (see Example 14.2) and the COLIST-observer* $[\iota_1, [\iota_1, [\iota_1, \iota_2 \circ \pi_1] \circ ht \circ \pi_2] \circ ht \circ \pi_2] \circ ht : clist(s) \to 1 + s$ *that, given a colist L, returns 1 if $|L| < 3$ and the third element of L otherwise (see Example 15.2).*

Proposition 4.2 *Let $\Sigma = (S_0, S, F, R)$ be a signature, A be an S_0-sorted set and B be an S-sorted set such that $A_s = B_s$ for all $s \in S_0$.*

- *If for all $s \in S$,*
$$s^B = \coprod_{t:dom \to s \in Gen_\Sigma} dom^A, \tag{1}$$
then (1) holds true for all $s \in \mathbb{T}_S$ as well.
- *If for all $s \in S$,*
$$s^B = \prod_{t:s \to ran \in Obs_\Sigma} ran^A, \tag{2}$$
then (2) holds true for all $s \in \mathbb{T}_S$ as well.

Proof. Let $\{s_i \mid i \in I\} \subseteq \mathbb{T}_S$.

Suppose that (1) holds true for all $s \in S$. Then by induction on the structure of $s = \prod_{i\in I} s_i$,

$$s^B = \prod_{i\in I} s_i^B = \prod_{i\in I} \coprod_{t:dom\to s_i\in Gen_\Sigma} dom^A = \coprod_{\{t_i:dom_i\to s_i\in Gen_\Sigma \mid i\in I\}} \prod_{i\in I} dom_i^A = \coprod_{\prod_{i\in I} t_i:(\prod_{i\in I} dom_i)\to s\in Gen_\Sigma} (\prod_{i\in I} dom_i)^A = \coprod_{t:dom\to s\in Gen_\Sigma} dom^A.$$

Moreover, by induction on the structure of $s = \coprod_{i\in I} s_i$,

$$s^B = \coprod_{i\in I} s_i^B = \coprod_{i\in I} \coprod_{t:dom\to s_i\in Gen_\Sigma} dom^A = \coprod_{i\in I} \coprod_{\iota_i\circ t:dom\to s\in Gen_\Sigma} dom^A = \coprod_{\{\iota_i\circ t:dom\to s\in Gen_\Sigma \mid i\in I\}} dom^A = \coprod_{t:dom\to s\in Gen_\Sigma} dom^A.$$

Suppose that (2) holds true for all $s \in S$. Then by induction on the structure of $s = \prod_{i\in I} s_i$,

$$(s^B = \prod_{i\in I} s_i^B = \prod_{i\in I} \prod_{t:s_i\to ran\in Obs_\Sigma} ran^A = \prod_{i\in I} \prod_{t\circ\pi_i:s\to ran\in Obs_\Sigma} ran^A = \prod_{\{t\circ\pi_i:s\to ran\in Obs_\Sigma \mid i\in I\}} ran^A = \prod_{t:s\to ran\in Obs_\Sigma} ran^A.$$

Moreover, by induction on the structure of $s = \coprod_{i\in I} s_i$,

$$s^B = \coprod_{i\in I} s_i^B = \coprod_{i\in I} \prod_{t:s_i\to ran\in Obs_\Sigma} ran^A = \prod_{\{t_i:s_i\to ran_i\in Obs_\Sigma \mid i\in I\}} \coprod_{i\in I} ran_i^A = \prod_{\coprod_{i\in I} t_i:s\to\coprod_{i\in I} ran_i\in Obs_\Sigma} (\coprod_{i\in I} ran_i)^A = \prod_{t:s\to ran\in Obs_\Sigma} ran^A.$$ ❏

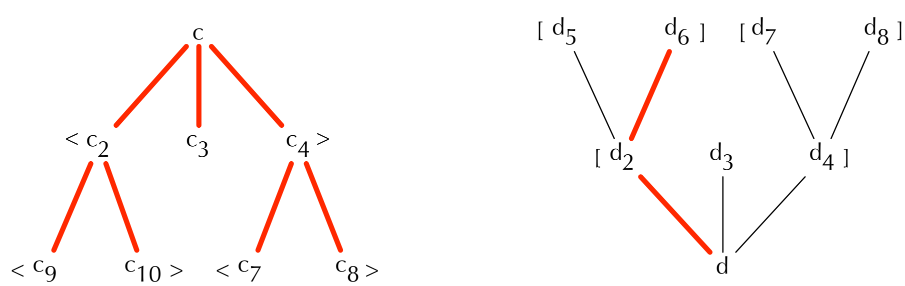


Figure 3. *Thick edges in the term resp. coterm of Fig. 1 denote possible flows of data when it is evaluated. In a term, each function (= node label)* **collects** *its arguments from a product domain. In a coterm, each function* **selects** *a summand of a sum range where it passes the resulting value to.*

Definition 4.3 *(interpretation of terms and formulas; models)* Let $\Sigma = (S_0, S, F, R)$ be a signature and A be a Σ-structure. The interpretation of a Σ-term $t : s_1 \to s_2$ in A is a function $t^A : s_1^A \to s_2^A$ whose definition extends the interpretation of F in A inductively on the structure of Σ-terms (see Def. 3.7):

The interpretation of a Σ-formula $\varphi : s$ in A is a subset of s^A that is defined inductively on the structure of Σ-formulas:

- For all $t : s \to s' \in T_\Sigma$ and $r : s' \in R$, $(r \circ t)^A = (t^A)^{-1}(r^A)$.
- For all $r : s \in R$ and $t : s \to s' \in T_\Sigma$, $(t \circ r)^A = t^A(r^A)$.
- $True^A = \coprod_{s \in S} s^A$ and $False^A = \emptyset$.
- For all $\varphi : s \in Form_\Sigma$, $(\neg\varphi)^A = s^A \setminus \varphi^A$.
- For all $\varphi : s, \psi : s \in Form_\Sigma$, $(\varphi \wedge \psi)^A = \varphi^A \cap \psi^A$ and $(\varphi \vee \psi)^A = \varphi^A \cup \psi^A$.
- For all $\varphi : s \in Form_\Sigma$ and $k \in I$, $(\forall k\varphi)^A = \bigcap\{\varphi^A[b/k] \mid b \in s_k^A\}$ and $(\exists k\varphi)^A = \bigcup\{\varphi^A[b/k] \mid b \in s_k^A\}$.
- For all $\varphi : s_1 \in Form_\Sigma$ and $s_2 \in \mathbb{T}_S$, $\varphi \times s_2^A = \varphi^A \times s_2^A$ and $\varphi + s_2^A = \varphi^A$.

$a \in s^A$ **satisfies** $\varphi : s$ if $a \in \varphi^A$. A **satisfies** $\varphi : s$, written as $A \models \varphi$, if $s^A \subseteq \varphi^A$.[3] A **satisfies** a set F of Σ-formula, written as $A \models F$, if for all $\varphi \in F$, $A \models \varphi$.

Let $SP = (\Sigma, AX)$ be a specification. A is an SP**-model** if A satisfies AX. $Mod(SP)$ denotes the full subcategory of $Mod(\Sigma)$ whose objects are SP-models. $Mod_{EU}(SP)$ denotes the full subcategory of $Mod_{EU}(\Sigma)$ whose objects are SP-models.

Given $S_1 \subseteq S$ and an S_1-sorted set A, $Mod(SP, A) =_{def} Mod(\Sigma, A) \cap Mod(SP)$ and $Mod_{EU}(SP, A) =_{def} Mod(\Sigma, A) \cap Mod_{EU}(SP)$. ❏

Proposition 4.4 *Let $\Sigma = (S_0, S, F, R)$ be a signature, $sub : S \to T_\Sigma$, A be a Σ-structure and $h : A \to B$ be an S-sorted function such that for all $s \in S$, $h_s = sub(s)^A$. Then for all $dom \in \mathbb{T}_S$, $h_{dom} = sub^*(dom)$.* ❏

Definition 4.5 *(reduct)* Let $\Sigma = (S_0, S, F, R)$, $\Sigma = (S'_0, S', F', R')$ be signatures and A be a Σ'-structure. Given a signature morphism $\sigma : \Sigma \to \Sigma'$, the σ**-reduct of** A, $A|_\sigma$, is the Σ-structure defined by $(A|_\sigma)_s = A_{\sigma(s)}$ for all $s \in \mathbb{T}_S$ and $f^{A|_\sigma} = \sigma(f)^A$ for all $F \cup R$.

Given a Σ'-homomorphism $h : A \to B$, $h|_\sigma : A|_\sigma \to B|_\sigma$ denotes the Σ'-homomorphism defined by $h|_\sigma(a) = h(a)$ for all $a \in A|_\sigma$.

If $\Sigma \subseteq \Sigma'$ and σ is the inclusion, then we write $A|_\Sigma$ instead of $A|_\sigma$ and call $A|_\Sigma$ the Σ**-reduct of** A. ❏

Proposition 4.6 *Let $\Sigma = (S, F, R)$ and $\Sigma = (S', F', R')$ be signatures, A be a Σ-structure and B be a Σ'-structure.*

(1) For all Σ-terms or -coterms $t : s \to s', t' : s' \to s''$, $(t' \odot t)^A = (t')^A \circ t^A$.
(2) For all signature morphisms $\sigma : \Sigma \to \Sigma'$ and Σ-terms t, $t^{B|_\sigma} = \sigma(t)^B$.
(3) For all Σ-homomorphisms $h : A \to B$ and Σ-terms $t : s \to s'$, $h_s \circ t^A = t^B \circ h_{s'}$.
(4) For all Σ-homomorphisms $h : A \to B$, $s \in S$ and $t \in MGen_{\Sigma,s}$, $h_s \circ t^A = t^B$.
(5) For all Σ-homomorphisms $h : A \to B$, $s \in S$ and $t \in MObs_{\Sigma,s}$, $t^B \circ h_s = t^A$. ❏

Proposition 4.6(1) tells us that each Σ-structure A provides a **functor** from the term category $\mathcal{T}_\Sigma$ to Set^S

[3] The inverse inclusion is always valid.

(see section 2):

$$
\begin{tikzcd}
\bullet \arrow[r, dash] \arrow[dd, dash, "t' \odot t"'] & s \arrow[r, mapsto, "\_^A"] \arrow[d, "t"'] & s^A \arrow[r, dash] \arrow[d, "t^A"] & \bullet \arrow[dd, dash, "(t' \odot t)^A"] \\
= & s' \arrow[r, mapsto, "\_^A"] \arrow[d, "t'"'] & (s')^A \arrow[d, "(t')^A"] & = \\
\bullet \arrow[r] & s'' \arrow[r, mapsto, "\_^A"] & (s'')^A & \bullet \arrow[l]
\end{tikzcd}
$$

By Proposition 4.6(3), each Σ-term $t : s \to s'$ yields a **natural transformation** from the functor $\_s$ to the functor $\_{s'}$ (see section 2) in the case that these functors are restricted from Set^S to the subcategory $Mod(\Sigma)$ of Σ-structure and -homomorphisms (see above):

$$
\begin{tikzcd}
A_s \arrow[r, "t^A"] \arrow[d, "h_s"'] \arrow[dr, phantom, "="] & A_{s'} \arrow[d, "h_{s'}"] \\
B_s \arrow[r, "t^B"] & B_{s'}
\end{tikzcd}
$$

Proposition 4.7 *Let $\Sigma = (S_0, S, F, R)$ and $\Sigma = (S_0', S', F', R')$ be signatures, A be a Σ-structure and B be a Σ'-structure.*

(1) For all $t : s \to s' \in T_\Sigma$ and $\varphi : s' \in Form_\Sigma$, $(\varphi \odot t)^A = (t^A)^{-1}(\varphi^A)$.

(2) For all $\varphi : s \in Form_\Sigma$ and $t : s \to s' \in T_\Sigma$, $(t \odot \varphi)^A = t^A(\varphi^A)$.

(3) For all signature morphisms $\sigma : \Sigma \to \Sigma'$ and Σ-formulas φ, $B|_\sigma \models \varphi$ iff $B \models \sigma(\varphi)$.
If for all $s \in \mathbb{T}_S$, $\sigma(s) = s$, then $\varphi^{B|_\sigma} = \sigma(\varphi)^B$.

(4) For all Σ-formulas $\varphi : s$ and $\psi : s$, $A \models (\varphi \Rightarrow \psi)$ iff $\varphi^A \subseteq \psi^A$.

(5) Let A be a Σ-structure with equality. For all Σ-terms $t, u : s \to s'$,
$(t \equiv u)^A = \{a \in s^A \mid t^A(a) = u^A(a)\}$.

(6) For all $\varphi : \prod_{i \in I} s_i \in Form_\Sigma$ and $k \in I$, $\begin{array}{l} (\forall k\varphi)^A = \{a \in \prod_{i\in I} s_i^A \mid \forall\ b \in s_k^A : a[b/k] \in \varphi^A\}, \\ (\exists k\varphi)^A = \{a \in \prod_{i\in I} s_i^A \mid \exists\ b \in s_k^A : a[b/k] \in \varphi^A\}. \end{array}$

(7) Let A be a Σ-structure with equality. For all Σ-terms $t : s_x \to s_y$ and Σ-formulas $\varphi : s_y$,

$$A \ \models\ \varphi \odot t \quad \textit{iff} \quad A\ \models\ \exists y(\varphi \times s_x \wedge \pi_y \equiv t \odot \pi_x).$$

Proof. (1) to (6) are easy to show. The proof of (7) is also straightforward. We present it here in detail for illustrating how logical equivalences known from applicative first-order logic carry over to our variable-free logic.

$$
\begin{array}{ll}
 & A \models \varphi \odot t \iff (\varphi \odot t)^A = s_x^A \stackrel{(1)}{\iff} (t^A)^{-1}(\varphi^A) = s_x^A \\
\iff & \{a \in s_x^A \mid t^A(a) \in \varphi^A\} = s_x^A \iff \{a \in s_x^A \mid \exists\, b \in s_y^A : b \in \varphi^A \wedge b = t^A(a)\} = s_x^A \\
\iff & \{a \in s_x^A \mid \exists\, b \in s_y^A : b \in \varphi^A \wedge \pi_y(a,b) = t^A(\pi_x(a,b))\} = s_x^A \\
\stackrel{(5)}{\iff} & \{a \in s_x^A \mid \exists\, b \in s_y^A : (a,b) \in s_x^A \times \varphi^A \wedge (a,b) \in (\pi_y \equiv t \circ \pi_x)^A\} = s_x^A \\
\iff & \{a \in s_x^A \mid \exists\, b \in s_y^A : (a,b) \in \varphi \times s_x^A \wedge (a,b) \in (\pi_y \equiv t \circ \pi_x)^A\} = s_x^A \\
\iff & \{a \in s_x^A \mid \exists\, b \in s_y^A : (a,b) \in (\varphi \times s_x \wedge \pi_y \equiv t \circ \pi_x)^A\} = s_x^A \\
\iff & \{(a,c) \in s_x^A \times s_y^A \mid \exists\, b \in s_y^A : (a,b) \in (\varphi \times s_x \wedge \pi_y \equiv t \circ \pi_x)^A\} = s_x^A \times s_y^A \\
\iff & \{(a,c) \in s_x^A \times s_y^A \mid \exists\, b \in s_y^A : (a,c)[b/y] \in (\varphi \times s_x \wedge \pi_y \equiv t \circ \pi_x)^A\} = s_x^A \times s_y^A \\
\stackrel{(6)}{\iff} & (\exists y(\varphi \times s_x \wedge \pi_y \equiv t \circ \pi_x))^A = s_x^A \times s_y^A \iff A \models \exists y(\varphi \times s_x \wedge \pi_y \equiv t \odot \pi_x). \quad ❏
\end{array}
$$

If $t : s \to s'$ represents an "action" transforming "states" of type s to states of type s', then Proposition 4.7(1) reflects the *forward-reasoning* semantics of *nexttime* modal-, temporal- or coalgebraic-logic operators [47, 55, 60]: $\varphi \odot t$ holds true in state st iff φ holds true in state $t(st)$. Conversely, Proposition 4.7(2) reflects the *backward-reasoning* of *lasttime* temporal-logic operators (see [55], section 4.3.1): $t \odot \varphi$ holds true in state $t(st)$ iff φ holds true in state st.

Proposition 4.8 *Let A and B be isomorphic Σ-structures and φ be a Σ-formula. A satisfies φ iff B satisfies $\sigma(\varphi)$.* ❏

Proposition 4.9 *Let $\sigma : \Sigma \to \Sigma'$ be a signature morphism, A be a Σ'-structure and φ be a Σ-formula. $A|_\sigma$ satisfies φ iff A satisfies $\sigma(\varphi)$.* ❏

Definition 4.10 *(special structures, kernel, image and product)* Let $\Sigma = (S_0, S, F, R)$ be a signature. A is a **Σ-structure with equality** if for all $s \in S$, $\equiv_s^A = \Delta_s^A$. A is a **Σ-structure with universe** if for all $s \in S$, $all_s^A = s^A$. Given a relation $r : s \in R$, a relation $\overline{r} : s \in R$ is called the **A-complement of** r if $\overline{r}^A = s^A \setminus r^A$.

The **kernel of** h, $ker(h)$, is the S-sorted binary relation

$$\{\{(a,b) \in s^A \times s^A \mid h_s(a) = h_s(b)\} \mid s \in S\}.$$

Let $h : A \to B$ be an S-sorted function. The **image of** h, $img(h)$ (or $h(A)$), is the S-sorted set $\{h(s^A) \mid s \in S\}$. Let h be Σ-homomorphic. Then $img(h)$ can be extended to a Σ-structure:

- for all $s \in S$, $s^{img(h)} = h(s^A) =_{def} \{b \in B \mid \exists\, a \in s^A : h(a) = b)\}$,
- for all $f : s \to s' \in F$ and $b \in s^{img(h)}$, $f^{img(h)}(b) = f^B(b)$,
- for all $r \in R$, $r^{img(h)} = h(r^A)$.

Let Σ be algebraic and A, B be Σ-structures. The following interpretation of Σ extends $A \times B$ to a Σ-structure:

- for all $s \in S$, $s^{A \times B} = s^A \times s^B$,
- for all $f : s \to s' \in F$, $a \in s^A$ and $b \in s^B$, $f^{A \times B}(\langle a,b \rangle) = \langle f^A(a), f^A(b) \rangle$,
- for all $r : s \in R$, $r^{A \times B} = \{\langle a,b \rangle \mid a \in r^A,\ b \in r^B\}$. ❏

Definition 21.7 *(free and cofree structures)* Let $\Sigma = (S_0, S, F, R)$ be a signature, A be an S_0-sorted set and $S_1 = S \setminus S_0$.

Suppose that for all $f : s \to s' \in F$, $s, s' \in \mathbb{T}_{S_0}$ or $s' \in S_1$. The **free Σ-structure over** A, $Free(\Sigma, A)$, is the Σ-structure B with equality and universe that is defined as follows:

- for all $s \in S_0$, $s^B = s^A$,

- for all $s \in S_1$, $s^B = \coprod_{t:dom \to s \in MGen_\Sigma} dom^A$,
- for all $f \in F$ and $a \in dom_f^B$,

$$f^B(a) = \begin{cases} (b, f \circ t) & \text{if } dom_f \in S \text{ and } a = (b,t), \\ ((b_i)_{i\in I}, f \circ \prod_{i\in I} t_i) & \text{if } dom_f = \prod_{i\in I} s_i \text{ and } a = (b_i, t_i)_{i\in I}, \\ (b, f \circ \iota_i \circ t) & \text{if } dom_f = \coprod_{i\in I} s_i \text{ and } a = ((b,t),i). \end{cases}$$

Suppose that for all $f : s \to s' \in F$, $s, s' \in \mathbb{T}_{S_0}$ or $s \in S_1$. The set $Beh^{\Sigma,A}$ of **Σ-behaviours** is the greatest S-sorted subset[4] of all $a \in \prod_{t:s\to ran \in MObs_\Sigma} ran^A$ such that

(1) for all $s \in S_0$, $Beh_s^{\Sigma,A} = s^A$,

(2) for all $f : s \to \coprod_{i\in I} s_i \in F$ there is $i_{a,f} \in I$ such that for all $T = \coprod_{i\in I} t_i \in \coprod_{i\in I} MObs_{\Sigma,s_i}$, $a_{T\circ f} \in ran^A_{i_{a,f}}$,

(3) for all $f : s \to s' \in F$, $(a_t)_{t\in MObs_{\Sigma,s'}} \in Beh_{s'}^{\Sigma,A}$ implies $(a_{t\circ f})_{t\in MObs_{\Sigma,s'}} \in Beh_s^{\Sigma,A}$.

The **cofree Σ-structure over** A, $Cofree(\Sigma, A)$, is the Σ-structure B with equality and universe that is defined as follows:

- for all $s \in S$, $s^B = Beh_s^{\Sigma,A}$,
- for all $f \in F$ and $a \in dom_f^B$,

$$f^B(a) = \begin{cases} (a_{t\circ f})_{t:ran_f \to s \in MObs_\Sigma} & \text{if } ran_f \in S, \\ ((a_{t_i\circ\pi_i\circ f})_{t_i \in MObs_{\Sigma,s_i}})_{i\in I} & \text{if } ran_f = \prod_{i\in I} s_i, \\ ((a_{T\circ f})_{T\in\coprod_{i\in I} MObs_{\Sigma,s_i}}, i_{a,f}) & \text{if } ran_f = \coprod_{i\in I} s_i.^5 \end{cases} \quad ❏$$

Hence, roughly said, the free Σ-structure is a sum over the set of maximal Σ-generators, while the cofree Σ-structure is a product over the set of maximal Σ-observers.

By Proposition 4.2, for all types s over S, $Free(\Sigma, A)_s = \coprod_{t:dom\to s\in MGen_\Sigma} dom^A$ and $Cofree(\Sigma, A)_s \subseteq \prod_{t:s\to ran \in MObs(\Sigma',S)} ran^A$. This is needed for the implicit assumption about the structure of the domain of $f^{Free(\Sigma,A)}$ and the range of $f^{Cofree(\Sigma,A)}$.

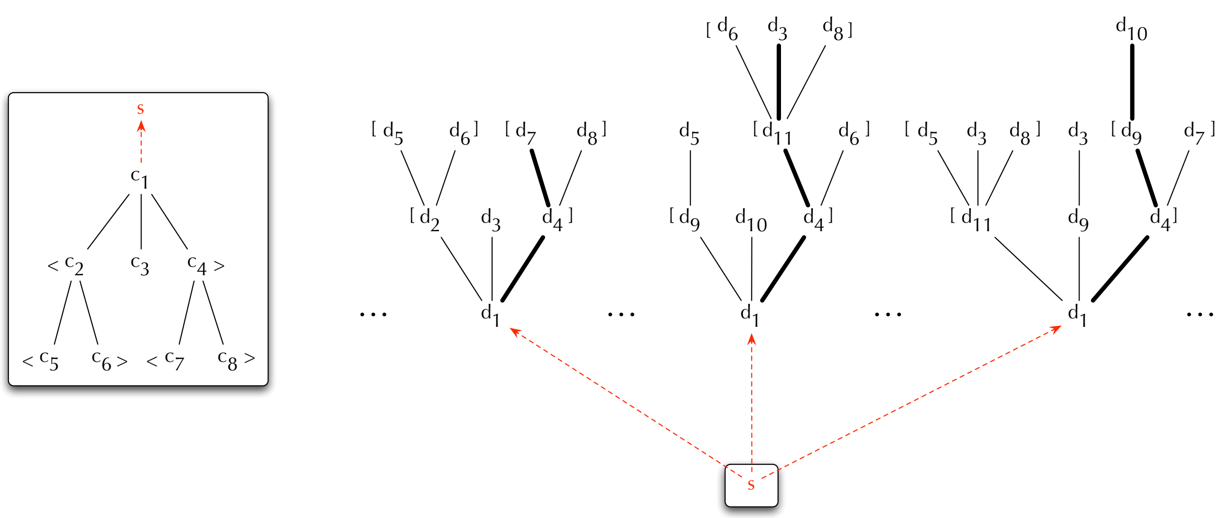


Figure 6.1. *Two elements of a free structure (left) and a cofree structure (right), respectively.*

At the first sight, one might define s^B for $s \in S_1$ as the entire product P that represents, for each s-object a, the set of *all* possible tuples of observations of a. However, a closer look at such a tuple reveals that some of

[4] The existence of this subset follows from Theorem 8.3(1).

[5] $i_{a,f}$ refers to (2).

its components depend on each other whenever their indices (generators), say t and t', have a common prefix into a coproduct, say $u : dom \to \coprod_{i\in I} s_i$. Then t and t' also map to coproducts, say $ran_t = \coprod_{i\in I} ran_i$ and $ran_{t'} \coprod_{i\in I} ran'_i$. Since t and t' observe the *same* object a, the resulting observations belong to two summands of ran_t resp. ran'_t with the *same* index, which is determined by the branch that—intuitively speaking—a takes when passing u. The actual definition of s^B selects exactly those tuples from P that take into account this dependency between observers into coproducts. If F does not include destructors into coproducts, s^B coincides with the entire product.

Given a sets S of sorts and an S-sorted set A, an S-sorted relation $\sim \subseteq A^2$ and an S-sorted set $B \subseteq A$ extend to $\mathbb{T}_S$-sorted sets as follows: Let $\{s_i\}_{i\in I} \subseteq \mathbb{T}_S$.

- For all $a, b \in \prod_{i\in I} s_i^A$, $a \sim_{\prod_{i\in I} s_i} b \iff_{def} \forall\, i \in I : a_i \sim_{s_i} b_i$.
- For all $i \in I$, $a \in s_i^A$ and $b \in A_{s_j}$, $(a, i) \sim_{\coprod_{i\in I} s_i} (b, j) \iff_{def} i = j \wedge a \sim_{s_i} b$.
- For all $a \in \prod_{i\in I} s_i^A$, $a \in (\prod_{i\in I} s_i)^B \iff_{def} \forall\, i \in I : a_i \in s_i^B$.
- For all $i \in I$, $a \in s_i^A$, $(a, i) \in (\coprod_{i\in I} s_i)^B \iff_{def} a \in s_i^B$.

Extending $\sim$ and B to products or sums is called **relation** resp. **predicate lifting** (see [55], §3.1 resp. 4.1). If liftings are regarded as mere extensions, the usual difference between a congruence and a bisimulation becomes obsolete. The former stands for compatibility with constructors, the latter for compatibility with destructors (see [55], Def. 3.1.2). Hence we subsume bisimulations under congruences. In contrast to binary ones, a notional difference between the two kinds of compatibility has never been made in the case of unary relations: both are called invariants (see [55], Def. 4.2.1).

Definition 4.12 *(congruences and invariants)* Let $\Sigma = (S_0, S, F, R)$ be a signature, $F_1 \subseteq F$ and A be an S-sorted set.

An **S-sorted binary relation on** A is a family $\sim = \{\sim_s \subseteq s^A \times s^A \mid s \in S\}$ of binary relations. $\sim$ is **F_1-compatible** if for all $f : s \to s' \in F_1$ and $a, b \in s^A$, $a \sim_s b$ implies $f^A(a) \sim_{s'} f^A(b)$. $\sim$ is **Σ-congruent** or a **Σ-congruence** if $\sim$ is F-compatible and for all $s \in S_0$, $\sim_s = \Delta_s^A$. $\sim$ is **R-compatible** if for all $r : s \in R$ and $a, b \in s^A$, $a \in r^A$ and $a \sim b$ imply $b \in r^A$.

Given an S-sorted binary relation $\sim$ on A, the least equivalence relation including $\sim$ is called the **equivalence closure** of $\sim$ and denoted by $\sim^{eq}$. If $\sim$ is Σ-congruent, then the **$\sim$-quotient of** A, $A/{\sim}$, is the Σ-structure that is defined as follows: For all $a \in A$, $[a] =_{def} \{b \in A \mid a \sim^{eq} b\}$ is the **equivalence class** of a.

- For all $s \in S$, $(A/{\sim})_s = \{[a] \mid a \in s^A\}$,
- for all $f : s \to s' \in F$ and $a \in s^A$, $f^{A/\sim}([a]) = [f^A(a)]$,
- for all $r : s \in R$, $r^{A/\sim} = \{c \in (A/{\sim})_s \mid c \cap r^A \neq \emptyset\}$.

The function $nat : A \to A/{\sim}$ that maps $a \in A$ to its equivalence class $[a]$ w.r.t. $\sim$ is called a **natural mapping**.

An S-sorted subset inv of A is **F_1-compatible** if for all $f : s \to s' \in F$ and $a \in inv_s$, $f^A(a) \in inv_{s'}$. inv is a **Σ-invariant** (on A) if inv is F-compatible and for all $s \in S_0$, $inv_s = s^A$. If inv is Σ-invariant, then the **inv-substructure of** A, $A|inv$, is the Σ-structure that is defined as follows:

- For all $s \in S$, $(A|inv)_s = inv_s$,
- for all $f : s \to s' \in F$ and $a \in (A|inv)_s$, $f^{A|inv}(a) = f^A(a)$,
- for all $r : s \in R$, $r^{A|inv} = r^A \cap inv_s$.

The function $inc : A|inv \to A$ that maps $a \in inv$ to itself is called an **inclusion mapping**.

A quotient resp. substructure B of A is a **proper quotient** resp. **proper substructure** of A if A and B

are not isomorphic. ❑

Proposition 4.13 *Given a Σ-homomorphism $h : A \to B$, the image of h is a Σ-invariant and the kernel of h is a Σ-congruence. If Σ is algebraic, then a Σ-congruence on A is a Σ-invariant on $A \times A$ and thus extends to a Σ-structure. The equivalence closure of a Σ-congruence is Σ-congruent. Natural mappings, inclusion mappings and the projections on the components of a product are Σ-homomorphic.* ❑

Proposition 4.14 *Let Σ be a signature, A, B be Σ-structures, $\sim$ be a Σ-congruence on A and inv be a Σ-invariant on A.*

- *For all $t : s \to s' \in T_\Sigma$ and $a \in s^A$, $t^{A/\sim}([a]) = [t^A(a)]$.*
- *For all $t : s \to s' \in T_\Sigma$ and $a \in inv_s$, $t^{A|inv}(a) = t^A(a)$.*
- *If Σ is algebraic, then for all products s, s' of sorts, $t : s \to s' \in T_\Sigma$, $a \in s^A$ and $b \in s^B$, $s^{A\times B} = \{\langle a, b\rangle \mid a \in s^A,\ b \in s^B\}$ and $t^{A\times B}(\langle a, b\rangle) = \langle t^A(a), t^A(b)\rangle$.* ❑

Proposition 4.15 *Let A be an SP-model that interprets $\equiv$ as a Σ-congruence. Then for all Horn or co-Horn clauses φ there is a normalized Horn resp. co-Horn clause ψ such that A satisfies φ iff A satisfies ψ.*

Proof. Let $\varphi = (r \circ t \Leftarrow \theta)$, $\prod_{i\in I} s_i$ be the type of r and $\prod_{i\in J} s'_i$ be the type of φ. W.l.o.g. I and J are disjoint. The conjecture holds true for $\psi = (r \Leftarrow \exists J : \langle\pi\rangle_{i\in J} \equiv t \wedge \theta'))$ where θ' is constructed from θ by driving all negation symbols innermost until they directly precede atomic formulas. If $\varphi = (r \Rightarrow \theta)$, then the conjecture holds true for $\psi = (r \circ t \Rightarrow \forall I : (\neg(\pi_s \equiv t) \vee \theta'))$. ❑

The congruence property of $\equiv$ is essential for the validity of Proposition 4.15.

Lemma 4.16 *(homomorphism criteria) Let $h : A \to C$ be a Σ-homomorphism.*

(1) Let $g : A \to B$ be a Σ-epimorphism and $h' : B \to C$ be a function such that $h = h' \circ g$. Then h' is a Σ-homomorphism and the only one satisfying $h = h' \circ g$.

(2) Let $g : B \to C$ be a Σ-monomorphism and $h' : A \to B$ be a function such that $h = g \circ h'$. Then h' is a Σ-homomorphism and the only one satisfying $h = g \circ h'$.

Proof. (1) h' is homomorphic: Let $f : s \to s' \in F$. Then

$$f^C \circ h' \circ g = f^C \circ h = h \circ f^A = h' \circ g \circ f^A = h' \circ f^B \circ g$$

and thus $f^C \circ h' = h' \circ f^B$ because g is surjective. Suppose that $h'' \circ g = h$ for some Σ-homomorphism $h'' : B \to C$. Then $h'' \circ g = h = h' \circ g$ and thus $h'' = h'$ because g is surjective.

(2) h' is homomorphic: Let $f : s \to s' \in F$. Then

$$g \circ h' \circ f^A = h \circ f^A = f^C \circ h = f^C \circ g \circ h' = g \circ f^B \circ h'$$

and thus $h' \circ f^A = f^B \circ h'$ because g is injective. Suppose that $g \circ h'' = h$ for some Σ-homomorphism $h'' : A \to B$. Then $g \circ h'' = h = g \circ h'$ and thus $h'' = h'$ because g is injective. ❑

Theorem 4.17 *(homomorphism theorems) Let $h : A \to C$ be a Σ-homomorphism.*

(1) Let $nat : A \to A/ker(h)$ be the corresponding natural mapping. Then there is a unique Σ-homomorphism $h' : A/ker(h) \to C$ such that $h' \circ nat = h$. Moreover, C is (isomorphic to) a quotient of A iff there is a Σ-epimorphism from A to C.

(2) Let $inc : img(h) \to C$ be the corresponding inclusion mapping. Then there is a unique Σ-homomorphism $h' : A \to img(h)$ such that $h = inc \circ h'$. Moreover, A is (isomorphic to) a substructure of C iff there is a Σ-monomorphism from A to C.

Proof.

$$\begin{tikzcd} A \arrow[rr, "h"] \arrow[dr, "nat"'] & \arrow[d, phantom, "="] & C \\ & A/ker(h) \arrow[ur, dashed, "h'"'] & \end{tikzcd}$$

(1) nat is a Σ-epimorphism. By the definition of $ker(h)$, the function $h' : A/ker(h) \to C$ sending $[a] \in A/ker(h)$ to $h(a)$ is well-defined and injective. Hence $h = h' \circ nat$ and thus by Lemma 4.16(1), h' is a Σ-homomorphism and the only one with $h = h' \circ nat$.

If h is surjective, then h' is also surjective and thus bijective, i.e., $A/ker(h)$ and C are isomorphic. Conversely, let $\sim$ be a Σ-congruence on A and $h' : A/ker(h) \to C$ be a Σ-isomorphism. Define $h : A \to C$ by $h = h' \circ nat$. Since nat and h' are Σ-epimorphisms, h is a Σ-epimorphism, too.

$$\begin{tikzcd} A \arrow[rr, "h"] \arrow[dr, dashed, "h'"'] & \arrow[d, phantom, "="] & C \\ & img(h) \arrow[ur, "inc"'] & \end{tikzcd}$$

(2) inc is a Σ-monomorphism. By the definition of $img(h)$, the function $h' : A \to img(h)$ sending $a \in A$ to $h(a)$ is well-defined and surjective. Hence $h = inc \circ h'$ and thus by Lemma 4.16(2), h' is a Σ-homomorphism and the only one with $h = inc \circ h'$.

If h is injective, then h' is also injective and thus bijective, i.e., A and $img(h)$ are isomorphic. Conversely, let C' be a Σ-substructure of C and $h' : A \to A'$ be a Σ-isomorphism. Define $h : A \to C$ by $h = inc \circ h'$. Since inc and h' are Σ-monomorphisms, h is a Σ-monomorphism, too. ❏

5 Swinging types

Definition 5.1 *(specification, swinging type)* Given a signature Σ and a set AX of Horn or co-Horn clauses over Σ, called **axioms**, the pair $SP = (\Sigma, AX)$ is a **specification** if each relation $r \in \Sigma$ is a **predicate** or a **copredicate**, i.e., occurs only in the heads of Horn clauses or only in the heads of co-Horn clauses.

Given signatures $\Sigma = (S_0, S, F, R)$ and $\Sigma' = (S_0', S', F', R')$, a specification $SP' = (\Sigma', AX')$ is a **swinging type (ST)** with **base type** $SP = (\Sigma, AX)$ and **primitive sort set** S_0' if SP is a swinging type and either $SP' = SP = (\emptyset, \emptyset)$ or one of the following conditions (1) to (8) holds true.

Let $S_1 = S' \setminus S$, $equals = \{\equiv_s \mid s \in S_1\}$ and $univs = \{all_s \mid s \in S_1\}$.

(1) *Data model.* $S_0' = \mathbb{T}_S$, $R' = R$ and $AX' = AX$. $F' \setminus F$ is a set of S_1-constructors (see Def. 3.6). The sorts of S_1 are called **visible sorts** of SP'.

(2) *State model.* $S_0' = \mathbb{T}_S$, $R' = R$ and $AX' = AX$. $F' \setminus F$ is a set of S_1-destructors (see Def. 3.6). The sorts of S_1 are called **hidden sorts** of SP'.

(3) *Recursive functions.* SP satisfies (1). $\Sigma' \setminus \Sigma$ is a set of S_1-destructors, called **recursive functions**. Define the S'-sorted set rec and the substitutions $sub_1 : S' \to \mathbb{T}_{S'}$ and $sub_2 : S' \to T_{\Sigma'}$ as follows: For all $s \in S$, $rec(s) = \{id_s\}$, for all $s \in S_1$, $rec(s) = \{f \in F' \setminus F \mid dom_f = s\}$, and for all $s \in S'$, $sub_1(s) = \prod_{f \in rec(s)} ran_f$ and $sub_2(s) = \langle rec(s) \rangle$. For all $f \in \Sigma' \setminus \Sigma$ and all S_1-constructors $c : dom \to dom_f$ there

is a Σ-term

$$t_{f,c} : sub_1^*(dom) \to ran_f$$

such that $AX' \setminus AX$ contains the equation

$$f \circ c \equiv t_{f,c} \odot sub_2^*(dom).$$

These are the only axioms of $AX' \setminus AX$.[6]

(4) *Corecursive functions.* SP satisfies (2). $\Sigma' \setminus \Sigma$ is a set of S_1-constructors, called **corecursive functions**. Define the S'-sorted set cor and the substitutions $sub_1 : S' \to \mathbb{T}_{S'}$ and $sub_2 : S' \to T_{\Sigma'}$ as follows: For all $s \in S$, $cor(s) = \{id_s\}$, for all $s \in S_1$, $cor(s) = \{f \in F' \setminus F \mid ran_f = s\}$, and for all $s \in S'$, $sub_1(s) = \coprod_{f \in cor(s)} dom_f$ and $sub_2(s) = [cor(s)]$. For all $f \in \Sigma' \setminus \Sigma$ and all Σ-destructors $d : ran_f \to ran$ there is a Σ-term

$$t_{f,d} : dom_f \to sub_1^*(ran)$$

such that $AX' \setminus AX$ contains the equation

$$d \circ f \equiv sub_2^*(ran) \odot t_{f,d}.$$

These are the only axioms of $AX' \setminus AX$.[7]

(5) *Visible abstraction.* SP is visible. $\Sigma' \setminus \Sigma$ is a set of S_1-constructors and logical relations. $AX' \setminus AX$ consists of $(R_1 \cup equals)$-positive Horn clauses for $R' \setminus R \cup equals$ and includes CONH (see Def. 10.1).

(6) *Hidden abstraction.* $\Sigma' \setminus \Sigma$ is a set of logical relations. $AX' \setminus AX$ consists of $(R_1 \cup equals)$-positive co-Horn clauses for $R' \setminus R \cup equals$ and includes CONC (see Def. 10.1).

(7) *Visible restriction.* $\Sigma' \setminus \Sigma$ is a set of logical relations. $AX' \setminus AX$ consists of $(R_1 \cup univs)$-positive and restricted Horn clauses for $R' \setminus R \cup univs$ and includes INVH (see Def. 10.1).

(8) *Hidden restriction.* SP is hidden. $\Sigma' \setminus \Sigma$ is a set of S_1-destructors and logical relations. $AX' \setminus AX$ consists of $(R_1 \cup univs)$-positive and restricted co-Horn clauses for $R' \setminus R \cup univs$ and includes INVC (see Def. 10.1).

In cases (5) and (6), SP' is an **abstraction**. In cases (7) and (8), SP' is a **restriction**. In cases (1), (3), (5) and (7), SP' is **visible**. In cases (2), (4), (6) and (8), SP' is **hidden**.

A **predecessor** of SP' is a swinging type SP_0 such that there are swinging types $SP_1, \dots, SP_n = SP'$ and for all $1 \leq i \leq n$, SP_{i-1} is the base type of SP_i. The **sort-building predecessor** of SP' is the least[8] predecessor SP of SP' such that both specifications have the same set of sorts. ❑

The sort-building predecessor of SP' always satisfies 5.1(1) or (2).

Example 5.2 A visible type of Boolean arithmetic reads as follows.

BOOL

`vissorts`	$bool$	
`constructs`	$true, false : 1 \to bool$	
`defuncts`	$not : bool \to bool$	
	$and, or, eq : bool \times bool \to bool$	
`preds`	$\_ \not\equiv \_ : bool \times bool$	
`vars`	$b, c : bool$	
`axioms`	$not \circ true \equiv false$	$true \; and \; b \equiv b$

[6] For the category-theoretic presentation of recursion, see, e.g., [65], Section 5.
[7] For the category-theoretic presentation of corecursion, see, e.g., [2], §§5.11-5.14.
[8] with respect to set inclusion

$$\begin{array}{ll} not \circ false \equiv true & false \; and \; b \equiv false \\ or \circ \langle true, id \rangle \equiv true & eq(true, b) \equiv b \\ false \; or \; b \equiv b & eq(false, b) \equiv not(b) \\ true \not\equiv false & false \not\equiv true \end{array}$$

Several swinging types given later have BOOL as a visible subtype. ❏

Definition 5.3 *(model-based specification, parameterized type)* Let $SP = (\Sigma, AX)$ be a specification, $\Sigma = (S_0, S, F, R)$, $\Sigma_1 \subseteq \Sigma$ and A be a Σ_1-structure. The pair $SP(A) = (\Sigma(A), AX(A))$ with

$$\begin{array}{rcl} \Sigma(A) & = & \Sigma \cup \{a : 1 \to s \mid a \in s^A\}, \\ AX(A) & = & AX \cup \{f \circ a \equiv f^A(a) \mid f : s \to s' \in F, \; a \in s^A\} \cup \{r \circ a \mid r \in R, \; a \in r^A\} \end{array}$$

is the **specification based on** (SP, A).

A **parameter type** $PAR = (P\Sigma, PAX)[SP_1, \ldots, SP_k]$ consists of a set $P\Sigma$ of signature elements, a set PAX of formulas and $k \geq 0$ swinging types $SP_i = (\Sigma_i, AX_i)$, $1 \leq i \leq k$, called **constant subtypes** of PAR, such that

$$\Sigma(PAR) \;=_{def}\; P\Sigma \cup \bigcup_{i=1}^{k} \Sigma_i$$

is a signature and

$$AX(PAR) \;=_{def}\; PAX \cup \bigcup_{i=1}^{k} AX_i$$

is a set of $\Sigma(PAR)$-formulas. A **parameterized (swinging) type** $PSP = (\Sigma, AX)[PAR_1, \ldots, PAR_n]$ consists of a set Σ of signature elements, a set AX of Horn or co-Horn clauses and $n \geq 0$ parameter types $PAR_1, \ldots, PAR_n$ such that

$$\Sigma(PSP) \;=_{def}\; \Sigma \cup \bigcup_{i=1}^{n} \Sigma(PAR_i)$$

is the signature and AX is the set of axioms of a swinging type.

Given a parameter type $PAR = (P\Sigma, PAX)[SP_1, \ldots, SP_k]$, a $\Sigma(PAR)$-structure A with equality is a **parameter model** of PAR if A satisfies $AX(PAR)$ and for all $1 \leq i \leq k$, $A|_{\Sigma(SP_i)} \cong Ini(SP_i)$.

Given a parameterized type $PSP = (\Sigma, AX)[PAR_1, \ldots, PAR_n]$ and parameter models $A_1, \ldots, A_n$ of $PAR_1, \ldots, PAR_n$, respectively, the **actualization of** PSP **by** $(A_1, \ldots, A_n)$ is the swinging type

$$(\Sigma, AX)[A_1, \ldots, A_n] \;=_{def}\; (\Sigma(A_1) \cup \cdots \cup \Sigma(A_n), AX(A_1) \cup \cdots \cup AX(A_n)). \quad ❏$$

Example 5.4 The simplest parameter type consists of a single sort with equality and inequality:

NEQ(s)
  `sorts` s
  `preds` $\_ \not\equiv \_ : s \times s$
  `axioms` $x \not\equiv y \;\Leftrightarrow\; \neg(x \equiv y)$

Since s is a type variable, s may be instantiated by any other type. For example, NEQ(*bool*) is the parameter type that agrees with NEQ(s) except for the replacement of all occurrences of s in NEQ(s) by *bool*. Here is a parameterized type using NEQ(s) as parameter:

STACK[NEQ(s)] `where` STACK =
  `primsorts` s

`vissorts`	$stack(s)$
`constructs`	$empty : 1 \to stack(s)$
	$push : s \times stack \to stack(s)$
`defuncts`	$pop : stack(s) \to stack(s)$
	$top : stack(s) \to 1 + s$
`preds`	$\_ \not\equiv \_ : stack(s) \times stack(s)$
`vars`	$x, y : s \quad L, L' : stack(s)$
`axioms`	$top \circ empty \equiv \iota_1$
	$top \circ push \equiv \iota_2 \circ \pi_1$
	$pop \circ empty \equiv empty$
	$pop \circ push \equiv \pi_2$
	$empty \not\equiv push(x, L)$
	$push(x, L) \not\equiv empty$
	$push(x, L) \not\equiv push(x, L') \Leftarrow L \not\equiv L'$

Finally, we extend NEQ(s) to a parameter type with a constant subtype:[9]

TRIV(s)[BOOL] `where` TRIV(s) = NEQ(s) `and`

`functs`	$eq, neq : s \times s \to bool$
`axioms`	$eq(x, y) \equiv true \Leftarrow x \equiv y$
	$eq(x, y) \equiv false \Leftarrow x \not\equiv y$
	$neq(x, y) \equiv true \Leftarrow x \not\equiv y$
	$neq(x, y) \equiv false \Leftarrow x \equiv y$

❏

hidden abstraction ***** final algebra semantics [109, 57, 78]

Labelled transition systems may be integrated into swinging types by declaring them as ternary relations. Although Def. 5.1 excludes the specification of alternating fixpoints [84], it is sufficient for axiomatizing all common modal- or temporal-logic operators in terms of swinging types (see also [89], Example 2.7). General results that have been drawn from the incorporation of modal into many-sorted logic and that may be fruitful for modal logic itself deal with *bisimulation invariant* formulas and their syntactic characterization (see [89], Theorems 3.8 and 7.9).

6 Data and states

Definition 6.1 *(initial, final)* Let Σ be a signature and $\mathcal{C}$ be a class of Σ-structures.

$Ini \in Mod(\Sigma)$ is **initial in** $\mathcal{C}$ or the **initial object of** $\mathcal{C}$ if $Ini \in \mathcal{C}$ and for all $B \in \mathcal{C}$ there is a unique Σ'-homomorphism from Ini to B.

$Fin \in Mod(\Sigma)$ is **final in** $\mathcal{C}$ or the **final object of** $\mathcal{C}$ if $Fin \in \mathcal{C}$ and for all $B \in \mathcal{C}$ there is a unique Σ'-homomorphism from B to Fin. ❏

Lemma 6.2 *Let Σ be a signature and $\mathcal{C}$ be a class of Σ-structures.*

(1) All initial objects of $\mathcal{C}$ are Σ-isomorphic.

(2) All final objects of $\mathcal{C}$ are Σ-isomorphic.

(3) Initial objects of $\mathcal{C}$ do not have proper substructures in $\mathcal{C}$.
Consequently, for all initial objects $A \in \mathcal{C}$ and Σ-invariants inv on A, $inv^A = A$.

[9] The operator `and` denotes the componentwise union of its arguments. It is adopted from the algebraic specification language CASL [12].

(4) *Final objects of $\mathcal{C}$ do not have proper quotients in $\mathcal{C}$.*
Consequently, for all final objects $A \in \mathcal{C}$ and Σ-congruences $\sim$ on A, $\sim^A = \Delta^A$.

(5) *Let Ini be initial in $\mathcal{C}$. Then for all $A \in \mathcal{C}$, the image of the unique Σ-homomorphism $h : Ini \to A$ is the least Σ-invariant of A that belongs to $\mathcal{C}$.*

(6) *Let Fin be final in $\mathcal{C}$. Then for all $A \in \mathcal{C}$, the kernel of the unique Σ-homomorphism $h : A \to Fin$ is the greatest Σ-congruence $\sim$ on A such that $A/{\sim} \in \mathcal{C}$.*

Proof. (1) Let A, B be initial in $\mathcal{C}$. Then there are Σ-homomorphisms $g : A \to B$ and $h : B \to A$. Hence $h \circ g$ and id^A are Σ-homomorphisms from A to A, and $g \circ h$ and id^B are Σ-homomorphisms from B to B. By uniqueness, $h \circ g = id^A$ and $g \circ h = id^B$.

(2) Analogously.

(3) Let Ini be initial in $\mathcal{C}$, A be a substructure of Ini, h be the unique Σ-homomorphism from Ini to A and inc be the Σ-homomorphic inclusion mapping from A to Ini. By (1), $inc \circ h = id^{Ini}$. Hence inc is surjective and thus bijective, i.e., A and Ini are Σ-isomorphic.

(4) Let Fin be final in $\mathcal{C}$, A be a quotient of Fin, h be the unique Σ-homomorphism from A to Fin and nat be the Σ-homomorphic natural mapping from Fin to A. By (2), $h \circ nat = id^{Fin}$. Hence nat is injective and thus bijective, i.e., A and Fin are Σ-isomorphic.

(5) Of course, $h(Ini)$ is a Σ-invariant of A. Let inv be a Σ-invariant of A such that $inv \in \mathcal{C}$. Then there is a Σ-homomorphism $g : Ini \to inv$. Moreover, the inclusion mapping $inc : inv \to A$ is Σ-homomorphic. Hence by uniqueness, $h = inc \circ g$ and thus for all $b \in Ini$, $h(b) = inc(g(b)) = g(b) \in inv$, i.e., $h(Ini) \subseteq inv$.

(6) Of course, $ker(h)$ is a Σ-congruence on A. Let $\sim$ be a Σ-congruence on A such that $A/{\sim} \in \mathcal{C}$. Then there is a Σ-homomorphism $g : A/{\sim} \to Fin$. Moreover, the natural mapping $nat : A \to A/{\sim}$ is Σ-homomorphic. Hence by uniqueness, $h = g \circ nat$ and thus for all $(a, b) \in A^2$, $a \sim b$ implies $h(a) = g(nat(a)) = g(nat(b)) = h(b)$, i.e., ${\sim} \subseteq ker(h)$. ❑

Theorem 6.3 *Let $\Sigma = (S_0, S, F, R)$ be a signature, A be an S_0-sorted set and $S_1 = S \setminus S_0$. Suppose that for all $f : s \to s' \in F$, $s, s' \in \mathbb{T}_{S_0}$ or $s' \in S_1$. The free Σ-structure Ini over A is initial in $Mod_{EU}(\Sigma, A)$. Moreover, for all $t \in MGen_\Sigma$, $t^{Free(\Sigma,A)} = \iota_t$.*

Proof. Let $C \in Mod_{EU}(\Sigma, A)$ and $h : Ini \to C$ be the S-sorted function defined by $h_s = id_s$ for all $s \in S_0$ and

$$h_s \circ \iota_t = t^C \tag{1}$$

for all $s \in S_1$ and $t : dom \to s \in MGen_\Sigma$. First we show by induction on the structure of s that (1) holds true for all $s \in \mathbb{T}_S$. Let $t : dom \to s \in MGen_\Sigma$. If s is a product, say $s = \prod_{i \in I} s_i$, then for all $i \in I$ there is $t_i : dom_i \to s_i \in MGen_\Sigma$ such that $t = \prod_{i \in I} t_i$. Hence for all $a \in dom^A$ and $i \in I$,

$$\pi_i(h_s(\iota_t(a))) = \pi_i(h_{\prod_{i \in I} s_i}(\iota_t(a))) = h_{s_i}(\pi_i(\iota_t(a))) = h_{s_i}(\pi_i(\iota_{\prod_{i \in I} t_i}(a))) = h_{s_i}(\iota_{t_i}(\pi_i(a))) \stackrel{i.h.}{=} t_i^C(\pi_i(a)) =$$
$$\pi_i((t_i^C(\pi_i(a)))_{i \in I}) = \pi_i((\textstyle\prod_{i \in I} t_i^C)(a)) = \pi_i(t^C(a)).$$

If s is a sum, say $s = \coprod_{i \in I} s_i$, then there are $i \in I$ and $u : dom \to s_i \in MGen_\Sigma$ such that $t = \iota_i \circ u$ and thus, by the isomorphism in the proof of Proposition 4.2(1), $\iota_t = \iota_i \circ \iota_u$. Hence for all $a \in dom^A$,

$$h_s(\iota_t(a)) = h_{\coprod_{i \in I} s_i}(\iota_i(\iota_u(a))) = \iota_i(h_{s_i}(\iota_u(a)) \stackrel{i.h.}{=} \iota_i(u^C(a)) = (\iota_i \circ u)^C(a) = t^C(a).$$

Next we show that h is Σ-homomorphic, i.e., for all $f : s \to ran \in F'$, $h_{ran} \circ f^{Ini} = f^C \circ h_s$. Let $s \in S'$ and $a \in s^{Ini}$. If $ran \in \mathbb{T}_S$, then $f \in F$ and thus by Proposition 4.6(4),

$$h_{ran}(f^{Ini}(a)) = h_{ran}(f^A(a)) = f^A(a) = f^A(h_s(a)) = f^C(h_s(a)).$$

Otherwise $f \in F' \setminus F$ and $ran \in S' \setminus S$. If $s \in S'$, then for all $a = (b,t) \in s^{Ini}$,

$$h_{ran}(f^{Ini}(a)) = h_{ran}(b, f{\circ}t) = h_{ran}(\iota_{f\circ t}(b)) \stackrel{(1)}{=} (f{\circ}t)^C(b) = f^C(t^C(b)) \stackrel{(1)}{=} f^C(h_s(\iota_t(b))) = f^C(h_s(b,t)) = f^C(h_s(a)).$$

If $s = \prod_{i\in I} s_i$, then for all $a = (b_i,t_i)_{i\in I} \in \prod_{i\in I} s_i^{Ini}$,

$$h_{ran}(f^{Ini}(a)) = h_{ran}((b_i)_{i\in I}, f \circ \prod_{i\in I} t_i) = h_{ran}(\iota_{f\circ\prod_{i\in I} t_i}((b_i)_{i\in I}) \stackrel{(1)}{=} (f \circ \prod_{i\in I} t_i)^C((b_i)_{i\in I}) =$$
$$f^C((\prod_{i\in I} t_i)^C((b_i)_{i\in I})) = f^C((t_i^C(b_i))_{i\in I}) \stackrel{(1)}{=} f^C((h_s(\iota_{t_i}(b_i)))_{i\in I}) = f^C((h_s(b_i,t_i))_{i\in I}) = f^C(h_s(a)).$$

If $s = \coprod_{i\in I} s_i$, then for all $a = ((b,t),i) \in \coprod_{i\in I} s_i^{Ini}$,

$$h_{ran}(f^{Ini}(a)) = h_{ran}(b, f \circ \iota_i \circ t) = h_{ran}(\iota_{f\circ\iota_i\circ t}(b)) \stackrel{(1)}{=} (f \circ \iota_i \circ t)^C(b) = f^C(\iota_i(t^C(b))) \stackrel{(1)}{=}$$
$$f^C(\iota_i(h_s(\iota_t(b)))) = f^C(\iota_i(h_s(b,t))) = f^C(h(\iota_i(b,t))) = f^C(h_s((b,t),i)) = f^C(h_s(a)).$$

A suitable re-arrangement of the equations in this proof leads to a proof that h is the only Σ-homomorphism from Ini to C. In particular, if $C = Ini$, then $h = id^{Ini}$ and thus (1) implies $t^{Free(\Sigma,A)} = \iota_t$ for all $t \in MGen_\Sigma$. ❑

Theorem 6.4 *Let $\Sigma = (S_0, S, F, R)$ be a signature, A be an S_0-sorted set and $S_1 = S \setminus S_0$. Suppose that for all $f : s \to s' \in F$, $s, s' \in \mathbb{T}_{S_0}$ or $s \in S_1$. The cofree Σ-structure Ini over A is final in $Mod_{EU}(\Sigma, A)$. Moreover, for all $t \in MObs_\Sigma$, $t^{Cofree(\Sigma,A)} = \pi_t$.*

Proof. Let $C \in Mod_{EU}(\Sigma, A)$ and $h : C \to Fin$ be the S-sorted function defined by $h_s = id_s$ for all $s \in S_0$ and

$$\pi_t \circ h_s = t^C \tag{2}$$

for all $s \in S_1$ and $t : s \to ran \in MObs_\Sigma$. First we show by induction on the structure of s that (2) holds true for all $s \in \mathbb{T}_S$. Let $t : s \to ran \in MObs_\Sigma$. If s is a product, say $s = \prod_{i\in I} s_i$, then there are $i \in I$ and $u : s_i \to ran \in MObs_\Sigma$ such that $t = u \circ \pi_i$ and thus, by the isomorphism in the proof of Proposition 4.2(2), $\pi_t = \pi_u \circ \pi_i$. Hence for all $a \in s^{Fin}$,

$$\pi_t(h_s(a)) = \pi_u(\pi_i(h_s(a))) = \pi_u(\pi_i(h_{\prod_{i\in I} s_i}(a))) = \pi_u(h_{s_i}(\pi_i(a))) \stackrel{i.h.}{=} u^C(\pi_i(a)) = (u \circ \pi_i)^C(a) = t^C(a).$$

If s is a sum, say $s = \coprod_{i\in I} s_i$, then for all $i \in I$ there is $t_i : s_i \to ran_i \in MObs_\Sigma$ such that $t = \coprod_{i\in I} t_i$. Hence for all $i \in I$ and $a \in s_i^{Fin}$,

$$\pi_t(h_s(\iota_i(a))) = \pi_t(h_{\coprod_{i\in I} s_i}(\iota_i(a))) = \pi_t(\iota_i(h_{s_i}(a))) = \pi_{\coprod_{i\in I} t_i}(\iota_i(h_{s_i}(a))) = \pi_{t_i}(h_{s_i}(a)) \stackrel{i.h.}{=} t_i^C(a) =$$
$$(\coprod_{i\in I} t_i^C)(\iota_i(a)) = t^C(\iota_i(a)).$$

Next we show that h is Σ-homomorphic, i.e., for all $f : dom \to s \in F'$, $h_s \circ f^C = f^{Fin} \circ h_{dom}$. Let $s \in S'$ and $a \in s^C$. If $dom \in \mathbb{T}_S$, then $f \in F$ and thus by Proposition 4.6(5),

$$h_s(f^C(a)) = h_s(f^A(a)) = f^A(a) = f^A(h_{dom}(a)) = f^{Fin}(h_{dom}(a)).$$

Otherwise $f \in F' \setminus F$ and $dom \in S' \setminus S$. If $s \in S'$, then for all $a \in dom^C$ and $t : s \to ran \in MObs_\Sigma$,

$$\pi_t(h_s(f^C(a))) \stackrel{(2)}{=} t^C(f^C(a)) = (t \circ f)^C(a) \stackrel{(2)}{=} \pi_{t\circ f}(h_{dom}(a)) = h_{dom}(a)_{t\circ f} = \pi_t(f^{Fin}(h_{dom}(a))).$$

If $s = \prod_{i\in I} s_i$, then for all $a \in dom^C$, $i \in I$ and $t_i : s_i \to ran_i \in MObs_\Sigma$,

$$\pi_{t_i}(\pi_i(h_s(f^C(a)))) = \pi_{t_i}(h_{s_i}(\pi_i(f^C(a)))) \stackrel{(2)}{=} t_i^C(\pi_i(f^C(a))) = (t_i \circ \pi_i \circ f)^C(a) \stackrel{(2)}{=} \pi_{t_i\circ\pi_i\circ f}(h_{dom}(a)) =$$
$$h_{dom}(a)_{t_i\circ\pi_i\circ f} = \pi_{t_i}(\pi_i(f^{Fin}(h_{dom}(a)))).$$

If $s = \coprod_{i \in I} s_i$, then for all $a \in dom^C$ and $T = \coprod_{i \in I}(t_i : s_i \to ran_i) \in \coprod_{i \in I} MObs_\Sigma$,

$$\pi_T(h_s(f^C(a))) \stackrel{(2)}{=} T^C(f^C(a)) = (T \circ f)^C(a) \stackrel{(2)}{=} \pi_{T \circ f}(h_{dom}(a)) = h_{dom}(a)_{T \circ f} = \pi_T(f^{Fin}(h_{dom}(a))).$$

A suitable re-arrangement of the equations in this proof leads to a proof that h is the only Σ'-homomorphism from C to Fin. In particular, if $C = Fin$, then $h = id^{Fin}$ and thus (2) implies $t^{Cofree(\Sigma,A)} = \pi_t$ for all $t \in MObs_\Sigma$. ❑

If SP' satisfies Def. 5.1(1), then the sum of all constructors of $SP' \setminus SP$ is an isomorphism:

Theorem 6.5 *Let $SP' = (\Sigma', AX')$ be a swinging type with base type $SP = (\Sigma, AX)$ such that SP' satisfies 5.1(1). There is a Σ'-isomorphism h from the initial object Ini of $Mod_{EU}(SP')$ to the Σ'-structure B with equality and universe that is defined as follows: Let $S_1 = S' \setminus S$ and for all $s \in S_1$, $C(s) = \{c \in F' \setminus F \mid s_c = s\}$.*

- $B|_\Sigma = Ini|_\Sigma$,
- *for all* $s \in S_1$, $s^B = \coprod_{c \in C(s)} dom_c^{Ini}$,
- *for all* $c : dom_c \to s \in F' \setminus F$, $c^B = \iota_c \circ dom_c[[C(s)^{Ini})/s \mid s \in S_1]$.

Moreover, for all $s \in S_1$ and s-constructors $c : dom \to s$, $c^{Ini} = h^{-1} \circ \iota_c$.

Proof. Since SP satisfies 5.1(1), $B \in Mod_{EU}(SP')$ and thus there is a unique Σ'-homomorphism $h : Ini \to B$. First we show that $g : Ini \to Ini$ defined by $g_s = id_s^{Ini|_\Sigma}$ for all $s \in S$ and $g_s = [C(s)^{Ini}) \circ h_s$ for all $s \in S_1$ is a Σ'-homomorphism: For all $c : dom_c \to s \in F' \setminus F$,

$$\begin{aligned} g_s \circ c^{Ini} = [C(s)^{Ini}) \circ h_s \circ c^{Ini} = [C(s)^{Ini}) \circ c^B \circ h_{dom_c} \\ = [C(s)^{Ini}) \circ \iota_c \circ dom_c[[C(s)^{Ini})/s \mid s \in S_1] \circ h_{dom_c} \\ = c^{Ini} \circ dom_c[[C(s)^{Ini}) \circ h_s/s \mid s \in S_1] = c^{Ini} \circ dom_c[g_s/s \mid s \in S_1] = c^{Ini} \circ g_{dom_c} \end{aligned}$$

Since Ini is initial in $Mod_{EU}(SP')$, there is only one Σ'-homomorphism from Ini to Ini. Hence $g = id^{Ini}$.

Next we show that $h' : B \to Ini$ defined by $h'_s = id_s^{Ini|_\Sigma}$ for all $s \in S$ and $h'_s = [C(s)^{Ini})$ for all $s \in S_1$ is an inverse of h. For all $s \in S$, h'_s is an inverse of h_s because $h_s = id_s^{Ini|_\Sigma}$. Let $s \in S_1$. Then

$$h'_s \circ h_s = [C(s)^{Ini}) \circ h_s = g_s = id_s^{Ini}.$$

Moreover, for all s-constructors $c : dom_c \to s$,

$$\begin{aligned} h_s \circ h'_s \circ \iota_c = h_s \circ [C(s)^{Ini}) \circ \iota_c = h_s \circ c^{Ini} = c^B \circ h_{dom_c} \\ = \iota_c \circ dom_c[[C(s)^{Ini})/s \mid s \in S_1] \circ h_{dom_c} = \iota_c \circ dom_c[[C(s)^{Ini}) \circ h_s/s \mid s \in S_1] \\ = \iota_c \circ dom_c[g_s/s \mid s \in S_1] = \iota_c \circ dom_c[id_s^{Ini}/s \mid s \in S_1] = \iota_c \circ id_{dom_c}^{Ini} = \iota_c \end{aligned}$$

and thus $h_s \circ h'_s = id_s^{Ini}$. This finishes the proof that h is an isomorphism with inverse h'.

For all $c : dom \to s \in F' \setminus F$, $c^{Ini} = [C(s)^{Ini}) \circ \iota_c = h'_s \circ \iota_c = h^{-1} \circ \iota_c$. ❑

If SP' satisfies Def. 5.1(2), then the product of all destructors of $SP' \setminus SP$ is an isomorphism:

Theorem 6.6 *Let $SP' = (\Sigma', AX')$ be a swinging type with base type $SP = (\Sigma, AX)$ such that SP' satisfies 5.1(2). There is a Σ'-isomorphism h from the Σ'-structure B with equality and universe to the final object Fin of $Mod_{EU}(SP')$ that is defined as follows: Let $S_1 = S' \setminus S$ and for all $s \in S_1$, $D(s) = \{d \in F' \setminus F \mid s_d = s\}$.*

- $B|_\Sigma = Fin|_\Sigma$,
- *for all* $s \in S_1$, $s^B = \prod_{d \in D(s)} ran_d^{Fin}$,
- *for all* $d : s \to ran_d \in F' \setminus F$, $d^B = ran_d[\langle D(s)^{Fin}\rangle/s \mid s \in S_1] \circ \pi_d$.

Moreover, for all $d : s \to ran \in F' \setminus F$, $d^{Fin} = \pi_d \circ h^{-1}$.

Proof. Since SP satisfies 5.1(2), $B \in Mod_{EU}(SP)$ and thus there is a unique Σ'-homomorphism $h : B \to Fin$. The statement of the lemma follows by dualizing the proof of Theorem 6.5. ❏

7 Recursion and corecursion

Theorem 7.1 *Let* $SP' = (\Sigma', AX')$ *be a swinging type with base type* $SP = (\Sigma, AX)$ *such that* SP' *satisfies 5.1(3). The initial object of* $Mod_{EU}(SP)$ *can be extended to the initial object of* $Mod_{EU}(SP')$.

Proof. Let $A \in Mod_{EU}(SP)$ and $f \in F' \setminus F$. Using the notations of Def. 5.1(3) we define a Σ-structure A' as follows:

- for all $s \in S_1$, $s^{A'} = \prod_{f \in rec(s)} ran_f^A (= sub_1(s)^A)$,
- for all $s \in S_1$ and constructors $c : dom \to s$, $c^{A'} = \langle t_{f,c}^A \rangle_{f \in rec(s)}$,
- for all other symbols $s \in \Sigma$, $s^{A'} = s^A$.

Let Ini be initial in $Mod_{EU}(SP)$. Since SP satifies 5.1(1), $Ini' \in Mod_{EU}(SP)$. Since Ini is initial in $Mod_{EU}(SP)$, there is a unique Σ-homomorphism $h : Ini \to Ini'$. Hence for all $s \in S_1$ and constructors $c : dom \to s$,

$$h_s \circ c^{Ini} = c^{Ini'} \circ h_{dom} = \langle t_{f,c}^{Ini} \rangle_{f \in rec(s)} \circ h_{dom}. \tag{1}$$

Let $s \in S_1$. $f : s \to ran \in rec(s)$ can be interpreted in Ini as the composition $s^{Ini} \stackrel{h_s}{\to} s^{Ini'} \stackrel{\pi_f}{\to} ran^{Ini}$, i.e., $f^{Ini} = \pi_f \circ h_s$. Consequently,

$$h_s = \langle f^{Ini} \rangle_{f \in rec(s)} = \langle rec(s) \rangle^{Ini} \tag{2}$$

and thus by Proposition 4.4, for all constructors $c : dom \to s$,

$$f^{Ini} \circ c^{Ini} = \pi_f \circ h_s \circ c^{Ini} \stackrel{(1)}{=} \pi_f \circ \langle t_{f,c}^{Ini} \rangle_{f \in rec(s)} \circ h_{dom} = t_{f,c}^{Ini} \circ h_{dom} \stackrel{(2)}{=} t_{f,c}^{Ini} \circ sub_2^{\#}(dom)^{Ini},$$

i.e., Ini satisfies the equation

$$f \circ c \equiv t_{f,c} \odot sub_2^{\#}(dom) \tag{3}$$

of $AX' \setminus AX$ (see Def. 5.1(3)). To sum up, we have concluded the validity of $AX' \setminus AX$ in Ini from the fact that h is Σ-homomorphic.

It remains to show that Ini is initial in $Mod_{EU}(SP')$. So let $B \in Mod_{EU}(SP')$ and $A = B|_\Sigma$. Conversely to the preceding proof step, let us now conclude from the validity of $AX' \setminus AX$ in B that $h' : A \to A'$, defined by $\pi_f \circ h'_s = f^B$ for all $s \in S$, is Σ-homomorphic. Let $s \in S_1$ and $c : dom \to s$ be a constructor. Then

$$\pi_f \circ h'_s \circ c^A = f^B \circ c^B \stackrel{(3)}{=} t_{f,c}^B \circ sub_2^{\#}(dom)^B = \pi_f \circ \langle t_{f,c}^A \rangle_{f \in rec(s)} \circ sub_2^{\#}(dom)^B = \pi_f \circ c^{A'} \circ h'_{dom}$$

and thus $h'_s \circ c^A = c^{A'} \circ h'_{dom_c}$, i.e., h' is Σ-homomorphic.

Since $A \in Mod_{EU}(SP)$, there is a unique Σ-homomorphism $g : Ini \to A$. Define $g' : Ini' \to A'$ by $\pi_f \circ g'_s = g_{ran_f} \circ \pi_f$ for all $s \in S_1$ and $f \in rec(s)$ and by $g'_s = g_s$ for all $s \in S_0$. Let $s \in S_1$ and $c : dom \to s$ be a constructor. Since $t_{f,c} : sub_1^{\#}(dom) \to ran_f$ is a Σ-term, g is Σ-homomorphic and $g'_s = \prod_{f \in rec(s)} g_{ran_f} = g_{sub_1(s)}$, Proposition 3.5 implies

$$g_{ran_f} \circ t_{f,c}^{Ini} = t_{f,c}^A \circ g_{sub_1^{\#}(dom)} = t_{f,c}^A \circ g'_{dom} \tag{4}$$

and thus

$$\pi_f \circ g'_s \circ c^{Ini'} = g_{ran_f} \circ \pi_f \circ \langle t_{f,c}^{Ini} \rangle_{f \in rec(s)} = g_{ran_f} \circ t_{f,c}^{Ini} \stackrel{(4)}{=} t_{f,c}^A \circ g'_{dom} = \pi_f \circ \langle t_{f,c}^A \rangle_{f \in rec(s)} \circ g'_{dom} = \pi_f \circ c^{A'} \circ g'_{dom}.$$

Hence $g'_s \circ c^{Ini'} = c^{A'} \circ g'_{dom}$, i.e., g' is Σ-homomorphic.

Consequently, there are two Σ-homomorphisms from Ini to A': $h' \circ g$ and $g' \circ h$. Since Ini is initial in $Mod_{EU}(SP)$, they are equal. Hence for all $s \in S_1$ and $f \in rec(s)$,

$$f^B \circ g_s = \pi_f \circ h'_s \circ g_s = \pi_f \circ g'_s \circ h_s = g_{ran_f} \circ \pi_f \circ h_s = g_{ran_f} \circ f^{Ini},$$

i.e., g extends to a Σ'-homomorphism from Ini to B. Since each Σ'-homomorphism from Ini to B reduces to a Σ-homomorphism from Ini to A, Ini is initial in $Mod_{EU}(SP)$ and $S' = S$, g is the only Σ'-homomorphism from Ini to B. We conclude that Ini is initial in $Mod_{EU}(SP')$. ❏

Theorem 7.2 *Let $SP' = (\Sigma', AX')$ be a swinging type with base type $SP = (\Sigma, AX)$ such that SP' satisfies 5.1(4). The final object of $Mod_{EU}(SP)$ can be extended to the final object of $Mod_{EU}(SP')$.*

Proof. Let $A \in Mod_{EU}(SP)$. Using the notations of Def. 5.1(4) we define a Σ-structure A' as follows:

- for all $s \in S_1$, $s^{A'} = \coprod_{f \in cor(s)} dom_f^A (= sub_1(s)^A)$,
- for all $s \in S_1$ and destructors $d : s \to ran$, $d^{A'} = [t^A_{f,d}]_{f \in cor(s)}$,
- for all other symbols $s \in \Sigma$, $s^{A'} = s^A$.

Let Fin be final in $Mod_{EU}(SP)$. Since SP satifies 5.1(2), $Fin' \in Mod_{EU}(SP)$. Since Fin is final in $Mod_{EU}(SP)$, there is a unique Σ-homomorphism $h : Fin' \to Fin$. Hence for all $s \in S_1$ and destructors $d : s \to ran$,

$$d^{Fin} \circ h_s = h_{ran} \circ d^{Fin'} = h_{ran} \circ [t^{Fin}_{f,d}]_{f \in cor(s)}. \tag{1}$$

Let $s \in S_1$. $f : dom \to s \in cor(s)$ can be interpreted in Fin as the composition $dom^{Fin} \xrightarrow{\iota_f} s^{Fin'} \xrightarrow{h_s} s^{Fin}$, i.e., $f^{Fin} = h_s \circ \iota_f$. Consequently,

$$h_s = [f^{Fin}]_{f \in cor(s)} = [cor(s))^{Fin} \tag{2}$$

and thus by Proposition 4.4, for all destructors $d : s \to ran$,

$$d^{Fin} \circ f^{Fin} = d^{Fin} \circ h_s \circ \iota_f \overset{(1)}{=} h_{ran} \circ [t^{Fin}_{f,d}]_{f \in cor(s)} \circ \iota_f = h_{ran} \circ t^{Fin}_{f,d} \overset{(2)}{=} sub_2^*(ran)^{Fin} \circ t^{Fin}_{f,d},$$

i.e., Fin satisfies the equation

$$d \circ f \equiv sub_2^*(ran) \odot t_{f,d} \tag{3}$$

of $AX' \setminus AX$ (see Def. 5.1(4)). To sum up, we have concluded the validity of $AX' \setminus AX$ in Fin from the fact that h is Σ-homomorphic.

It remains to show that Fin is final in $Mod_{EU}(SP')$. So let $B \in Mod_{EU}(SP')$ and $A = B|_\Sigma$. Conversely to the preceding proof step, let us now conclude from the validity of $AX' \setminus AX$ in B that $h' : A' \to A$, defined by $h'_s \circ \iota_f = f^B$ for all $s \in S$, is Σ-homomorphic. Let $s \in S_1$ and $d : s \to ran$ be a destructor. Then

$$d^A \circ h'_s \circ \iota_f = d^B \circ f^B \overset{(3)}{=} sub_2^*(ran)^B \circ t^B_{f,d} = sub_2^*(ran)^B \circ [t^A_{f,d}]_{f \in cor(s)} \circ \iota_f = h'_{ran} \circ d^{A'} \circ \iota_f$$

and thus $d^A \circ h'_s = h'_{ran} \circ d^{A'}$, i.e., h' is Σ-homomorphic.

Since $A \in Mod_{EU}(SP)$, there is a unique Σ-homomorphism $g : A \to Fin$. Define $g' : A' \to Fin'$ by $g'_s \circ \iota_f = \iota_f \circ g_{dom_f}$ for all $s \in S_1$ and $f \in cor(s)$ and by $g'_s = g_s$ for all $s \in S_0$. Let $s \in S_1$ and $d : s \to ran$ be a destructor. Since $t_{f,d} : dom_f \to sub_1^*(ran)$ is a Σ-term, g is Σ-homomorphic and $g'_s = \coprod_{f \in cor(s)} g_{dom_f} = g_{sub_1(s)}$, Proposition 3.5 implies

$$t^{Fin}_{f,d} \circ g_{dom_f} = g_{sub_1^*(ran)} \circ t^A_{f,d} = g'_{ran} \circ t^A_{f,d} \tag{4}$$

and thus

$$d^{Fin'} \circ g'_s \circ \iota_f = [t^{Fin}_{f,d}]_{f \in cor(s)} \circ \iota_f \circ g_{dom_f} = t^{Fin}_{f,d} \circ g_{dom_f} \overset{(4)}{=} g'_{ran} \circ t^A_{f,d} = g'_{ran} \circ [t^A_{f,d}]_{f \in cor(s)} \circ \iota_f = g'_{ran} \circ d^{A'} \circ \iota_f.$$

Hence $d^{Fin'} \circ g'_s = g'_{ran} \circ d^{A'}$, i.e., g' is Σ-homomorphic.

Consequently, there are two Σ-homomorphisms from A' to Fin: $g \circ h'$ and $h \circ g'$. Since Fin is final in $Mod_{EU}(SP)$, they are equal. Hence for all $s \in S_1$ and $f \in cor(s)$,

$$g_s \circ f^B = g_s \circ h'_s \circ \iota_f = h_s \circ g'_s \circ \iota_f = h_s \circ \iota_f \circ g_{dom_f} = f^{Fin} \circ g_{dom_f},$$

i.e., g extends to a Σ'-homomorphism from B to Fin. Since each Σ'-homomorphism from B to Fin reduces to a Σ-homomorphism from A to Fin, Fin is final in $Mod_{EU}(SP)$ and $S' = S$, g is the only Σ'-homomorphism from B to Fin. We conclude that Fin is final in $Mod_{EU}(SP')$. ❏

8 Relations

Definition 8.1 *(μ- and ν-extensions)* Let $\Sigma' = (S_0, S, F', R')$ be a signature, $\Sigma = (S_0, S, F, R)$ be a subsignature of Σ', $SP = (\Sigma, AX)$, $SP' = (\Sigma', AX')$ be specifications with $AX \subseteq AX'$ such that $AX_1 =_{def} AX' \setminus AX$ consists of

(1) R_1-positive Horn clauses for $R_1 =_{def} (R' \setminus R) \cup \{\equiv_s \mid s \in S_1\}$ or

(2) restricted R_1-positive Horn clauses for $R_1 =_{def} (R' \setminus R) \cup \{all_s \mid s \in S_1\}$ or

(3) restricted R_1-positive co-Horn clauses for $R_1 =_{def} (R' \setminus R) \cup \{all_s \mid s \in S_1\}$ or

(4) R_1-positive co-Horn clauses for $R_1 =_{def} (R' \setminus R) \cup \{\equiv_s \mid s \in S_1\}$

where $S_1 = S \setminus S_0$. R_1 is called the set of **relations defined by** AX_1. In cases (1) and (2), SP' is a **μ-extension of** SP and thus R_1 is a set of predicates. In cases (3) and (4), SP' is a **ν-extension of** SP and thus R_1 is a set of copredicates (see Def. 5.1). ❏

Proposition 8.2 *Let $SP' = (\Sigma', AX')$ be a swinging type with base type $SP = (\Sigma, AX)$.*

If SP' is an abstraction, then SP' is a μ-extension of SP.

If SP' is a restriction, then SP' is a ν-extension of SP. ❏

We recapitulate the classical fixpoint theorems for monotone resp. continuous functions for complete lattices:

Definition and Theorem 8.3 *(fixpoints, continuity)* Let L be a complete lattice with partial order $\leq$, least element $\bot$ and greatest element $\top$. Given a set A, the lattice structure of L induces a lattice structure on the function space L^A as usual.

Let $F : L \to L$ be a function. $F^* =_{def} \sqcup_{i \in \mathbb{N}} F^i$ and $F_* =_{def} \sqcap_{i \in \mathbb{N}} F^i$.

$a \in L$ is F**-closed** if $F(a) \leq a$. a is F**-dense** if $a \leq F(a)$. a is a **fixpoint** of F if a is F-closed and F-dense. F is **monotone** if for all $a, b \in L$, $a \leq b$ implies $F(a) \leq F(b)$. F is **continuous** if for all increasing chains $a_0 \leq a_1 \leq a_2 \leq \ldots$ of elements of L, $F(\sqcup_{i \in \mathbb{N}} a_i) \leq \sqcup_{i \in \mathbb{N}} F(a_i)$. F is **cocontinuous** if for all decreasing chains $a_0 \geq a_1 \geq a_2 \geq \ldots$ of elements of L, $\sqcap_{i \in \mathbb{N}} F(a_i) \leq F(\sqcap_{i \in \mathbb{N}} a_i)$.

Let F be monotone.

(1) (Knaster-Tarski) lfp$(F) = \sqcap\{a \in L \mid F(a) \leq a\}$ is the least fixpoint of F and a superset of $F^*(\bot)$. gfp$(F) = \sqcup\{a \in L \mid a \leq F(a)\}$ is the greatest fixpoint of F and a subset of $F_*(\top)$.

(2) (Kleene) If F is continuous, then $F(F^*(\bot)) \leq F^*(\bot)$ and thus by (1), lfp$(F) = F^*(\bot)$. If F is cocontinuous, then $F_*(\top) \leq F(F_*(\top))$ and thus by (1), gfp$(F) = F_*(\top)$. ❏

The following lemma is fundamental for the most important proof rules for reasoning about extensions by predicates or copredicates, respectively (see Theorem 8.15).

Theorem 8.4 *Let L be a complete lattice, $F, G : L \to L$ be monotone functions and $a \in L$.*

(1) **Induction**. $lfp(F) \leq a$ if $F^n(a) \leq a$ for some $n > 0$.

(2) **Strong induction**. $lfp(F) \leq a$ if $F^n(a \sqcap lfp(F)) \leq a$ for some $n > 0$.

(3) **Coinduction**. $a \leq gfp(F)$ if $a \leq F^n(a)$ for some $n > 0$.

(4) **Strong coinduction**. $a \leq gfp(F)$ if $a \leq F^n(a \sqcup gfp(F))$ for some $n > 0$.

(5) **Extended strong coinduction**. Suppose that all $b \in L$ are G-dense and for all $b \in L$ and $n > 0$,

$$b \leq F^n(G^*(b)) \text{ implies } G^*(b) \leq F^n(G^*(b)). \qquad (5.1)$$

$a \leq gfp(F)$ if (5.2) $a \leq F^n(G^*(a \sqcup gfp(F)))$ for some $n > 0$.

Proof. (1) Let $F^n(a) \leq a$ for some $n > 0$, i.e., a is F^n-closed. Let $b =_{def} \sqcap_{i>0} F^i(a)$. Then

$$b \leq F^n(a) \leq a = F^0(a). \qquad (*)$$

Moreover, by the definition of b, $b \leq F^i(a)$ for all $i > 0$. Hence for all $i > 0$, $b \leq F^{i-1}(a)$ and thus $F(b) \leq F^i(a)$ because F is monotone. We conclude that $F(b)$ is a lower bound of $\{F^i(a) \mid i > 0\}$. Hence $F(b) \leq b$, i.e., b is F-closed. By Theorem 8.3(1), $lfp(F) = \sqcap\{c \in L \mid F(c) \leq c\}$ and thus $lfp(F) \leq b \leq a$ by $(*)$ and because $lfp(F)$ is the least F-closed element of L.

(2) Let $F^n(a \sqcap lfp(F)) \leq a$ for some $n > 0$. Then

$$\begin{aligned}
&F^n(a \sqcap lfp(F)) = F^n(a \sqcap lfp(F)) \sqcap F^n(a \sqcap lfp(F)) \\
&\leq F^n(a \sqcap lfp(F)) \sqcap F^n(lfp(F)) \quad \text{(since } F^n \text{ is monotone)} \\
&\leq a \sqcap F^n(lfp(F)) \quad \text{(by assumption)} \\
&\leq a \sqcap lfp(F). \quad \text{(since } lfp(F) \text{ is } F\text{- and thus } F^n\text{-closed because } F \text{ is monotone)}
\end{aligned}$$

Hence $a \sqcap lfp(F)$ is F^n-closed. Let $b =_{def} \sqcap_{i>0} F^i(a \sqcap lfp(F))$. Then

$$b \leq F^n(a \sqcap lfp(F)) \leq a \sqcap lfp(F) = F^0(a \sqcap lfp(F)). \qquad (*)$$

Moreover, by the definition of b, $b \leq F^i(a \sqcap lfp(F))$ for all $i > 0$. Hence for all $i > 0$, $b \leq F^{i-1}(a \sqcap lfp(F))$ and thus $F(b) \leq F^i(a \sqcap lfp(F))$ because F is monotone. We conclude that $F(b)$ is a lower bound of $\{F^i(a \sqcap lfp(F)) \mid i > 0\}$. Hence $F(b) \leq b$, i.e. b is F-closed. By Theorem 8.3(1), $lfp(F) = \sqcap\{c \in L \mid F(c) \leq c\}$ and thus $lfp(F) \leq b \leq a \sqcap lfp(F) \leq a$ by $(*)$ and because $lfp(F)$ is the least F-closed element of L.

(3) Let $a \leq F^n(a)$ for some $n > 0$, i.e., a is F^n-dense. Let $b =_{def} \sqcup_{i>0} F^i(a)$. Then

$$b \geq F^n(a) \geq a = F^0(a). \qquad (*)$$

Moreover, by the definition of b, $b \geq F^i(a)$ for all $i > 0$. Hence for all $i > 0$, $b \geq F^{i-1}(a)$ and thus $F(b) \geq F^i(a)$ because F is monotone. We conclude that $F(b)$ is an upper bound of $\{F^i(a) \mid i > 0\}$. Hence $b \leq F(b)$, i.e. b is F-dense. By Theorem 8.3(1), $gfp(F) = \sqcup\{c \in L \mid F(c) \geq c\}$ and thus $gfp(F) \geq b \geq a$ by $(*)$ and because $gfp(F)$ is the greatest F-dense element of L.

(4) Let $a \leq F^n(a \sqcup gfp(F)) \leq a$ for some $n > 0$. Then

$$\begin{aligned}
&F^n(a \sqcup gfp(F)) = F^n(a \sqcup gfp(F)) \sqcup F^n(a \sqcup gfp(F)) \\
&\geq F^n(a \sqcup gfp(F)) \sqcup F^n(gfp(F)) \quad \text{(since } F^n \text{ is monotone)} \\
&\geq a \sqcup F^n(gfp(F)) \quad \text{(by assumption)} \\
&\geq a \sqcup gfp(F). \quad \text{(since } gfp(F) \text{ is } F\text{- and thus } F^n\text{-dense because } F \text{ is monotone)}
\end{aligned}$$

Hence $a \sqcup \mathit{gfp}(F)$ is F^n-dense. Let $b =_{def} \sqcup_{i>0} F^i(a \sqcup \mathit{gfp}(F))$. Then

$$b \geq F^n(a \sqcup \mathit{gfp}(F)) \geq a \sqcup \mathit{gfp}(F) = F^0(a \sqcup \mathit{gfp}(F)). \qquad (*)$$

Moreover, by the definition of b, $b \geq F^i(a \sqcup \mathit{gfp}(F))$ for all $i > 0$. Hence for all $i > 0$, $b \geq F^{i-1}(a \sqcup \mathit{gfp}(F))$ and thus $F(b) \geq F^i(a \sqcup \mathit{gfp}(F))$ because F is monotone. We conclude that $F(b)$ is an upper bound of $\{F^i(a \sqcup \mathit{lfp}(F)) \mid i > 0\}$. Hence $b \leq F(b)$, i.e. b is F-dense. By Theorem 8.3(1), $\mathit{gfp}(F) = \sqcup\{c \in L \mid c \leq F(c)\}$ and thus $\mathit{gfp}(F) \geq b \geq a \sqcup \mathit{gfp}(F) \geq a$ by $(*)$ and because $\mathit{gfp}(F)$ is the greatest F-dense element of L.

(5) Let $a \leq F^n(G^*(a \sqcup \mathit{gfp}(F))) \leq a$ for some $n > 0$. Then

$F^n(G^*(a \sqcup \mathit{gfp}(F))) = F^n(G^*(a \sqcup \mathit{gfp}(F))) \sqcup F^n(G^*(a \sqcup \mathit{gfp}(F)))$
$\geq F^n(G^*(a \sqcup \mathit{gfp}(F))) \sqcup F^n(a \sqcup \mathit{gfp}(F))$ (since $a \sqcup \mathit{gfp}(F) \leq G^*(a \sqcup \mathit{gfp}(F))$ and F^n is monotone)
$\geq F^n(G^*(a \sqcup \mathit{gfp}(F))) \sqcup F^n(\mathit{gfp}(F))$ (since F^n is monotone)
$\geq a \sqcup F^n(\mathit{gfp}(F))$ (by assumption 5.2)
$\geq a \sqcup \mathit{gfp}(F)$. (since $\mathit{gfp}(F)$ is F- and thus F^n-dense because F is monotone)

By assumption 5.2, we conclude that $G^*(a \sqcup \mathit{gfp}(F))$ is F^n-dense. Let $b =_{def} \sqcup_{i>0} F^i(G^*(a \sqcup \mathit{gfp}(F)))$. Then

$$b \geq F^n(G^*(a \sqcup \mathit{gfp}(F))) \geq G^*(a \sqcup \mathit{gfp}(F)) = F^0(G^*(a \sqcup \mathit{gfp}(F))). \qquad (*)$$

Moreover, by the definition of b, $b \geq F^i(G^*(a \sqcup \mathit{gfp}(F)))$ for all $i > 0$. Hence for all $i > 0$, $b \geq F^{i-1}(G^*(a \sqcup \mathit{gfp}(F)))$ and thus $F(b) \geq F^i(G^*(a \sqcup \mathit{gfp}(F)))$ because F is monotone. We conclude that $F(b)$ is an upper bound of $\{F^i(G^*(a \sqcup \mathit{lfp}(F))) \mid i > 0\}$. Hence $b \leq F(b)$, i.e. b is F-dense. By Theorem 8.3(1), $\mathit{gfp}(F) = \sqcup\{c \in L \mid c \leq F(c)\}$ and thus $\mathit{gfp}(F) \geq b \geq G^*(a \sqcup \mathit{gfp}(F)) \geq a \sqcup \mathit{gfp}(F) \geq a$ by $(*)$ and because $\mathit{gfp}(F)$ is the greatest F-dense element of L. ❑

Lemma 8.5 (not used) *Let L be a complete lattice and $F, G : L \to L$ be monotone functions.*

(1) $\mathit{lfp}(F) \leq \mathit{lfp}(F \sqcup G)$.
(2) $\mathit{gfp}(F \sqcap G) \leq \mathit{gfp}(F)$.
(3) If $\mathit{lfp}(F) \sqcap \mathit{lfp}(G)$ *is F-closed, then* $\mathit{lfp}(F) \leq \mathit{lfp}(G)$.
(4) If $\mathit{gfp}(F) \sqcup \mathit{lfp}(G)$ *is F-dense, then* $\mathit{gfp}(G) \leq \mathit{gfp}(F)$.
(5) If $G(\mathit{lfp}(F \circ G))$ *is F-closed and* $G \leq id$*, then* $\mathit{lfp}(F) \leq \mathit{lfp}(F \circ G)$.
(6) If $G(\mathit{gfp}(F \circ G))$ *is F-dense and* $id \leq G$*, then* $\mathit{gfp}(F \circ G) \leq \mathit{gfp}(F)$.

Proof. (1) We show that lfp is a monotone function from the lattice $[L \to L]$ of monotone functions on L to L. $\leq$ and $\sqcup$ are lifted as usually from L to $[L \to L]$. Let $F', G' \in [L \to L]$ such that $F' \leq G'$. By Theorem 8.3(1),

$$\mathit{lfp}(F') = \sqcap\{a \in L \mid F'(a) \leq a\} \leq \sqcap\{a \in L \mid G'(a) \leq a\} = \mathit{lfp}(G').$$

Hence in particular, $\mathit{lfp}(F) \leq \mathit{lfp}(F \sqcup G)$.

(2) We show that gfp is a monotone function from the lattice $[L \to L]$ of monotone functions on L to L. $\leq$ and $\sqcap$ are lifted as usually from L to $[L \to L]$. Let $F', G' \in [L \to L]$ such that $F' \leq G'$. By Theorem 8.3(1),

$$\mathit{gfp}(F') = \sqcup\{a \in L \mid a \leq F'(a)\} \leq \sqcup\{a \in L \mid a \leq G'(a)\} = \mathit{gfp}(G').$$

Hence in particular, $\mathit{gfp}(F \sqcap G) \leq \mathit{gfp}(F)$.

(3) If $\mathit{lfp}(F) \sqcap \mathit{lfp}(G)$ is F-closed, then $\mathit{lfp}(F) = \sqcap\{a \in L \mid F(a) \leq a\} \leq \mathit{lfp}(F) \sqcap \mathit{lfp}(G) \leq \mathit{lfp}(G)$.

(4) If $\mathit{gfp}(F) \sqcup \mathit{lfp}(G)$ is F-dense, then $\mathit{gfp}(G) \leq \mathit{gfp}(F) \sqcup \mathit{lfp}(G) \leq \sqcup\{a \in L \mid F(a) \leq a\} = \mathit{gfp}(F)$.

(5) If $G(\mathit{lfp}(F \circ G))$ is F-closed and $G \leq id$, $\mathit{lfp}(F) = \sqcap\{a \in L \mid F(a) \leq a\} \leq G(\mathit{lfp}(F \circ G)) \leq \mathit{lfp}(F \circ G)$.

(6) If $G(gfp(F \circ G))$ is F-dense and $id \leq G$, $gfp(F \circ G) \leq G(gfp(F \circ G)) \leq \sqcup\{a \in L \mid F(a) \leq a\} = gfp(F)$.❑

Given the assumptions of Def. 8.1 and an $(SP \cup F')$-model A, the class $Mod(\Sigma', A)$ of Σ'-structures over A forms a complete lattice: The partial order $\leq$ and the corresponding least element $\bot$, greatest element $\top$, suprema and infima are defined as follows. For all $B, C \in Mod(\Sigma', A)$,

$$B \leq C \iff \forall\, r \in R_1 : r^B \subseteq r^C.$$

For all $r : s \in R_1$ and $\mathcal{B} \subseteq Mod(\Sigma', A)$, $r^\bot = \emptyset$, $r^\top = s^A$, $r^{\sqcup\mathcal{B}} = \bigcup_{B\in\mathcal{B}} r^B$ and $r^{\sqcap\mathcal{B}} = \bigcap_{B\in\mathcal{B}} r^B$. Moreover, if R_1 is an S-sorted set of binary relations $r_s : s \times s$, then for all $B, C \in Mod(\Sigma', A)$, $B \cdot C \in Mod(\Sigma', A)$ is defined as follows: For all $r \in R_1$, $r^{B\cdot C} = r^B \cdot r^C$ (see Section 1).

The monotone function on $Mod(\Sigma', A)$ we are interested in here is the functor $\_|_\sigma$ induced by the signature morphism σ that maps each $r \in R_1$ to the AX_1-*definition* of r (see Definition 4.5):

Definition and Proposition 8.6 *(AX-definition)* Let $\Sigma = (S_0, S, F, R)$ be a signature, AX be a finite set of either only Horn or only co-Horn clauses over Σ, A be a Σ-structure and $r : s \in R$.

(1) Let $AX_r = \{(r \circ t_i \Leftarrow \varphi_i) : s_i\}_{i=1}^n$ be the set of Horn clauses for r among the clauses of AX. The Σ-formula

$$\varphi_{r,AX}(x) \quad =_{def} \quad \bigvee_{i=1}^{n} \exists i(x \equiv t_i(i) \wedge \varphi_i) : s$$

is called the AX**-definition of** r. *A satisfies AX_r iff A satisfies $r(x) \Leftarrow \varphi_{r,AX}(x)$.*

(2) Let $AX_r = \{(r \circ t_i \Rightarrow \varphi_i) : s_i\}_{i=1}^n$ be the set of co-Horn clauses for r among the clauses of AX. The Σ-formula

$$\varphi_{r,AX}(x) \quad =_{def} \quad \bigwedge_{i=1}^{n} \forall i(\neg x \equiv t_i(i) \vee \varphi_i) : s$$

is called the AX**-definition of** r. *A satisfies AX_r iff A satisfies $r(x) \Rightarrow \varphi_{r,AX}(x)$.*

Proof. Proposition 4.7 and a couple of simple logical transformations. ❑

Definition 8.7 *(step functor)* Let the assumptions of Def. 8.1 hold true. The signature morphism $\sigma : \Sigma' \to \Sigma'$ that is the identity on Σ and maps each relation r defined by SP' wrt SP to φ_{r,AX_1} is called the **relation transformer defined by** $AX' \setminus AX$.

Let $\Sigma_1 = (S_0, S, F', R)$ and A be a Σ_1-structure. The (A, σ)**-step functor** maps each $B \in Mod(\Sigma', A)$ to $B|_\sigma$. ❑

Proposition 8.8 *Let the assumptions of Def. 8.7 hold true and F be the (A, σ)-step functor.*

(1) For all $B \in Mod(\Sigma', A)$, $\varphi \in Form_{\Sigma'}$ and $i \in \mathbb{N}$, $r^{F^i(B)} = \sigma^i(r)^B$ and $r^{F^*(B)} = \sigma^*(r)^B$.

(2) Let SP' be a μ-extension of SP. $B \in Mod(\Sigma', A)$ satisfies AX_1 iff for all $r \in R_1$, B satisfies $r \Leftarrow \sigma(r)$, iff B is F-closed.

(3) Let SP' be a ν-extension of SP $B \in Mod(\Sigma', A)$ satisfies AX_1 iff for all $r \in R_1$, B satisfies $r \Rightarrow \sigma(r)$, iff B is F-dense.

(4) Let SP' be a μ- or ν-extension of SP. B is a fixpoint of F iff for all $r \in R_1$, $B \in Mod(\Sigma', A)$ satisfies $r \Leftrightarrow \sigma(r)$, iff for all Σ'-formulas φ, B satisfies $\varphi \Leftrightarrow \sigma(\varphi)$.

Proof. (1) follows by induction on i: Assume that for all $B \in Mod(\Sigma', A)$ and $\varphi \in Form_{\Sigma'}$, $r^{F^i(B)} = \sigma^i(r)^B$. Then $r^{F^{i+1}(B)} = r^{F(F^i(B))} = r^{F^i(B)|_\sigma)} = \sigma(r)^{F^i(B)} = \sigma^i(\sigma(r))^B = \sigma^{i+1}(r)^B$.

(2) follows from Propositions 8.6 and 4.7(4). (3) follows from Propositions 8.6 and 4.7(4). (4) follows from Proposition 4.7(4). ❑

Moreover, given a fixpoint B of $\Phi_{A,\sigma}$, $\overline{r}^B$ is the B-complement of r^B if the AX_1-definition of $\overline{r}$ is the negation of the AX_1-definition of r:

Proposition 8.9 *Let the assumptions of Def. 8.7 hold true, F be the (A,σ)-step functor and B be a fixpoint of F. Suppose that for each relation $r : s$ defined by SP' wrt SP there is a relation $\overline{r} : s$ defined by AX_1 such that B satisfies $\sigma(\overline{r}) \Leftrightarrow \neg\sigma(r)$. Then $\overline{r}^B$ is the B-complement of r^B (see Def. 3.6).*

Proof. Let $r : s$ be a relation defined by AX_1. Since B is a fixpoint of F, we obtain

$$\overline{r}^B \;=\; \sigma(\overline{r})^B \;=\; (\neg\sigma(r))^B \;=\; s^B \setminus \sigma(r)^B \;=\; s^B \setminus r^B$$

by Proposition 8.8(4) and the assumption. ❏

Example 8.10 The parameter type TRIV(s)[BOOL] (see Example 5.4) is extended by binary relations on s:

ORD(s)[BOOL] `where` ORD(s) = TRIV(s) `and`

`preds`	$<, >, \leq, \geq, \not\equiv : s \times s$
`vars`	$x, y : s$
`axioms`	$x < y \iff y > x$
	$x \leq y \iff \neg x > y$
	$x \geq y \iff \neg x < y$
	$x \not\equiv y \iff \neg x \equiv y$

The following swinging type extends LIST[ORD(s)[BOOL]] (see Example 14.2) by the relations *sorted* and $\in$ for list membership and their complements *unsorted* and $\notin$:

COMPL[ORD(s)[BOOL]] `where` COMPL = LIST `and`

`preds`	$\in, \notin : s \times list(s)$
	$sorted, unsorted : list(s)$
`vars`	$x, y : s \quad L, L' : list(s)$
`axioms`	$x \in y : L \;\Leftarrow\; x \equiv y \vee x \in L$
	$sorted([])$
	$sorted(x : [])$
	$sorted(x : y : L) \;\Leftarrow\; x \leq y \wedge sorted(y : L)$
	$x \notin y : L \;\Rightarrow\; x \not\equiv y \wedge x \notin L$
	$unsorted([]) \;\Rightarrow\; False$
	$unsorted(x : []) \;\Rightarrow\; False$
	$unsorted(x : y : L) \;\Rightarrow\; x > y \vee unsorted(y : L)$

Let σ be the relation transformer defined by the axioms for COMPL\LIST. In fact, $\sigma(unsorted)$ and $\sigma(\notin)$ are the negations of $\sigma(sorted)$ and $\sigma(\in)$, respectively (see Def. 8.6). Hence for all LIST[TRIV(s)[BOOL]]-models A and fixpoints B of $\Phi_{A,\sigma}$, $unsorted^B$ and $\notin^B$ are the B-complements of $sorted^B$ and $\in^B$, respectively. ❏

Definition 8.11 *(monotone, (co)compact formula)* Given the assumptions of Def. 8.7, a Σ'-formula φ is **Σ'-monotone over** A if for all $B, C \in Mod(\Sigma', A)$,

$$B \leq C \quad \text{implies} \quad \varphi^B \subseteq \varphi^C. \tag{1}$$

φ is **Σ'-compact over** A if for all increasing chains $B_0 \leq B_1 \leq B_2 \leq \ldots$ of $Mod(\Sigma', A)$,

$$\varphi^{\sqcup_{i\in\mathbb{N}} B_i} \quad \subseteq \quad \bigcup_{i\in\mathbb{N}} \varphi^{B_i}. \tag{2}$$

φ is Σ'**-cocompact over** A if for all decreasing chains $B_0 \geq B_1 \geq B_2 \geq \dots$ of $Mod(\Sigma', A)$,

$$\bigcap_{i\in\mathbb{N}} \varphi^{B_i} \quad \subseteq \quad \varphi^{\sqcap_{i\in\mathbb{N}} B_i}. \quad ❏ \tag{3}$$

Proposition 8.12 *Let the assumptions of Def. 8.7 hold true. $\Phi_{A,\sigma}$ is monotone iff the premises of all Horn clauses and the conclusions of all co-Horn clauses of AX_1 are Σ'-monotone over A. $\Phi_{A,\sigma}$ is continuous over A iff the premises of all Horn clauses AX_1 are Σ'-compact over A. $\Phi_{A,\sigma}$ is cocontinuous over A iff the conclusions of all co-Horn clauses AX_1 are Σ'-cocompact over A.* ❏

Theorems 8.13 and 9.1 given below are shown by *transfinite induction* on ordinal numbers. Remember the principle of transfinite induction: A property P holds true for all ordinals if for all ordinals β, if $P(\beta)$ can be concluded from the assumption that P holds true for all ordinals $\alpha < \beta$. The correctness of transfinite induction rule follows from the fact that ordinal numbers form a well-ordered set $\mathcal{O}$, i.e., $\mathcal{O}$ is a totally ordered set such that each nonempty subset M of $\mathcal{O}$ has a least element (see [106], §13). The least element of the entire set $\mathcal{O}$ is denoted by 0. An ordinal is either 0, a *successor ordinal* β, i.e., β has an immediate predecessor α w.r.t. $<$, or a *limit ordinal* denoted by $sup(M)$ where M is the set of all predecessors of $sup(M)$ w.r.t. $<$.[10]

For proving Theorems 8.13 and 9.1 by transfinite induction, we use the following $\mathcal{O}$-sorted set M of pairs (φ, a) consisting of an R_1-positive Σ'-formula $\varphi : s$ and some $a \in s^A$: Let I be a nonempty set.

- For all Σ-atoms $\varphi : s$ and $a \in s^A$, $(\varphi, a) \in M_0$.
- Let $(\varphi, a) \in M_\alpha$ and β be the least ordinal that is greater than α. Then $(\neg\varphi, a) \in M_\beta$.
- Let $a \in \prod_{i\in\cup\{I_j|j\in J\}} s_i^A$, for all $j \in J$, $(\varphi_j : \prod_{i\in I_j} s_i, \pi_{I_j}(a)) \in M_{\alpha_j}$ and β be the least ordinal that is greater than α_j, $j \in J$. Then $(\bigwedge \varphi_j, a), (\bigvee \varphi_j, a) \in M_\beta$.
- Let $k \in I$, $a \in \prod_{i\in I\setminus\{k\}} s_i^A$, for all $b \in s_k^A$, $(\varphi : \prod_{i\in I} s_i, a *_k b) \in M_{\alpha_b}$ and β be the least ordinal that is greater than all α_b, $b \in s_k^A$. Then $(\forall k\varphi, a), (\exists k\varphi, a) \in M_\beta$.

Theorem 8.13 *Given the assumptions of Def. 8.7, R_1-positive Σ'-formulas are Σ'-monotone over A.*

Proof. Let $B, C \in Mod(\Sigma', A)$ with $B \leq C$. We show 8.11(1) for all R_1-positive Σ'-formulas φ. Let $(\varphi, a) \in M_\alpha$, $a \in \varphi^B$ and I be a nonempty set.

- Let $\varphi = r(t)$ be a Σ'-atom. If $r \in R$, then φ is a Σ-formula and thus $a \in \varphi^B = \varphi^A = \varphi^C$. If $r \in R_1$, then $a \in \varphi^B$ implies $t^A(a) \in r^B \subseteq r^C$ because $B \leq C$. Hence $a \in \varphi^C$.
- Let $\varphi = \neg\psi$ for some Σ-atom $\psi : s$. Hence $\psi^B = \psi^A = \psi^C$ and thus $a \in \varphi^B = s^A \setminus \psi^B = s^A \setminus \psi^C = \varphi^C$.
- Let $\varphi = \psi \wedge \vartheta$ (resp. $\varphi = \psi \vee \vartheta$). Then there are $\beta < \alpha$ and $\gamma < \alpha$ such that $(\varphi, \pi_I(a)) \in M_\beta$ and $(\psi, \pi_J(a)) \in M_\gamma$. $a \in \varphi^B$ implies $\pi_I(a) \in \psi^B$ and (resp. or) $\pi_J(a) \in \vartheta^B$. Hence by induction hypothesis, $\pi_I(a) \in \psi^C$ and $\pi_J(a) \in \vartheta^C$ and thus $a \in \varphi^C$.
- Let $\varphi = \forall k\psi$ (resp. $\varphi = \exists k\psi$) for some $k \in I$. Then for all $b \in s_k^A$ there is $\alpha_b < \beta$ such that $(\psi, a *_k b) \in M_{\alpha_b}$. $a \in \varphi^B$ implies $a *_k b \in \psi^B$ for all (resp. some) $b \in s_k^A$. Hence by induction hypothesis, $a *_k b \in \psi^C$ and thus $a \in \varphi^C$. ❏

Proposition 8.8 and Theorems 8.13 and 8.3 (Knaster-Tarski) immediately imply:

Theorem 8.14 *(fixpoint semantics of μ- and ν-extensions) Let the assumptions of Def. 8.7 hold true and F be the (A, σ)-step functor.*

- *If SP' is a μ-extension of SP, then F has a least fixpoint, which agrees with the least SP'-model over A (with respect to $\leq$).*

[10] 0, β and $sup(M)$ are usually interpreted as follows: $0 = \emptyset$, $\beta = \alpha \cup \{\alpha\}$ and $sup(M) = \cup M$. Consequently, $<$ is strict set inclusion and thus $\mathcal{O}$ is well-ordered.

- *If SP' is a ν-extension of SP, then F has a greatest fixpoint, which agrees with the greatest SP'-model over A.* ❑

Let us reformulate Theorem 8.4 in terms of a μ- or ν-extension $SP' = (\Sigma', AX')$ and the lattice $Mod(\Sigma', A)$:

Theorem 8.15 *Let the assumptions of Def. 8.7 hold true, SP' be a μ- or ν-extension of SP, σ be the relation transformer defined by $AX' \setminus AX$ and F be the (A,σ)-step functor. Moreover, let $\tau, \tau_1, \tau_2 : \Sigma' \to \Sigma'$ be signature morphisms such that for all $r \in \Sigma$, $\tau(r) = r$, and for all $r \in R_1$, $\tau_1(r) = \tau(r) \wedge r$ and $\tau_2(r) = \tau(r) \vee r$.*

(1) **Induction**. $lfp(F)$ satisfies $\bigwedge_{r\in R_1}(r \Rightarrow \tau(r))$ if $lfp(F)$ satisfies $\bigwedge_{r\in R_1}(\tau(\sigma^n(r)) \Rightarrow \tau(r))$ for some $n > 0$.

(2) **Strong induction**.
$lfp(F)$ satisfies $\bigwedge_{r\in R_1}(r \Rightarrow \tau(r))$ if $lfp(F)$ satisfies $\bigwedge_{r\in R_1}(\tau_1(\sigma^n(r)) \Rightarrow \tau(r))$ for some $n > 0$.

(3) **Coinduction**.
$gfp(F)$ satisfies $\bigwedge_{r\in R_1}(\tau(r) \Rightarrow r)$ if $gfp(F)$ satisfies $\bigwedge_{r\in R_1}(\tau(r) \Rightarrow \tau(\sigma^n(r)))$ for some $n > 0$.

(4) **Strong coinduction**.
$gfp(F)$ satisfies $\bigwedge_{r\in R_1}(\tau(r) \Rightarrow r)$ if $gfp(F)$ satisfies $\bigwedge_{r\in R_1}(\tau(r) \Rightarrow \tau_2(\sigma^n(r)))$ for some $n > 0$.

(5) **Extended strong coinduction**. Let $\gamma : \Sigma' \to \Sigma'$ be a further signature morphism[11] such that for all $r \in \Sigma_1$, $\gamma(r) = r$ or r is binary and

$$\gamma(r) \quad = \quad r \ \vee \ \equiv \ \vee \ r \circ \langle \pi_2, \pi_1 \rangle \ \vee \ \exists 3(r \circ \langle \pi_1, \pi_3 \rangle \wedge r \circ \langle \pi_3, \pi_2 \rangle).$$

$gfp(F)$ satisfies $\bigwedge_{r\in R_1}(\tau(r) \Rightarrow r)$ if $gfp(F)$ satisfies $\bigwedge_{r\in R_1}(\tau(r) \Rightarrow \gamma^*(\tau_2(\sigma^n(r))))$ for some $n > 0$.

Proof. (1) Suppose that for some $n > 0$ and all $r \in R_1$, $lfp(F)$ satisfies $\tau(\sigma^n(r)) \Rightarrow \tau(r)$, i.e., $\tau(\sigma^n(r))^{lfp(F)} \subseteq \tau(r)^{lfp(F)}$. Let B be the τ-reduct of $lfp(F)$. By the definition of F and Proposition 4.7(3), $r^{F^n(B)} = r^{B|_{\sigma^n}} = \sigma^n(r)^B = \tau(\sigma^n(r))^{lfp(F)} \subseteq \tau(r)^{lfp(F)} = r^B$. Hence $F^n(B) \leq B$. By Theorem 8.4(1), $lfp(F) \leq B$ and thus $r^{lfp(F)} \subseteq r^B = \tau(r)^{lfp(F)}$, i.e., $lfp(F)$ satisfies $r \Rightarrow \tau(r)$.

(2) Suppose that for some $n > 0$ and all $r \in R_1$, $lfp(F)$ satisfies $\tau_1(\sigma^n(r)) \Rightarrow \tau(r)$, i.e., $\tau_1(\sigma^n(r))^{lfp(F)} \subseteq \tau(r)^{lfp(F)}$. Let B be the τ-reduct of $lfp(F)$. By the definition of F and Proposition 4.7(3), $r^{F^n(B\sqcap lfp(F))} = r^{(B\sqcap lfp(F))|_{\sigma^n}} = \sigma^n(r)^{B\sqcap lfp(F)} = \sigma^n(r)^B \cap \sigma^n(r)^{lfp(F)} = \tau(\sigma^n(r))^{lfp(F)} \cap \sigma^n(r)^{lfp(F)} = (\tau(\sigma^n(r)) \wedge \sigma^n(r))^{lfp(F)} = \tau_1(\sigma^n(r))^{lfp(F)} \subseteq \tau(r)^{lfp(F)} = r^B$. Hence $F^n(B \sqcap lfp(F)) \leq B$. By Theorem 8.4(2), $lfp(F) \leq B$ and thus $r^{lfp(F)} \subseteq r^B = \tau(r)^{lfp(F)}$, i.e., $lfp(F)$ satisfies $r \Rightarrow \tau(r)$.

(3) Suppose that for some $n > 0$ and all $r \in R_1$, $gfp(F)$ satisfies $\tau(r) \Rightarrow \tau(\sigma^n(r))$, i.e., $\tau(r)^{gfp(F)} \subseteq \tau(\sigma^n(r))^{gfp(F)}$. Let B be the τ-reduct of $gfp(F)$. By Proposition 4.7(3), $r^B = \tau(r)^{gfp(F)} \subseteq \tau(\sigma^n(r))^{gfp(F)} = \sigma^n(r)^B = r^{B|_{\sigma^n}} = r^{F^n(B)}$. Hence $B \leq F^n(B)$. By Theorem 8.4(3), $B \leq gfp(F)$ and thus $\tau(r)^{gfp(F)} = r^B \subseteq r^{gfp(F)}$, i.e., $gfp(F)$ satisfies $\tau(r) \Rightarrow r$.

(4) Suppose that for some $n > 0$ and all $r \in R_1$, $gfp(F)$ satisfies $\tau(r) \Rightarrow \tau_2(\sigma^n(r))$, i.e., $\tau(r)^{gfp(F)} \subseteq \tau_2(\sigma^n(r))^{gfp(F)}$. Let B be the τ-reduct of $gfp(F)$. By Proposition 4.7(3) and the definition of F, $r^B = \tau(r)^{gfp(F)} \subseteq \tau_2(\sigma^n(r))^{gfp(F)} = (\tau(\sigma^n(r)) \vee \sigma^n(r))^{gfp(F)} = \tau(\sigma^n(r))^{gfp(F)} \cup \sigma^n(r)^{gfp(F)} = \sigma^n(r)^B \cup \sigma^n(r)^{gfp(F)} = \sigma^n(r)^{B\sqcup gfp(F)} = r^{(B\sqcup gfp(F))|_{\sigma^n}} = r^{F^n(B\sqcup gfp(F))}$. Hence $B \leq F^n(B \sqcup gfp(F))$. By Theorem 8.4(4), $B \leq gfp(F)$ and thus $\tau(r)^{gfp(F)} = r^B \subseteq r^{gfp(F)}$, i.e., $gfp(F)$ satisfies $\tau(r) \Rightarrow r$.

(5) Suppose that for some $n > 0$ and all $r \in R_1$, $gfp(F)$ satisfies $\tau(r) \Rightarrow \gamma^*(\tau_2(\sigma^n(r)))$, i.e., $\tau(r)^{gfp(F)} \subseteq \gamma^*(\tau_2(\sigma^n(r)))^{gfp(F)}$. Let B, C be the τ-reducts of $gfp(F)$ and $G^*(gfp(F))$, respectively, and G be the (A,γ)-step functor. By Propositions 4.7(3) and 8.8(1), $r^B = \tau(r)^{gfp(F)} \subseteq \gamma^*(\tau_2(\sigma^n(r)))^{gfp(F)} = \tau_2(\sigma^n(r))^{G^*(gfp(F))} = (\tau(\sigma^n(r)) \vee \sigma^n(r))^{G^*(gfp(F))} = \tau(\sigma^n(r))^{G^*(gfp(F))} \cup \sigma^n(r)^{G^*(gfp(F))} = \sigma^n(r)^C \cup \sigma^n(r)^{G^*(gfp(F))} = \sigma^n(r)^{C\sqcup G^*(gfp(F))} = ???\sigma^n(r)^{G^*(B)\sqcup G^*(gfp(F))} \subseteq r^{F^n(G^*(B\sqcup gfp(F)))}$. Hence $B \leq F^n(B \sqcup G^*(gfp(F)))$. By Theorem 8.4(5), $B \leq gfp(F)$ and thus $\tau(r)^{gfp(F)} = r^B \subseteq r^{gfp(F)}$, i.e., $gfp(F)$ satisfies $\tau(r) \Rightarrow r$. ❑

[11] γ is the relation transformer defined by axioms for the equivalence closure of r.

Here are some alternative axiomatizations of relations that satisfy the induction or coinduction assumption of Theorem 8.15. [39]

Corollary 8.16 *Let the assumptions of Theorem 8.15 hold true.*

(1) *Suppose that SP_1 and SP_2 are μ-extensions of SP, $R_2 = R_1 \cup \{r' : s \mid r : s \in R_1\}$ and*

$$AX_2 = \{r \Leftarrow (\sigma_1(r) \wedge r') \mid r \in R_1\} \cup \{r' \Leftarrow \sigma_1(r)[r'/r|r \in R_1] \mid r \in R_1\}.$$

Then $lfp(\Phi_{A,\sigma_1}) \leq lfp(\Phi_{A,\sigma_2})$.

(2) *Suppose that SP_1 and SP_2 are ν-extensions of SP, $R_2 = R_1 \cup \{r' : s \mid r : s \in R_1\}$ and*

$$AX_2 = \{r \Rightarrow (\sigma_1(r) \vee r') \mid r \in R_1\} \cup \{r' \Rightarrow \sigma_1(r)[r'/r|r \in R_1] \mid r \in R_1\}.$$

Then $gfp(\Phi_{A,\sigma_2}) \leq gfp(\Phi_{A,\sigma_1})$.

(3) *Suppose that SP_1 and SP_2 are μ-extensions of SP, R_1 is an S-sorted set of binary relations $\sim_s : s \times s$, $R_2 = R_1 \cup \{\approx_s : s \times s \mid s \in S\}$,*

$$AX_1 = \{f(x) \sim_{s'} f(y) \Leftarrow x \sim_s y \mid f : s \to s' \in F\},$$
$$AX_2 = \{\sim_s \Leftarrow \approx_s \cdot \sigma_1(\sim_s) \cdot \approx_s \ \mid s \in S\} \cup \{\approx_s \Leftarrow \sigma_1(\sim_s)[\approx_s / \sim_s |s \in S] \mid s \in S\}$$

and $lfp(\Phi_{A,\sigma_1}) \cdot lfp(\Phi_{A,\sigma_1}) \leq lfp(\Phi_{A,\sigma_1})$. Then $lfp(\Phi_{A,\sigma_1}) \leq lfp(\Phi_{A,\sigma_2})$.

(4) *Suppose that SP_1 and SP_2 are ν-extensions of SP, R_1 is an S-sorted set of binary relations $\sim_s : s \times s$, $R_2 = R_1 \cup \{\approx_s : s \times s \mid s \in S\}$,*

$$AX_1 = \{x \sim_s y \Rightarrow f(x) \sim_{s'} f(y) \mid f : s \to s' \in F\},$$
$$AX_2 = \{\sim_s \Rightarrow \approx_s \cdot \sigma_1(\sim_s) \cdot \approx_s \ \mid s \in S\} \cup \{\approx_s \Rightarrow \sigma_1(\sim_s)[\approx_s / \sim_s |s \in S] \mid s \in S\}$$

and $gfp(\Phi_{A,\sigma_1}) \cdot gfp(\Phi_{A,\sigma_1}) \leq gfp(\Phi_{A,\sigma_1})$. Then $gfp(\Phi_{A,\sigma_2}) \leq gfp(\Phi_{A,\sigma_1})$.

Proof. (1) Let $lfp_i = lfp(\Phi_{\sigma_i,A})$, $i = 1, 2$. Define a function $F : Mod(\Sigma_1, A) \to Mod(\Sigma_1, A)$ by $F(B) = B \sqcap lfp_1$. Let $B \in Mod(\Sigma_1, A)$ and $r \in R_1$. The construction of AX_2 from AX_1 implies $\sigma_2(r)^B = \sigma_1(r)^B \cap r^{lfp_1}$. Hence

$$r^{\sigma_2(B)} = \sigma_2(r)^B = \sigma_1(r)^B \cap r^{lfp_1} = \sigma_1(r)^B \cap r^{\sigma_1(lfp_1)} = \sigma_1(r)^B \cap \sigma_1(r)^{lfp_1} = \sigma_1(r)^{F(B)} = r^{\sigma_1(F(B))},$$

i.e., $\sigma_2 = \sigma_1 \circ F$. By Lemma 8.5(3), it remains to show that $lfp_1 \sqcap lfp_2$ is σ_1-closed. Since σ_1 is monotone, $\sigma_1(lfp_1 \sqcap lfp_2) \leq \sigma_1(lfp_1) = lfp_1$. Moreover, $\sigma_1(lfp_1 \sqcap lfp_2) = \sigma_1(F(lfp_2)) = \sigma_2(lfp_2) = lfp_2$. Hence $\sigma_1(lfp_1 \sqcap lfp_2) \leq lfp_1 \sqcap lfp_2$.

(2) Let $gfp_i = gfp(\Phi_{\sigma_i,A})$, $i = 1, 2$. Define a function $F : Mod(\Sigma_1, A) \to Mod(\Sigma_1, A)$ by $F(B) = B \sqcup gfp_1$. Let $B \in Mod(\Sigma_1, A)$ and $r \in R_1$. The construction of AX_2 from AX_1 implies $\sigma_2(r)^B = \sigma_1(r)^B \cup r^{gfp_1}$. Hence

$$r^{\sigma_2(B)} = \sigma_2(r)^B = \sigma_1(r)^B \cup r^{gfp_1} = \sigma_1(r)^B \cup r^{\sigma_1(gfp_1)} = \sigma_1(r)^B \cup \sigma_1(r)^{gfp_1} = \sigma_1(r)^{F(B)} = r^{\sigma_1(F(B))},$$

i.e., $\sigma_2 = \sigma_1 \circ F$. By Lemma 8.5(4), it remains to show that $gfp_1 \sqcup gfp_2$ is σ_1-dense. Since σ_1 is monotone, $gfp_1 = \sigma_1(gfp_1) \leq \sigma_1(gfp_1 \sqcup gfp_2)$. Moreover, $gfp_2 = \sigma_2(gfp_2) = \sigma_1(F(gfp_2)) = \sigma_1(gfp_1 \sqcup gfp_2)$. Hence $gfp_1 \sqcup gfp_2 \leq \sigma_1(gfp_1 \sqcup gfp_2)$.

(3) Let $lfp_i = lfp(\Phi_{\sigma_i,A})$, $i = 1, 2$.

(4) Let $gfp_i = gfp(\Phi_{\sigma_i,A})$, $i = 1, 2$. Define a function $F : Mod(\Sigma_1, A) \to Mod(\Sigma_1, A)$ by $F(B) = gfp_1 \cdot B \cdot gfp_1$. Let $B \in Mod(\Sigma_1, A)$ and $s \in S$. The construction of AX_2 from AX_1 implies $\sigma_2(\sim_s)^B = \sim_s^{gfp_1} \cdot \sigma_1(\sim_s)^B \cdot \sim_s^{gfp_1}$. Hence

$$\begin{aligned}\sim_s^{\sigma_2(B)} &= \sigma_2(\sim_s)^B = \sim_s^{gfp_1} \cdot \sigma_1(\sim_s)^B \cdot \sim_s^{gfp_1} = \sim_s^{\sigma_1(gfp_1)} \cdot \sigma_1(\sim_s)^B \cdot \sim_s^{\sigma_1(gfp_1)} \\ &= \sigma_1(\sim_s)^{gfp_1} \cdot \sigma_1(\sim_s)^B \cdot \sigma_1(\sim_s)^{gfp_1} = \sigma_1(\sim_s)^{F(B)} = \sim_s^{\sigma_1(F(B))},\end{aligned}$$

i.e., $\sigma_2 = \sigma_1 \circ F$. By Lemma 8.5(6), it remains to show that $F(gfp_2)$ is σ_1-dense. Let $s \in S$. Since $\sigma_2 = \sigma_1 \circ F$ and $gfp_1 \cdot gfp_1 \leq gfp_1$,

$$\begin{aligned}
\sim_s^{F(gfp_2)} &= \sim_s^{gfp_1} \cdot \sim_s^{gfp_2} \cdot \sim_s^{gfp_1} = \sim_s^{gfp_1} \cdot \sim_s^{\sigma_2(gfp_2)} \cdot \sim_s^{gfp_1} \\
&= \sim_s^{gfp_1} \cdot \sigma_1(\sim_s)^{F(gfp_2)} \cdot \sim_s^{gfp_1} = \sim_s^{gfp_1} \cdot \sim_s^{gfp_1} \cdot \sigma_1(\sim_s)^{gfp_2} \cdot \sim_s^{gfp_1} \cdot \sim_s^{gfp_1} \\
&= \sim_s^{gfp_1} \cdot \sigma_1(\sim_s)^{gfp_2} \cdot \sim_s^{gfp_1} = \sim_s^{\sigma_1(gfp_1)} \cdot \sigma_1(\sim_s)^{gfp_2} \cdot \sim_s^{\sigma_1(gfp_1)} \\
&= \sigma_1(\sim_s)^{gfp_1} \cdot \sigma_1(\sim_s)^{gfp_2} \cdot \sigma_1(\sim_s)^{gfp_1} = \sigma_1(\sim_s)^{F(gfp_2)} = \sim_s^{\sigma_1(F(gfp_2))}. \quad ❑
\end{aligned}$$

9 Finitely branching swinging types

Theorem 9.1 *Given the assumptions of Def. 8.1, an R_1-positive Σ'-formula φ is Σ'-compact over A iff for all increasing chains $B_0 \leq B_1 \leq B_2 \leq \ldots$ of $Mod(\Sigma', A)$,*

(1) sets $\{\psi_j : \prod_{i \in I_j} s_i\}_{j \in J}$ of Σ'-formulas and $a \in \prod_{i \in \cup\{I_j | j \in J\}} s_i^A$ such that $\bigwedge_{j \in J} \psi_j$ is a subformula of φ,

$$\forall j \in J\ \exists i \in \mathbb{N} : \pi_{I_j}(a) \in \psi_j^{B_i} \quad \textit{implies} \quad \exists i \in \mathbb{N}\ \forall j \in J : \pi_{I_j}(a) \in \psi_j^{B_i}, \tag{3}$$

(2) Σ'-formulas $\psi : \prod_{i \in I} s_i$, $k \in I$ and $a \in \prod_{i \in I \setminus \{k\}} s_i^A$ such that $\forall k \psi$ is a subformula of φ,

$$\forall b \in s_k^A\ \exists i \in \mathbb{N} : a *_k b \in \psi^{B_i} \quad \textit{implies} \quad \exists i \in \mathbb{N}\ \forall b \in s_k^A : a *_k b \in \psi^{B_i}. \tag{4}$$

Proof. Let $B_0 \leq B_1 \leq B_2 \leq \ldots$ be an increasing chain of $Mod(\Sigma', A)$. We show 8.11(2) for all R_1-positive Σ'-formulas φ. Let $B = \sqcup_{i \in \mathbb{N}} B_i$, $(\varphi, a) \in M_\alpha$, $a \in \varphi^B$ and I be a nonempty set.

- Let $\varphi = r(t)$ be a Σ'-atom. If $r \in R$, then φ is a Σ-formula and thus $a \in \varphi^B = \varphi^A = \bigcup_{i \in \mathbb{N}} \varphi^{B_i}$. If $r \in R_1$, then $a \in \varphi^B = \bigcup_{i \in \mathbb{N}} \varphi^{B_i}$.
- Let $\varphi = \neg\psi$ for some Σ-atom $\psi : s$. Hence $\psi^B = \psi^A = \bigcap_{i \in \mathbb{N}} \psi^{B_i}$ and thus
$$a \in \varphi^B = s^A \setminus \psi^B = s^A \setminus \bigcap_{i \in \mathbb{N}} \psi^{B_i} = \bigcup_{i \in \mathbb{N}} (s^A \setminus \psi^{B_i}) = \bigcup_{i \in \mathbb{N}} \varphi^{B_i}.$$
- Let $\varphi = \bigwedge_{j \in J} \psi_j : \prod_{i \in I_j} s_i$. Then for all $j \in J$ there is $\alpha_j < \alpha$ such that $(\psi_j, \pi_{I_j}(a)) \in M_{\alpha_j}$. $a \in \varphi^B$ implies $\pi_{I_j}(a) \in \psi_j^B$ for all $j \in J$. Hence by induction hypothesis, $\pi_{I_j}(a) \in \bigcup_{i \in \mathbb{N}} \psi_j^{B_i}$, i.e., there is $i \in \mathbb{N}$ such that $\pi_{I_j}(a) \in \psi^{B_i}$. By (3), there is $i \in \mathbb{N}$ such that for all $j \in J$, $\pi_{I_j}(a) \in \psi_j^{B_i}$. Hence $a \in \varphi^{B_i} \subseteq \bigcup_{i \in \mathbb{N}} \varphi^{B_i}$.
- Let $\varphi = \bigvee_{j \in J} \psi_j : \prod_{i \in I_j} s_i$. Then for all $j \in J$ there is $\alpha_j < \alpha$ such that $(\psi_j, \pi_{I_j}(a)) \in M_{\alpha_j}$. Hence by induction hypothesis, $\pi_{I_j}(a) \in \bigcup_{i \in \mathbb{N}} \psi_j^{B_i}$, i.e., there is $i \in \mathbb{N}$ such that $\pi_{I_j}(a) \in \psi^{B_i}$. Hence $a \in \varphi^{B_i} \subseteq \bigcup_{i \in \mathbb{N}} \varphi^{B_i}$.
- Let $\varphi = \forall k \psi$ for some $k \in I$. Then for all $b \in s_k^A$ there is $\alpha_b < \alpha$ such that $(\psi, a *_k b) \in M_{\alpha_b}$. $a \in \varphi^B$ implies $a *_k b \in \psi^B$ for all $b \in s_k^A$. Hence by induction hypothesis, $a *_k b \in \bigcup_{i \in \mathbb{N}} \psi^{B_i}$, i.e., there is $i \in \mathbb{N}$ such that $a *_k b \in \psi^{B_i}$. By (4), there is $i \in \mathbb{N}$ such that for all $b \in s_k^A$, $a *_k b \in \psi^{B_i}$. Hence $a \in \varphi^{B_i} \subseteq \bigcup_{i \in \mathbb{N}} \varphi^{B_i}$.
- Let $\varphi = \exists k \psi$ for some $k \in I$. Then for all $b \in s_k^A$ there is $\alpha_b < \alpha$ such that $(\psi, a *_k b) \in M_{\alpha_b}$. $a \in \varphi^B$ implies $a *_k b \in \psi^B$ for some $b \in s_k^A$. Hence by induction hypothesis, $a *_k b \in \bigcup_{i \in \mathbb{N}} \psi^{B_i}$, i.e., there is $i \in \mathbb{N}$ such that $a *_k b \in \psi^{B_i}$. Hence $a \in \varphi^{B_i} \subseteq \bigcup_{i \in \mathbb{N}} \varphi^{B_i}$. ❑

By dualizing the preceding proof, one obtains:

Theorem 9.2 *Given the assumptions of Def. 8.1, an R_1-positive Σ'-formula φ is Σ'-cocompact over A iff for all decreasing chains $B_0 \geq B_1 \geq B_2 \geq \ldots$ of $Mod(\Sigma', A)$,*

(1) sets $\{\psi_j : \prod_{i \in I_j} s_i\}_{j \in J}$ of Σ'-formulas and $a \in \prod_{i \in \cup\{I_j | j \in J\}} s_i^A$ such that $\bigvee_{j \in J} \psi_j$ is a subformula of φ,

$$\forall i \in \mathbb{N}\ \exists j \in J : \pi_{I_j}(a) \in \psi_j^{B_i} \quad \textit{implies} \quad \exists j \in J\ \forall i \in \mathbb{N} : \pi_{I_j}(a) \in \psi_j^{B_i},$$

(2) Σ'*-formulas* $\psi : \prod_{i\in I} s_i$, $k \in I$ *and* $a \in \prod_{i\in I\setminus\{k\}} s_i^A$ *such that* $\exists k\psi$ *is a subformula of* φ,

$$\forall i \in \mathbb{N}\ \exists b \in s_k^A : a *_k b \in \psi^{B_i} \quad \textit{implies} \quad \exists b \in s_k^A\ \forall i \in \mathbb{N} : a *_k b \in \psi^{B_i}.$$

Theorems 9.1 and 9.2 lead to the following criterion for compactness resp. cocompactness:

Definition 9.3 *(finitely branching)* Let the assumptions of Def. 8.1 hold true and $B \in Mod(\Sigma', A)$.

A set $\{\varphi_j : \prod_{i\in I_j} s_i\}_{j\in J}$ is **finitely B-solvable** if for all $a \in \prod_{i\in\cup\{I_j|j\in J\}} s_i^B$, the set of $j \in J$ such that $\pi_{I_j}(a) \in \varphi_j^B$ is finite.

A Σ'-formula $\varphi : \prod_{i\in I} s_i$ is **finitely B-solvable in** $k \in I$ if for all $a \in \prod_{i\in I\setminus\{k\}} s_i^B$, the set of $b \in s_k^B$ such that $a *_k b \in \varphi^B$ is finite.

SP' is **finitely branching in** B if for all Horn clauses $p \Leftarrow \varphi \in AX_1$, co-Horn clauses $q \Rightarrow \psi \in AX_1$, subformulas $\bigwedge_{j\in J} \varphi_j$ and $\forall k : \vartheta$ of φ and subformulas $\bigvee_{j\in J} \varphi_j$ and $\exists k : \vartheta$ of ψ, $\{\varphi_j\}_{j\in J}$ is finitely B-solvable and ϑ is finitely B-solvable in k. ❏

Modal logic achieves continuity by restricting the bodies of modal operators like $\Box$ and $\Diamond$ to propositions about *finitely branching* transition systems. Roughly said, the restriction implies that the set of solutions of these bodies in the quantified variables is finite. The following example may illustrate the connection between finitely branching transition systems and (co)compact formulas.

Given a specification of a set *State* of states and a transition system $\to$ on *State*, the least relation $r \subseteq State$ that satisfies the Horn clause

$$r(s) \quad \Leftarrow \quad \forall s' : (s \to s' \Rightarrow q(s')) \tag{1}$$

consists of all states that admit only finite runs w.r.t. $\to$. Since, in this example, the models B_i in Def. 8.11 reduce to sets S_i of states (i.e., the different interpretations of r), the premise of (1) is compact iff for all increasing chains $S_0 \subseteq S_1 \subseteq S_2 \subseteq \ldots$ of sets of states,

$$\forall s' \exists i_{s'} : (s \to s' \Rightarrow s' \in S_{i_{s'}}) \quad \text{implies} \quad \exists i \forall s' : (s \to s' \Rightarrow s' \in S_i). \tag{2}$$

Now suppose that the premise of (2) holds true and $\to$ is finitely branching. If the chain becomes stationary, i.e., there is $n \in \mathbb{N}$ such that $S_j = S_n$ for all $j \geq n$, then the conclusion of (2) holds true for $i = n$. Otherwise there is $n \in \mathbb{N}$ such that S_n consists of all direct successors of s w.r.t. $\to$. Again $i = n$ satisfies the conclusion of (2). Duality suggests that a finitely branching transition system $\to$ also entails that the conclusion of the co-Horn clause that results from negating (1):

$$r(s) \quad \Rightarrow \quad \exists s' : (s \to s' \wedge q(s')), \tag{3}$$

is cocompact. Indeed, (3) is cocompact iff for all decreasing chains $S_0 \supseteq S_1 \supseteq S_2 \supseteq \ldots$ of sets of states,

$$\forall i \exists s'_i : (s \to s' \wedge s'_i \in S_i) \quad \text{implies} \quad \exists s' \forall i : (s \to s' \wedge s' \in S_i). \tag{4}$$

Suppose that the premise of (4) holds true and $\to$ is finitely branching. Then there is a state s' such that $s \to s'$ and $s' = s'_i$ for infinitely many $i \in \mathbb{N}$. Hence for all $i \in \mathbb{N}$ there is $n_i \geq i$ with $s'_{n_i} = s'$. Since $s' = s'_{n_i} \in S_{n_i} \subseteq S_i$, we obtain the conclusion of (4).

Moreover, the invariance and congruence axioms of Definition 10.1 are finitely branching.

Lemma 9.4 *Let the assumptions of Def. 8.1 hold true. If SP' is finitely branching in all $B \in Mod(\Sigma', A)$, then the premises of all Horn clauses of AX_1 are Σ'-compact over A and the conclusions of all co-Horn clauses of AX_1 are Σ'-cocompact over A.*

Proof. Let $p \Leftarrow \varphi \in AX_1$ and $B_0 \leq B_1 \leq B_2 \leq \ldots$ be an increasing chain of $Mod(\Sigma', A)$.

Let $\{\varphi_j : \prod_{i\in I_j} s_i\}_{j\in J}$ be a set of Σ'-formulas and $a \in \prod_{i\in\cup\{I_j|j\in J\}} s_i^B$ such that $\bigwedge_{j\in J}\varphi_j$ is a subformula of φ. Suppose that for all $j\in J$ there is $i\in\mathbb{N}$ such that $\pi_{I_j}(a)\in\varphi_j^{B_i}$. Since SP' is finitely branching in B_i, the set of $j\in J$ such that $\pi_{I_j}(a)\in\varphi_j^B$ is finite. Hence there is $n\in\mathbb{N}$ such that for all $j\in J$, $\pi_{I_j}(a)\in\varphi_j^{B_i}$ for some $i\le n$. Since $B_i\le B_n$ and thus, by Theorem 8.13, $\varphi_j^{B_i}\subseteq\varphi_j^{B_n}$, we conclude that for all $j\in J$, $\pi_{I_j}(a)\in\varphi_j^{B_n}$.

Let $\psi : \prod_{i\in I} s_i$ be a Σ'-formula, $k\in I$ and $a\in\prod_{i\in I\setminus\{k\}} s_i^A$ such that $\forall k\psi$ is a subformula of φ. Suppose that for all $b\in s_k^A$ there is $i\in\mathbb{N}$ such that $a*_k b\in\psi^{B_i}$. Since SP' is finitely branching in B_i, the set of $b\in s_k^{B_i}$ such that $a*_k b\in\psi^{B_i}$ is finite. Hence there is $n\in\mathbb{N}$ such that for all $b\in s_k^B$, $a*_k b\in\psi^{B_i}$ for some $i\le n$. Since $B_i\le B_n$ and thus, by Theorem 8.13, $\psi^{B_i}\subseteq\psi^{B_n}$, we conclude that for all $b\in s_k^{B_i}$, $a*_k b\in\psi^{B_n}$.

Let $p\Rightarrow\varphi\in AX_1$ and $B_0\ge B_1\ge B_2\ge\dots$ be a decreasing chain of $Mod(\Sigma',A)$.

Let $\{\varphi_j : \prod_{i\in I_j} s_i\}_{j\in J}$ be a set of Σ'-formulas and $a \in \prod_{i\in\cup\{I_j|j\in J\}} s_i^B$ such that $\bigvee_{j\in J}\varphi_j$ is a subformula of φ. Suppose that for all $i\in\mathbb{N}$ there is $j_i\in J$ such that $\pi_{I_{j_i}}(a)\in\varphi_{j_i}^{B_i}$. Since SP' is finitely branching in B_i, the set of $j\in J$ such that $\pi_{I_j}(a)\in\varphi_j^{B_i}$ is finite. Hence there are $j\in J$ and infinitely many $i\in\mathbb{N}$ such that $j_i=j$. Consequently, for all $i\in\mathbb{N}$ there is $n_i\ge i$ such that $j_{n_i}=j$ and thus $\pi_{I_j}(a)=\pi_{I_{j_{n_i}}}(a)\in\varphi_{j_{n_i}}^{B_{n_i}}=\varphi_j^{B_{n_i}}$. Since $B_i\ge B_{n_i}$ and thus, by Theorem 8.13, $\varphi_j^{B_i}\supseteq\varphi_j^{B_{n_i}}$, we conlude that for all $i\in\mathbb{N}$, $\pi_{I_j}(a)\in\varphi_j^{B_i}$.

Let $\psi : \prod_{i\in I} s_i$ be a Σ'-formula, $k\in I$ and $a\in\prod_{i\in I\setminus\{k\}} s_i^A$ such that $\exists k\psi$ is a subformula of φ. Suppose that for all $i\in\mathbb{N}$ there is $b_i\in s_k^A$ such that $a*_k b_i\in\psi^{B_i}$. Since SP' is finitely branching in B_i, the set of $b\in s_k^{B_i}$ such that $a*_k b\in\psi^{B_i}$ is finite. Hence there are $b\in s_k^B$ and infinitely many $i\in\mathbb{N}$ such that $b_i=b$. Consequently, for all $i\in\mathbb{N}$ there is $n_i\ge i$ such that $b_{n_i}=b$ and thus $a*_k b=a*_k b_{n_i}\in\psi^{B_{n_i}}$. Since $B_i\ge B_{n_i}$ and thus, by Theorem 8.13, $\psi^{B_i}\supseteq\psi^{B_{n_i}}$, we conlude that for all $i\in\mathbb{N}$, $a*_k b\in\psi^{B_i}$. ❑

Proposition 8.12, Theorems 9.1 and 9.2 and Lemma 9.4 immediately imply:

Theorem 9.5 *Let the assumptions of Def. 8.1 hold true, $\Sigma_1=(S_0,S,F',R)$ and A be a Σ_1-structure. $\Phi_{A,\sigma}$ is continuous resp. cocontinuous over A iff SP' is finitely branching over A.* ❑

If the clauses of $AX'\setminus AX$ are Σ'-compact resp. -cocompact over A, then by Proposition 8.12, $\Phi_{A,\sigma}$ is continuous resp. cocontinuous over A. Hence Theorem 8.3 (Kleene) provides us with an inductive construction of the least resp. greatest fixpoint of σ: for all $r\in R'\setminus R$,

$$r^{lfp(\Phi_{A,\sigma})} = \bigcup_{i\in\mathbb{N}} r^{\Phi^i_{A,\sigma}(\bot)} \quad\text{resp.}\quad r^{gfp(\Phi_{A,\sigma})} = \bigcap_{i\in\mathbb{N}} r^{\Phi^i_{A,\sigma}(\top)}.$$

Consequently, a property P holds true for $lfp(\Phi_{A,\sigma})$ iff there is $i\in\mathbb{N}$ such that P is valid for $\Phi^i_{A,\sigma}(\bot)$, while P holds true for $gfp(\Phi_{A,\sigma})$ iff for all $i\in\mathbb{N}$, P is valid for $\Phi^i_{A,\sigma}(\top)$.

10 Abstraction and restriction

Definition 10.1 *(congruence and invariant axioms)* Let $\Sigma=(S_0,S,F,R)$ be a signature. The congruence property of equalities can be axiomatized either by Horn clauses (**CONH**) or by co-Horn clauses (**CONC**):

$f(x)\equiv_{s'} f(y) \quad\Leftarrow\quad x\equiv_s y$	for all $f:s\to s'\in F$	CONH1
$r(x) \quad\Leftarrow\quad x\equiv_s y\wedge r(y)$	for all $r:s\in R$	CONH2
$(x_i)_{i\in I}\equiv_{\prod_{i\in I}s_i}(y_i)_{i\in I} \quad\Leftarrow\quad \bigwedge_{i\in I}x_i\equiv_{s_i}y_i$	for all $\{s_i\}_{i\in I}\subseteq\mathbb{T}_S$	CONH3
$\iota_i(x)\equiv_{\coprod_{i\in I}s_i}\iota_i(y) \quad\Leftarrow\quad x\equiv_{s_i}y$	for all $\{s_i\}_{i\in I}\subseteq\mathbb{T}_S$ and $i\in I$	CONH4

$x \equiv_s y \quad \Rightarrow \quad f(x) \equiv_{s'} f(y)$	for all $f : s \to s' \in F$	CONC1
$r(x) \quad \Rightarrow \quad (x \equiv_s y \Rightarrow r(y))$	for all $r : s \in R$	CONC2
$(x_i)_{i\in I} \equiv_{\prod_{i\in I} s_i} (y_i)_{i\in I} \quad \Rightarrow \quad x_i \equiv_{s_i} y_i$	for all $\{s_i\}_{i\in I} \subseteq \mathbb{T}_S$ and $i \in I$	CONC3
$\iota_i(x) \equiv_{\coprod_{i\in I} s_i} \iota_i(y) \quad \Rightarrow \quad x \equiv_{s_i} y$	for all $\{s_i\}_{i\in I} \subseteq \mathbb{T}_S$ and $i \in I$	CONC4
$\iota_i(x) \equiv_{\coprod_{i\in I} s_i} \iota_j(y) \quad \Rightarrow \quad False$	for all $\{s_i\}_{i\in I} \subseteq \mathbb{T}_S$ and $i, j \in I$ with $i \neq j$	CONC4

Analogously, the invariant property of universes can be axiomatized either by Horn clauses (**INVH**) or by co-Horn clauses (**INVC**):

$all_{s'}(f(x)) \Leftarrow all_s(x)$	for all $f : s \to s' \in F$	INVH1
$all_{\prod_{i\in I} s_i}(x_i)_{i\in I} \quad \Leftarrow \quad \bigwedge_{i\in I} all_{s_i}(x_i)$	for all $\{s_i\}_{i\in I} \subseteq \mathbb{T}_S$	INVH2
$all_{\coprod_{i\in I} s_i}(\iota_i(x)) \quad \Leftarrow \quad all_{s_i}(x)$	for all $\{s_i\}_{i\in I} \subseteq \mathbb{T}_S$ and $i \in I$	INVH3

$all_s(x) \Rightarrow all_{s'}(f(x))$	for all $f : s \to s' \in F$	INVC1
$all_{\prod_{i\in I} s_i}(x_i)_{i\in I} \quad \Rightarrow \quad all_{s_i}(x_i)$	for all $s_1, \ldots, s_n \in \mathbb{T}_S$ and $1 \leq i \leq n$	INVC2
$all_{\coprod_{i\in I} s_i}(\iota_i(x)) \quad \Rightarrow \quad all_{s_i}(x)$	for all $s_1, \ldots, s_n \in \mathbb{T}_S$ and $1 \leq i \leq n$	INVC3

These axioms include the extensions of $\equiv$ and *all* to products and sums that are also called relation resp. predicate liftings (see section 4).

Lemma 10.2 *Let the assumptions of Def. 8.1 hold true, $A \in Mod(SP)$ and $B \in Mod(\Sigma', A)$.*

(1) If SP' is a μ-extension of SP and $CONH \subseteq AX_1$, then $\equiv^B$ is an R'-compatible Σ'-congruence.

(2) If SP' is a ν-extension of SP and $CONC \subseteq AX_1$, then $\equiv^B$ is an R'-compatible Σ'-congruence.

(3) If SP' is a μ-extension of SP and $INVH \subseteq AX_1$, then all^B is a Σ'-invariant.

(4) If SP' is a ν-extension of SP and $INVC \subseteq AX_1$, then all^B is a Σ'-invariant.

(5) If SP' is a ν-extension of SP, $CONC \subseteq AX_1$ and $B = gfp(\Phi_{A,\sigma})$, then $\equiv^B$ is an R'-compatible equivalence relation.

Proof. (1)-(4) hold true trivially.

(5) Let $B^* = \sqcup_{i\in\mathbb{N}} B_i$ where $B_i \in Mod(\Sigma', A)$ is defined as follows: Let $s \in \mathbb{T}_S$ and $r : s' \in R' \setminus R$.

- $\equiv_s^{B_0} = \Delta_s^B \cup \equiv_s^B \cup (\equiv_s^B)^{-1}$ and $r^{B_0} = r^B$.
- For all $i > 0$, $\equiv_s^{B_{i+1}} = \{(a, b) \in s^A \times s^A \mid \exists\, c : (a \equiv_s^{B_i} c \wedge c \equiv_s^{B_i} b)\}$.
- For all $i > 0$, $r^{B_{i+1}} = \{a \in A_{s'} \mid \exists\, b : (a \equiv_{s'}^B b \wedge b \in r^{B_i})\}$.

Suppose that B^* satisfies CONC. Since B is the greatest Σ'-structure over A that satisfies CONC, $B^* \leq B$ and thus $B^* = B$ because $B \leq B^*$. Since B^* is an R'-compatible equivalence relation, the proof is complete.

It remains to prove $B^* \models$ CONC. we show $B_i \models$ CONC for all $i \in \mathbb{N}$. Let $s \in S_1$ and $r : s \in R' \setminus R$. Since $\equiv^{B_0}$ is a Σ'-congruence, $B_0 \models$ CONC. Let $i > 0$. By induction hypothesis, B_{i-1} satisfies CONC.

We show that B_i satisfies CONC1. Let $f : s \to s' \in F \setminus F_0$ and $a \equiv_s^{B_i} b$. Then there is $c \in s^A$ such that $a \equiv_s^{B_{i-1}} c$ and $c \equiv_s^{B_{i-1}} b$. Since B_{i-1} satisfies CONC1, $f^A(a) \equiv_s^{B_{i-1}} f^A(c)$ and $f^A(c) \equiv_s^{B_{i-1}} f^A(b)$. Hence $f^A(a) \equiv_s^{B_i} f^A(b)$.

We show that B_i satisfies CONC2. Let $s = \prod_{i\in I} s_i \in \mathbb{T}_S$ and $(a_j)_{j\in I} \equiv_s^{B_i} (b_j)_{j\in I}$. Then there is $\{c_j\}_{j\in I} \subseteq s^A$ such that $(a_j)_{j\in I} \equiv^B (c_j)_{j\in I}$ and $(c_j)_{j\in I} \equiv_s^{B_{i-1}} (b_j)_{j\in I}$. Since B satisfies CONC2, for all $j \in I$, $a_j \equiv_{s_j}^{B_{i-1}} c_j$

and $c_j \equiv_{s_j}^{B_{i-1}} b_j$. Hence $a_j \equiv_{s_j}^{B_i} b_j$.

Analogously, B_i satisfies CONC3-CONC9. ❏

Theorem 10.3 *Let $\Sigma = (S_0, S, F, R)$ be a signature, $A \in Mod(\Sigma)$, $\sim$ be a Σ-congruent and R-compatible equivalence relation on A and φ be a Σ-formula. Then $B =_{def} A/{\sim}$ satisfies φ iff A satisfies φ.*

Proof. The conjecture holds true if

$$\{[a] \in B \mid a \in \varphi^A\} = \varphi^{A/\sim}. \tag{1}$$

(1) is shown by induction on the structure of φ: Let $r(t)$ be a Σ-atom. Then by Proposition 4.14,

$$\begin{gathered} a \in (r \circ t)^A = (t^A)^{-1}(r^A) \iff t^A(a) \in r^A \iff t^B([a]) = [t^A(a)] \in r^B \\ \iff [a] \in (t^B)^{-1}(r^A) = (r \circ t)^B. \end{gathered}$$

Let $\varphi : s$ be a Σ-formula. By induction hypothesis,

$$a \in (\neg\varphi)^A = s^A \setminus \varphi^A \iff [a] \in B_s \setminus \varphi^{A/\sim} = (\neg\varphi)^B.$$

Let $\{\varphi_j : \prod_{i \in I_j} s_i\}_{j \in J}$ be a set of Σ-formulas. Then, by induction hypothesis,

$$\begin{gathered} a \in (\textstyle\bigwedge_{j \in J} \varphi_j)^A = \bigcap_{j \in J} \pi_{I_j}^{-1}(\varphi_j^A) \iff \forall\, j \in J : \pi_{I_j}(a) \in \varphi_j^A \iff \forall\, j \in J : \pi_{I_j}([a]) = [\pi_{I_j}(a)] \in \varphi_j^B \\ \iff [a] \in \textstyle\bigcap_{j \in J} \pi_{I_j}^{-1}(\varphi_j^B) = (\bigwedge_{j \in J} \varphi_j)^B. \end{gathered}$$

Let $\varphi : \prod_{i \in I} s_i$ be a Σ-formula and $k \in I$. By induction hypothesis,

$$\begin{gathered} a \in (\forall k\varphi)^A = \textstyle\bigcap_{b \in s_k^A} (\varphi^A \div_k b) \iff \forall\, b \in s_k^A : a \in \varphi^A \div_k b \\ \iff \forall\, b \in s_k^A : a *_k b \in \varphi^A \iff \forall\, [b] \in s_k^B : [a] *_k [b] = [a *_k b] \in \varphi^B \\ \iff \forall\, [b] \in s_k^B : [a] \in \varphi^B \div_k [b] \iff [a] \in \textstyle\bigcap_{[b] \in s_k^B} (\varphi^B \div_k [b]) = (\forall k\varphi)^B. \end{gathered}$$ ❏

Theorem 10.4 *Let $\Sigma = (S_0, S, F, R)$ be a signature, $A \in Mod(\Sigma)$, $inv =_{def} all^A$ be a Σ-invariant on A, $S_1 = S \setminus S_0$ and $\varphi : s$ be a restricted Σ-formula. Then $B =_{def} A|inv$ satisfies φ if A satisfies φ.*

Proof. The conjecture holds true if

$$\varphi^A \cap inv_s = \varphi^B. \tag{1}$$

(1) is shown by induction on the structure of φ: Let $r(t) : s$ be a Σ-atom and $a \in inv_s$. Since inv is Σ-invariant, $t^A(a) \in inv$ and thus by Proposition 4.14,

$$\begin{gathered} a \in (r \circ t)^A = (t^A)^{-1}(r^A) \iff t^A(a) \in r^A \iff t^B(a) = t^A(a) \in r^A \cap inv = r^B \\ \iff a \in (t^B)^{-1}(r^B) = (r \circ t)^B. \end{gathered}$$

Let $\varphi : s$ be an restricted Σ-formula and $a \in inv_s$. By induction hypothesis,

$$a \in (\neg\varphi)^A = s^A \setminus \varphi^A \iff a \in inv_s \setminus \varphi^A = inv_s \setminus (\varphi^A \cap inv_s) = inv_s \setminus \varphi^B = (\neg\varphi)^B.$$

Let $\{\varphi_j : \prod_{i \in I_j} s_i\}_{j \in J}$ be a set of Σ-formulas and $s = \prod_{i \in \cup\{I_j | j \in J\}}$ such that $\varphi = \bigwedge_{j \in J} \varphi_j$ is restricted and $a \in inv_s$. Then for all $j \in J$, $\pi_{I_j}(a) \in inv_{\prod_{i \in I_j} s_i}$, and thus by induction hypothesis,

$$\begin{gathered} a \in (\textstyle\bigwedge_{j \in J} \varphi_j)^A = \bigcap_{j \in J} \pi_{I_j}^{-1}(\varphi_j^A) \iff \forall\, j \in J : \pi_{I_j}(a) \in \varphi_j^A \\ \iff \forall\, j \in J : \pi_{I_j}(a) \in \varphi_j^A \cap inv_{\prod_{i \in I_j} s_i} = \varphi_j^B \iff a \in \textstyle\bigcap_{j \in J} \pi_{I_j}^{-1}(\varphi_j^B) = (\bigwedge_{j \in J} \varphi_j)^B. \end{gathered}$$

Let $\varphi : \prod_{i\in I} s_i$ be a Σ-formula, $k \in I$ and $s = \prod_{i\in I\setminus\{k\}} s_i$ such that $\forall k\varphi$ is restricted and $a \in inv_s$. Let $s_k \in S_0$. Then $inv_{s_k} = A_{s_k}$. Hence for all $b \in A_{s_k}$, $a *_k b \in inv_{\prod_{i\in I} s_i}$, and thus by induction hypothesis,

$$\begin{gathered}
a \in (\forall k\varphi)^A = \textstyle\bigcap_{b\in s_k^A}(\varphi^A \div_k b) = \bigcap_{b\in inv_{s_k}}(\varphi^A \div_k b) \iff \forall\ b \in inv_{s_k} : a \in \varphi^A \div_k b \\
\iff \forall\ b \in inv_{s_k} : a *_k b \in \varphi^A \iff \forall\ b \in inv_{s_k} : a *_k b \in \varphi^A \cap inv_{\prod_{i\in I} s_i} = \varphi^B \\
\iff \forall\ b \in inv_{s_k} : a \in \varphi^B \div_k b \iff a \in \textstyle\bigcap_{b\in inv_{s_k}}(\varphi^B \div_k b) = (\forall k\varphi)^B.
\end{gathered}$$

Let $s_k \in S_1$. Since $\forall k\varphi$ is restricted, w.l.o.g. $\varphi = \neg all_{s_k}(k) \vee \psi$ for some Σ-formula ψ. Hence $a \in inv_s$ and the induction hypothesis imply

$$\begin{gathered}
a \in (\forall k\varphi)^A = \textstyle\bigcap_{b\in s_k^A}(\varphi^A \div_k b) \iff \forall\ b \in s_k^A : a \in \varphi^A \div_k b \iff \forall\ b \in s_k^A : a *_k b \in \varphi^A = (\neg all_{s_k}(k) \vee \psi)^A \\
\iff \forall\ b \in s_k^A : (b \notin all_{s_k}^A = inv_{s_k} \vee a *_k b \in \psi^A) \iff \forall\ b \in inv_{s_k} : a *_k b \in \psi^A \\
\iff \forall\ b \in inv_{s_k} : a *_k b \in \psi^A \cap inv_{\prod_{i\in I} s_i} = \psi^B \\
\iff \forall\ b \in inv_{s_k} : (a *_k b = b \notin inv_{s_k} = all_{s_k}^B \vee a *_k b \in \psi^B) \\
\iff \forall\ b \in inv_{s_k} : a *_k b \in (\neg all_{s_k}(k) \vee \psi)^B = \varphi^B \iff \forall\ b \in inv_{s_k} : a \in \varphi^B \div_k b \\
\iff a \in \textstyle\bigcap_{b\in inv_{s_k}}(\varphi^B \div_k b) = (\forall k\varphi)^B. \quad ❑
\end{gathered}$$

Theorem 10.5 *Let $\Sigma = (S, F, R)$ be an algebraic signature, $A, B \in Mod(\Sigma)$ and $\varphi : s$ be an implicational Σ-formula. Then $A \times B$ satisfies φ if A and B satisfy φ.*[12]

Proof. Note that s is a product of sorts because Σ is algebraic. The conjecture holds true if

$$\{\langle a, b\rangle \mid a \in \varphi^A \wedge b \in \varphi^B\} \subseteq \varphi^{A\times B}. \tag{1}$$

At first we show

$$\{\langle a, b\rangle \mid a \in \varphi^A \wedge b \in \varphi^B\} = \varphi^{A\times B} \tag{2}$$

for all universally quantified conjunctions φ of Σ-atoms by induction on the structure of φ: Let $r(t) : s$ be a Σ-atom, $a \in s^A$ and $b \in s^B$. Then by Proposition 4.14,

$$\begin{gathered}
a \in r(t)^A = (r \circ t)^A = (t^A)^{-1}(r^A) \wedge b \in r(t)^B = (r \circ t)^B = (t^B)^{-1}(r^B) \iff t^A(a) \in r^A \wedge t^B(b) \in r^B \\
\iff \langle t^A(a), t^B(b)\rangle \in r^{A\times B} \iff \langle a, b\rangle \in (t^{A\times B})^{-1}(r^{A\times B}) = (r \circ t)^{A\times B} = r(t)^{A\times B}.
\end{gathered}$$

Let $\{\varphi_j : \prod_{i\in I_j} s_i\}_{j\in J}$ be a set of implicational Σ-formulas. Then, by induction hypothesis,

$$\begin{gathered}
a \in (\textstyle\bigwedge_{j\in J} \varphi_j)^A = \bigcap_{j\in J} \pi_{I_j}^{-1}(\varphi_j^A) \wedge b \in (\bigwedge_{j\in J} \varphi_j)^B = \bigcap_{j\in J} \pi_{I_j}^{-1}(\varphi_j^B) \\
\iff \forall\ j \in J : \pi_{I_j}(a) \in \varphi_j^A \wedge \forall\ j \in J : \pi_{I_j}(b) \in \varphi_j^B \iff \forall\ j \in J : \pi_{I_j}(\langle a, b\rangle) = \langle \pi_{I_j}(a), \pi_{I_j}(b)\rangle \in \varphi_j^{A\times B} \\
\iff \langle a, b\rangle \in \textstyle\bigcap_{j\in J} \pi_{I_j}^{-1}(\varphi_j^{A\times B}) = (\bigwedge_{j\in J} \varphi_j)^{A\times B}.
\end{gathered}$$

Let $\varphi : \prod_{i\in I} s_i$ be an implicational Σ-formula and $k \in I$. By induction hypothesis,

$$\begin{gathered}
a \in (\forall k\varphi)^A = \textstyle\bigcap_{c\in s_k^A}(\varphi^A \div_k c) \wedge b \in (\forall k\varphi)^B = \bigcap_{d\in s_k^B}(\varphi^B \div_k d) \\
\iff \forall\ c \in s_k^A : a \in \varphi^A \div_k c \wedge \forall\ d \in s_k^B : b \in \varphi^B \div_k d \iff \forall\ c \in s_k^A : a *_k c \in \varphi^A \wedge \forall\ d \in s_k^B : b *_k d \in \varphi^B \\
\iff \forall\ \langle c, d\rangle \in s_k^{A\times B} : \langle a, b\rangle *_k \langle c, d\rangle = \langle a *_k c, b *_k d\rangle \in \varphi^{A\times B} \\
\iff \forall\ \langle c, d\rangle \in s_k^{A\times B} : \langle a, b\rangle \in \varphi^{A\times B} \div_k \langle c, d\rangle \iff \langle c, d\rangle \in \textstyle\bigcap_{\langle c,d\rangle\in s_k^{A\times B}}(\varphi^{A\times B} \div_k \langle c, d\rangle) = (\forall k\varphi)^{A\times B}.
\end{gathered}$$

Secondly, we show (1) by induction on the structure of an implicational Σ-formula φ: If φ is an atom, a conjunction or a universally quantified formula, the proof proceeds analogously to the above proof of (2): just

[12] Counterexamples showing that this theorem cannot be generalized to non-algebraic signatures or non-implicational formulas are obtained easily from the study of products and implicational classes in universal algebra. See, e.g., [71], Section 5.3, and [111], Section 3.3.

replace "universally quantified conjunction of atoms" by implicational formula". It remains to derive (1) for simple implications $\psi \Rightarrow \varphi$ (see Def. 3.12) from the validity of (1) for φ: Let $\varphi : \prod_{i\in J} s_i$ be an implicational Σ-formula and $\psi : \prod_{i\in I} s_i$ be a universally quantified conjunction of atoms. By (2) and induction hypothesis,

$$\begin{array}{c} a \in (\psi \Rightarrow \varphi)^A \wedge b \in (\psi \Rightarrow \varphi)^B \iff (a \notin \psi^A \vee a \in \varphi^A) \wedge (b \notin \psi^B \vee b \in \varphi^B) \\ \iff (a \notin \psi^A \wedge b \notin \psi^B) \vee (a \notin \psi^A \wedge b \in \varphi^B) \vee (a \in \varphi^A \wedge b \notin \psi^B) \vee (a \in \varphi^A \wedge b \in \varphi^B) \\ \implies a \notin \psi^A \vee b \notin \psi^B) \vee (a \in \varphi^A \wedge b \in \varphi^B) \\ \implies \langle a, b\rangle \notin \psi^{A\times B} \vee \langle a, b\rangle \in \varphi^{A\times B} \iff \langle a, b\rangle \in (\psi \Rightarrow \varphi)^{A\times B}. \quad ❏ \end{array}$$

Definition 10.6 *(reachable, observable, consistent, complete)* Let $\Sigma = (S_0, S, F, R)$ be a subsignature of a signature $\Sigma' = (S_0, S, F', R')$, $S_1 = S \setminus S_0$, $B \in Mod(\Sigma', A)$ for some S_0-sorted set A. The **Σ-reachability invariant of** B, $reach^B_\Sigma$, is the image of the S-sorted function $reach^B$ with

$$reach^B_s \quad = \quad [t]_{t\in MGen_{\Sigma,s}} : (\coprod_{t\in MGen_{\Sigma,s}} dom^A_t) \to s^B.$$

The **Σ-observability congruence of** B, obs^B_Σ, is the kernel of the S-sorted function obs^B with

$$obs^B_s \quad = \quad \langle t\rangle_{t\in MObs_{\Sigma,s}} : s^B \to \prod_{t\in MObs_{\Sigma,s}} ran^A_t.$$

B is **Σ-reachable** or **Σ-generated**[13] if $reach^B_\Sigma = B$.

B is **Σ-observable** or **Σ-cogenerated**[14] if $obs^B_\Sigma = \Delta^B$.

B is **Σ-consistent** if for all $t, u \in MGen_\Sigma$ and $a, b \in A$, $t^B(a) = u^B(b)$ implies $t^{Free(\Sigma,A)}(a) = u^{Free(\Sigma,A)}(b)$.

B is **Σ-complete** if for all $a \in Cofree(\Sigma, A)$, $obs^B(b) = a$ for some $b \in B$.

Given a class $\mathcal{C}$ of Σ'-structures, $\mathbf{R}\mathcal{C}$ and $\mathbf{O}\mathcal{C}$ denote the subclasses of Σ'-reachable and Σ'-observable structures of $\mathcal{C}$, respectively. ❏

In [83], both Σ-reachable and Σ-consistent structures are called *free*, while both Σ-observable and Σ-complete structures are called *cofree*.

Lemma 10.7 *Let the assumptions of Def. 10.6 hold true and $\Sigma = \Sigma'$.*

(1) Suppose that for all $f : s \to s' \in F$, $s, s' \in \mathbb{T}_{S_0}$ or $s' \in S_1$. Then the Σ-reachability invariant of B is a Σ-invariant.

(2) Suppose that for all $f : s \to s' \in F$, $s, s' \in \mathbb{T}_{S_0}$ or $s \in S_1$. Then the Σ-observability congruence of B is a Σ-congruence.

(3) Let $h : B \to C$ be a Σ-homomorphism. C is Σ-complete if B is Σ-complete. B is Σ-consistent if C is Σ-consistent.

(4) Let $h : B \to C$ be a Σ-epimorphism. C is Σ-reachable resp. Σ-observable if B is Σ-reachable resp. Σ-observable. B is Σ-complete if C is Σ-complete.

(5) Let $h : B \to C$ be a Σ-monomorphism. C is Σ-consistent if B is Σ-consistent. B is Σ-reachable resp. Σ-observable if C is Σ-reachable resp. Σ-observable.

Proof. (1) Let $s \in S$, $f : s \to s' \in F$ and $b \in reach^B_{\Sigma,s}$. Then $b = t^B(a)$ for some $t \in MGen_{\Sigma,s}$ and $a \in dom^A_t$. If $s, s' \in S_0$, then $id_{s'}$ is a maximal Σ-generator and thus $f^B(b) = id^B_{s'}(f^B(b)) \in reach^B_{\Sigma,s'}$. If $s' \in S_1$, then f is an S_1-constructor and thus $f \circ t : dom \to s'$ is a maximal Σ-generator. Hence $f^B(b) = f(t^B(a)) = (f \circ t)^B(a) \in reach^B_{\Sigma,s'}$.

[13] This notion is used in [83].
[14] dto.

(2) Let $s \in S$, $f : s \to s' \in F$ and $(a,b) \in obs^B_{\Sigma,s}$. Then for all $t : s \to ran \in MObs_\Sigma$, $t^B(a) = t^B(b)$. If $s, s' \in S_0$, then id_s is a maximal Σ-observer and thus $a = id^B_s(a) = id^B_s(b) = b$. Hence $f^B(a) = f^B(b)$ and thus $(f^B(a), f^B(b)) \in obs^B_{\Sigma,s'}$ because kernels are reflexive. If $s' \in S_1$, then f is an S_1-destructor and thus for all $t : s' \to ran \in MObs_\Sigma$, $t \circ f : s \to ran$ is a maximal Σ-observer. Hence $t^B(f^B(a)) = (t \circ f)^B(a) = (t \circ f)^B(b) = t^B(f^B(b))$ and thus $(f^B(a), f^B(b)) \in obs^B_{\Sigma,s'}$.

(3) Let B be Σ-complete and $a \in s^{Cofree(\Sigma,A)}$. Hence $obs^B_s(b) = a$ for some $b \in B$ and thus by Proposition 4.6(5),

$$obs^C_s(h(b)) = \langle t^C\rangle_{t\in MObs_{\Sigma,s}}(h(b)) = \langle t^C \circ h\rangle_{t\in MObs_{\Sigma,s}}(b) = \langle t^B\rangle_{t\in MObs_{\Sigma,s}}(b) = obs^B_s(b) = a.$$

Hence C is Σ-complete.

Let C be Σ-consistent, $t : s_1 \to s, u : s_2 \to s \in MGen_\Sigma$, $a \in s_1^A$ and $b \in s_2^A$ such that $t^B(a) = u^B(b)$. By Proposition 4.6(4), $t^C(a) = h(t^B(a)) = h(u^B(b) = u^C(b)$ and thus $t^{Free(\Sigma,A)}(a) = u^{Free(\Sigma,A)}(b)$ because C is Σ-consistent. We conclude that B be Σ-consistent.

(4) Let B be Σ-reachable and $c \in s^C$. Since h is surjective, $c = h(b)$ for some $b \in s^B$. Since B is Σ-reachable, $b = u^B(a)$ for some $u : dom \to s \in MGen_\Sigma$ and $a \in dom^A$. Hence by Proposition 4.6(4), $c = h(b) = h(u^B(a)) = u^C(a)$ and thus C is Σ-reachable.

Let B be Σ-observable and $a, b \in s^C$ such that $obs^C_s(a) = obs^C_s(b)$. Since h is surjective, $a = h(a')$ and $b = h(b')$ for some $a', b' \in s^B$. Hence by Proposition 4.6(5),

$$\begin{aligned}obs^B_s(a') = \langle t^B\rangle_{t\in MObs_{\Sigma,s}}(a') = \langle t^C \circ h\rangle_{t\in MObs_{\Sigma,s}}(a') = \langle t^C\rangle_{t\in MObs_{\Sigma,s}}(a) = obs^C_s(a) = \\ obs^C_s(b) = \langle t^C\rangle_{t\in MObs_{\Sigma,s}}(b) = \langle t^C \circ h\rangle_{t\in MObs_{\Sigma,s}}(b') = \langle t^B\rangle_{t\in MObs_{\Sigma,s}}(b') = obs^B_s(b')\end{aligned}$$

and thus $a' = b'$ because B be Σ-observable.

Let C be Σ-complete and $a \in s^{Cofree(\Sigma,A)}$. Hence $obs^C_s(c) = a$ for some $c \in C$. Since h is surjective, $c = h(b)$ for some $b \in s^B$, and thus by Proposition 4.6(5),

$$obs^B_s(b) = \langle t^B\rangle_{t\in MObs_{\Sigma,s}}(b) = \langle t^C \circ h\rangle_{t\in MObs_{\Sigma,s}}(b) = \langle t^C\rangle_{t\in MObs_{\Sigma,s}}(c) = a.$$

Hence B is Σ-complete.

(5) Let B be Σ-consistent, $t : s_1 \to s, u : s_2 \to s \in MGen_\Sigma$, $a \in s_1^A$ and $b \in s_2^A$ such that $t^C(a) = u^C(b)$. By Proposition 4.6(4), $h(t^B(a)) = t^C(a) = u^C(b) = h(u^B(b))$ and thus $t^B(a) = u^B(b)$ because h is injective. Hence $t^{Free(\Sigma,A)}(a) = u^{Free(\Sigma,A)}(b)$ because B is Σ-consistent. We conclude that C be Σ-consistent.

Let C be Σ-reachable and $b \in s^B$. Then $h(b) = u^C(a)$ for some $u : dom \to s \in MGen_\Sigma$ and $a \in dom^A$. Hence by Proposition 4.6(4), $h(b) = u^C(a) = h(u^B(a))$ and thus $b = u^B(a)$ because h is injective. We conlude that B is Σ-reachable.

Let C be Σ-observable and $a, b \in s^B$ such that $obs^B_s(a) = obs^B_s(b)$. Hence by Proposition 4.6(5),

$$\begin{aligned}obs^C_s(h(a)) = \langle t^C\rangle_{t\in MObs_{\Sigma,s}}(h(a)) = \langle t^C \circ h\rangle_{t\in MObs_{\Sigma,s}}(a) = \langle t^B\rangle_{t\in MObs_{\Sigma,s}}(a) = obs^B_s(a) = \\ obs^B_s(b) = \langle t^B\rangle_{t\in MObs_{\Sigma,s}}(b) = \langle t^C \circ h\rangle_{t\in MObs_{\Sigma,s}}(b) = \langle t^C\rangle_{t\in MObs_{\Sigma,s}}(h(b)) = obs^C_s(h(b))\end{aligned}$$

and thus $h(a) = h(b)$ because C is Σ-observable. Since h is injective, $a = b$, and we conclude that B is Σ-observable. ❏

Lemma 10.8 *Let the assumptions of Def. 10.6 hold true, $\Sigma = \Sigma'$, $\mathcal{C}$ be a class of Σ-structures and Ini be initial in $\mathcal{C}$ or Fin be final in $\mathcal{C}$. Let $B \in \mathcal{C}$, g be the unique Σ-homomorphism from Ini to B or h be the unique Σ-homomorphism B to Fin, respectively.*

(1) $reach^B_\Sigma \subseteq img(g)$.

(2) $ker(h) \subseteq obs_\Sigma^B$.

(3) *Ini is Σ-reachable. B is Σ-reachable iff g is surjective. If B is Σ-consistent, then g is injective. If g is injective and $\mathcal{C} = Mod_{EU}(\Sigma, A)$ for some set A, then B is Σ-consistent.*

(4) *Fin is Σ-observable. B is Σ-observable iff h is injective. If B is Σ-complete, then h is surjective. If h is surjective and $\mathcal{C} = Mod_{EU}(\Sigma, A)$ for some set A, then B is Σ-complete.*

Proof. (1) Let $b \in reach_\Sigma^B$. Then $b = t^A(a)$ for some $t : dom \to s \in MGen_\Sigma$ and $a \in dom^A$. Hence by Proposition 4.6(4), $b = t^A(a) = g(t^{Ini}(a)) \in img(g)$.

(2) Let $(a, b) \in ker(h)$ and $t : s \to ran \in MObs_\Sigma$. Then $h(a) = h(b)$ and thus by Proposition 4.6(5), $t^A(a) = t^{Fin}(h(a)) = t^{Fin}(h(b)) = t^A(b)$. Hence $(a, b) \in obs_\Sigma^B$.

(3) By Lemma 10.7(1), $reach_\Sigma^{Ini}$ is a Σ-invariant and thus $B = Ini|reach_\Sigma^{Ini}$ is a substructure of Ini. By Lemma 6.2(3), B is not a proper substructure of Ini. Hence for all $s \in S$, $reach_s^{Ini} = s^{Ini}$ and thus Ini is Σ-reachable. If B is Σ-reachable, then by (1), $B = B|reach^B \subseteq img(g)$ and thus g is surjective. If g is surjective, then by Lemma 10.7(3), B is Σ-reachable because Ini is Σ-reachable.

Suppose that B is Σ-consistent. By Proposition 4.6(4), for all $t \in MGen_\Sigma$, $g \circ t^{Ini} = t^B$. Since Ini is Σ-reachable, this equation defines g. Hence g is injective iff for all $t, u \in MGen_\Sigma$ and $a, b \in A$, $t^B(a) = u^B(b)$ implies $t^{Ini}(a) = u^{Ini}(b)$. So let $t, u \in MGen_\Sigma$ and $a, b \in A$ such that $t^B(a) = u^B(b)$. Since B is Σ-consistent, $t^{Free(\Sigma,A)}(a) = u^{Free(\Sigma,A)}(b)$. By Theorem 6.3, $Free(\Sigma, A)$ is initial in $Mod_{EU}(\Sigma, A)$. Hence there is a unique Σ-homomorphism $g' : Free(\Sigma, A) \to Ini$. By Proposition 4.6(4),

$$t^{Ini}(a) = g'(t^{Free(\Sigma,A)}(a)) = g'(u^{Free(\Sigma,A)}(b)) = u^{Ini}(b).$$

We have shown that g is injective.

Conversely, suppose that g is injective and $\mathcal{C} = Mod_{EU}(\Sigma, A)$. Then by Theorem 6.3, $Ini = Free(\Sigma, A)$ and thus $t^B(a) = u^B(b)$ implies

$$t^{Free(\Sigma,A)}(a) = t^{Ini}(a) = u^{Ini}(b) = u^{Free(\Sigma,A)}(b)$$

because g is injective. Hence B is Σ-consistent.

(4) By Lemma 10.7(2), obs_Σ^{Fin} is a Σ-congruence and thus a $B = Fin/obs_\Sigma^{Fin}$ is a quotient of Fin. By Lemma 6.2(4), B is not a proper quotient of Fin. Hence $obs_\Sigma^{Fin} = \Delta^{Fin}$ and thus Fin is Σ-observable. If B is Σ-observable, then by (2), $ker(h) \subseteq obs_\Sigma^B = \Delta^B$ and thus h is injective. If h is injective, then by Lemma 10.7(2), B is Σ-observable because Fin is Σ-observable.

Suppose that B is Σ-complete. By Proposition 4.6(5), for all $t \in Obs_\Sigma$, $t^{Fin} \circ h = t^B$. Since Fin is Σ-observable, this equation defines h. Hence h is surjective iff for all $c \in Fin$ there is $b \in B$ such that for all $t \in MObs_\Sigma$, $t^B(b)(= t^{Fin}(h(b))) = t^{Fin}(c)$. So let $c \in Fin$. By Theorem 6.4, $Cofree(\Sigma, A)$ is final in $Mod_{EU}(\Sigma, A)$. Hence there is a unique Σ-homomorphism $h' : Fin \to Cofree(\Sigma, A)$. Since B is Σ-complete, there is $b \in B$ such that for all $t : s \to ran \in MObs_\Sigma$, $t^B(b) = h'(c)_t = \pi_t(h'(c)) = t^{Fin}(c)$ (see Theorem 6.4(2)). We have shown that h is surjective.

Conversely, suppose that h is surjective and $\mathcal{C} = Mod_{EU}(\Sigma, A)$. Then by Theorem 6.4, $Fin = Cofree(\Sigma, A)$. Let $s \in S$ and $c = (a_t)_{t:s\to ran\in MObs_\Sigma} \in s^{Cofree(\Sigma,A)} = s^{Fin}$. Since h is surjective, $c = h(b)$ for some $b \in B$. Hence by Proposition 4.6(5) and Theorem 6.4(2), $t^B(b) = t^{Fin}(c) = t^{Cofree(\Sigma,A)}(c) = \pi_t(c) = a_t$ for all $t : s \to ran \in MObs_\Sigma$. Hence B is Σ-complete. ❏

Theorem 10.9 *(abstraction models) Let $SP' = (\Sigma', AX')$ be a swinging type with base type $SP = (\Sigma, AX)$, $\Sigma = (S_0, S, F, R)$, $\Sigma' = (S_0, S, F', R')$, σ be the relation transformer defined by $AX' \setminus AX$, Ini' be initial in $Mod(\Sigma', AX)$ and Φ be the (Ini', σ)-step functor.*

(1) If SP' is a visible abstraction, then $lfp(\Phi)/{\equiv^{lfp(\Phi)}}$ is initial in $Mod_{EU}(SP')$.

(2) If SP' is a hidden abstraction, then $gfp(\Phi)/{\equiv^{gfp(\Phi)}}$ is final in $RMod_{EU}(SP')$.

Proof. Let $equals = \{\equiv_s \mid s \in S_1\}$.

(1) By Lemma 10.2(1) and Theorem 10.3, $lfp(\Phi) \models AX'$ implies $lfp(\Phi)/{\equiv^{lfp(\Phi)}} \models AX'$ and thus $lfp(\Phi)/{\equiv^{lfp(\Phi)}} \in Mod_{EU}(SP')$. Let $B \in Mod_{EU}(SP')$. Hence $B \in Mod(\Sigma', AX)$ and thus there is a unique Σ'-homomorphism $h : Ini' \to B$.

Let A be the Σ'-structure that agrees with Ini' except for the interpretation of $R_1 \cup equals$: for all $r : s \in R_1 \cup equals$, $r^A =_{def} \{a \in Ini'_s \mid h(a) \in r^B\}$. Hence for all $s \in S$, $\equiv^A = ker(h)$ because B is a structure with equality.

Suppose that for all $r \in R_1 \cup equals$, $r^{lfp(\Phi)}$ is a subset of r^A. In particular, for all $s \in S_1$, $\equiv_s^{lfp(\Phi)} \subseteq \equiv_s^A$ and thus for all $a, b \in Ini'_s$, $a \equiv^{lfp(\Phi)} b$ implies $h(a) = h(b)$. Hence $g : lfp(\Phi)/{\equiv^{lfp(\Phi)}} \to B$ with $g([a]) =_{def} h(a)$ for all $a \in Ini'$ is well-defined. Therefore, $h = g \circ nat$. Since nat is epimorphic and h is homomorphic, Lemma 4.16(1) implies that g is homomorphic, too. Moreover, let g' be any Σ-homomorphism from $lfp(\Phi)/{\equiv^{lfp(\Phi)}}$ to B. Since h is the only Σ-homomorphism from Ini' to B, we obtain $g' \circ nat = h = g \circ nat$ and thus $g' = g$ because nat is surjective. Hence $lfp(\Phi)/{\equiv^{lfp(\Phi)}}$ is initial in $Mod_{EU}(SP')$.

It remains to show that for all $r : s \in R_1 \cup equals$, $r^{lfp(\Phi)}$ is a subset of r^A. By Theorem 8.4(1), $r^{lfp(\Phi)} \subseteq r^A$ follows from $r^{\Phi(A)} \subseteq r^A$. So let $a \in r^{\Phi(A)} = \sigma(r)^A$. Since for all $q \in R'$, $q^A = \{a \in Ini'_s \mid h(a) \in q^B\}$, $a \in \sigma(r)^A$ implies $h(a) \in \sigma(r)^B$. Since B satisfies $r \Leftarrow \sigma(r)$, $h(a) \in \sigma(r)^B$ implies $h(a) \in r^B$, i.e., $a \in r^A$.

(2) By Lemma 10.2(2) and (5) and Theorem 10.3, $gfp(\Phi) \models AX'$ implies $B = gfp(\Phi)/{\equiv^{gfp(\Phi)}} \models AX'$ and thus $B \in Mod_{EU}(SP')$. Since Ini' is initial in $Mod(\Sigma', AX)$, $gfp(\Phi)$ is so, too, and thus by Lemma 10.8(3), $gfp(\Phi)$ is Σ'-reachable. Hence there is unique Σ'-homomorphism h from $gfp(\Phi)$ to B that must agree with the natural mapping. Therefore, h is surjective and thus, again by Lemma 10.8(3), B is Σ'-reachable. We conclude that $B \in RMod_{EU}(SP')$.

Let $B \in RMod_{EU}(SP')$. Hence $B \in Mod_{EU}(SP)$ and thus there is a unique Σ-homomorphism $h : Ini' \to B$. Let A be the Σ'-structure that agrees with Ini' except for the interpretation of $R_1 \cup equals$: for all $r : s \in R_1 \cup equals$, $r^A =_{def} \{a \in Ini'_s \mid h(a) \in r^B\}$. Hence for all $s \in S$, $\equiv^A = ker(h)$ because B is a Σ-structure with equality.

Suppose that for all $r \in R_1 \cup equals$, r^A is a subset of $r^{gfp(\Phi)}$. In particular, for all $s \in S_1$, $\equiv_s^A \subseteq \equiv_s^{gfp(\Phi)}$ and thus for all $a, b \in Ini'_s$, $h(a) = h(b)$ implies $a \equiv^{gfp(\Phi)} b$. By Lemma 10.8(3), h is surjective because B is reachable. Hence $g : B \to gfp(\Phi)/{\equiv^{gfp(\Phi)}}$ with $g(h(a)) =_{def} [a]$ for all $a \in Ini'$ is well-defined. Therefore, $g \circ h = nat$. Since h is epimorphic and nat is homomorphic, Lemma 4.16(1) implies that g is homomorphic, too. Moreover, let g' be any Σ-homomorphism from B to $gfp(\Phi)/{\equiv^{gfp(\Phi)}}$. Since nat is the only Σ-homomorphism from Ini' to $gfp(\Phi)/{\equiv^{gfp(\Phi)}}$, we obtain $g' \circ h = nat = g \circ h$ and thus $g' = g$ because h is surjective. Hence $gfp(\Phi)/{\equiv^{gfp(\Phi)}}$ is final in $RMod_{EU}(SP')$.

It remains to show that for all $r : s \in R_1 \cup equals$, r^A is a subset of $r^{gfp(\Phi)}$. By Theorem 8.4(2), $r^A \subseteq r^{gfp(\Phi)}$ follows from $r^A \subseteq r^{\Phi(A)}$. So let $a \in r^A$. Hence $h(a) \in r^B$ and thus $h(a) \in \sigma(r)^B$ because B satisfies $r \Rightarrow \sigma(r)$. Since for all $q \in R'$, $q^A = \{a \in Ini'_s \mid h(a) \in q^B\}$, $h(a) \in \sigma(r)^B$ implies $a \in \sigma(r)^A = r^{\Phi(A)}$. ❑

Theorem 10.10 *(restriction models) Let $SP' = (\Sigma', AX')$ be a swinging type with base type $SP = (\Sigma, AX)$, $\Sigma = (S_0, S, F, R)$, $\Sigma' = (S_0, S, F', R')$, σ be the relation transformer defined by $AX' \setminus AX$, Fin' be final in $Mod(\Sigma', AX)$ and Φ be the (Fin', σ)-step functor.*

(1) If SP' is a hidden restriction, then $gfp(\Phi)|all^{gfp(\Phi)}$ is final in $Mod_{EU}(SP')$.

(2) If SP' is a visible restriction, then $lfp(\Phi)|all^{lfp(\Phi)}$ is initial in $OMod_{EU}(SP')$.

Proof. Let $univs = \{all_s \mid s \in S_1\}$.

(1) By Lemma 10.2(4) and Theorem 10.4, $gfp(\Phi) \models AX'$ implies $gfp(\Phi)|all^{gfp(\Phi)} \models AX'$ and thus $gfp(\Phi)|all^{gfp(\Phi)} \in Mod_{EU}(SP')$. Let $B \in Mod_{EU}(SP')$. Hence $B \in Mod(\Sigma', AX)$ and thus there is a unique Σ'-homomorphism $h : B \to Fin'$.

Let A be the Σ'-structure that agrees with Fin' except for the interpretation of $R_1 \cup univs$: for all $r \in R_1 \cup univs$, $r^A =_{def} h(r^B)$. Hence for all $s \in S$, $all_s^A = h(all_s^B) = h(s^B)$ because B is a structure with universe.

Suppose that for all $r \in R_1 \cup univs$, r^A is a subset of $r^{gfp(\Phi)}$. In particular, for all $s \in S_1$, $h(s^B) = all_s^A \subseteq all_s^{gfp(\Phi)}$. Hence $g : B \to gfp(\Phi)|all^{gfp(\Phi)}$ with $g(a) =_{def} h(a)$ for all $a \in B$ is well-defined. Therefore, $h = inc \circ g$. Since inc is monomorphic and h is homomorphic, Lemma 4.17(1) implies that g is homomorphic, too. Moreover, let g' be any Σ-homomorphism from B to $gfp(\Phi)|all^{gfp(\Phi)}$. Since h is the only Σ-homomorphism from B to Fin', we obtain $inc \circ g' = h = inc \circ g$ and thus $g' = g$ because inc is injective. Hence $gfp(\Phi)|all^{gfp(\Phi)}$ is final in $Mod_{EU}(SP')$.

It remains to show that for all $r : s \in R_1 \cup univs$, r^A is a subset of $r^{gfp(\Phi)}$. By Theorem 8.4(2), $r^A \subseteq r^{gfp(\Phi)}$ follows from $r^A \subseteq r^{\Phi(A)}$. So let $a \in r^A$. Hence $a = h(b)$ for some $b \in r^B$ and thus $b \in \sigma(r)^B$ because B satisfies $r \Rightarrow \sigma(r)$. Since for all $q \in R'$, $q^A = h(q^B)$, $b \in \sigma(r)^B$ implies $a = h(b) \in \sigma(r)^A = r^{\Phi(A)}$.

(2) By Lemma 10.2(3) and Theorem 10.4, $lfp(\Phi) \models AX'$ implies $B = lfp(\Phi)|all^{lfp(\Phi)} \models AX'$ and thus $B \in Mod_{EU}(SP')$. Since Fin' is final in $Mod(\Sigma', AX)$, $lfp(\Phi)$ is so, too, and thus by Lemma 10.8(4), $lfp(\Phi)$ is Σ'-observable. Hence there is unique Σ'-homomorphism h from B to $lfp(\Phi)$ that must agree with the inclusion mapping. Therefore, h is injective and thus, again by Lemma 10.8(4), B is Σ'-observable. We conclude that $B \in OMod_{EU}(SP')$.

Let $B \in OMod_{EU}(SP')$. Hence $B \in Mod_{EU}(SP')$ and thus there is a unique Σ-homomorphism $h : B \to Fin'$. Let A be the Σ'-structure that agrees with Fin' except for the interpretation of $R_1 \cup univs$: for all $r \in R_1 \cup univs$, $r^A =_{def} h(r^B)$. Hence for all $s \in S$, $all_s^A = h(all_s^B) = h(s^B)$ because B is a Σ-structure with universe.

Suppose that for all $r \in R_1 \cup univs$, $r^{lfp(\Phi)}$ is a subset of r^A. In particular, for all $s \in S_1$, $all_s^{lfp(\Phi)} \subseteq all_s^A = h(s^B) \subseteq Fin'$. By Lemma 10.8(4), h is injective because B is observable. Hence $g : lfp(\Phi)|all^{lfp(\Phi)} \to B$ with $g(h(b)) =_{def} b$ for all $b \in h^{-1}(all^{lfp(\Phi)})$ is well-defined. Therefore, $h \circ g = inc$. Since h is monomorphic and inc is homomorphic, Lemma 4.16(2) implies that g is homomorphic, too. Moreover, let g' be any Σ-homomorphism from $lfp(\Phi)|all^{lfp(\Phi)}$ to B. Since inc is the only Σ-homomorphism from $all_s^{lfp(\Phi)}$ to Fin', we obtain $h \circ g' = inc = h \circ g$ and thus $g' = g$ because h is injective. Hence $lfp(\Phi)|all^{lfp(\Phi)}$ is initial in $OMod_{EU}(SP')$.

It remains to show that for all $r : s \in R_1 \cup univs$, $r^{lfp(\Phi)}$ is a subset of r^A. By Theorem 8.4(1), $r^{lfp(\Phi)} \subseteq r^A$ follows from $r^{\Phi(A)} \subseteq r^A$. So let $a \in r^{\Phi(A)} = \sigma(r)^A$. Since for all $q \in R'$, $q^A = h(q^B)$, $a \in \sigma(r)^A$ implies $b \in \sigma(r)^B$ for some $b \in B$ with $h(b) = a$. Hence $b \in r^B$ and thus $a = h(b) \in r^A$ because B satisfies $r \Leftarrow \sigma(r)$. ❑

11 Conservative extension

Lemma 11.1 *Let the assumptions of Def. 10.6 hold true, $\mathcal{C}$ be a class of Σ-structures and B be a Σ'-structure.*

(1) Let Ini be initial in $\mathcal{C}$. If B is Σ-reachable and Σ-consistent, then the S-sorted function $abs : B \to Ini$ mapping $b \in B$ to $t^{Ini}(a)$ for some $t \in MGen_\Sigma$ with $t^B(a) = b$ is well-defined and surjective. Moreover, Ini can be extended to a Σ'-structure such that abs becomes Σ'-homomorphic. If $B|_\Sigma \in \mathcal{C}$, then abs is also injective and thus a Σ'-isomorphism.

(2) Let Fin be final in $\mathcal{C}$. If B is Σ-observable and Σ-complete, then the S-sorted function $rep : Fin \to B$ mapping $a \in Fin$ to $b \in B$ with $obs_s^B(b) = obs_s^{Fin}(a)$ is well-defined and injective. Moreover, Fin can be

extended to a Σ'-structure such that rep becomes Σ'-homomorphic. If $B|_\Sigma \in \mathcal{C}$, then rep is also surjective and thus a Σ'-isomorphism.

Proof. (1) Let $b \in B$. Since B is Σ-reachable, there are $t \in MGen_\Sigma$ and $a \in A$ such that $t^B(a) = b$. Suppose that $t^B(a) = u^B(b)$ for some $t, u \in MGen_\Sigma$ and $a, b \in A$. Since B is Σ-consistent, $t^{Free(\Sigma,A)}(a) = u^{Free(\Sigma,A)}(b)$. By Theorem 6.3, $Free(\Sigma, A)$ is initial in $Mod_{EU}(\Sigma, A)$. Hence by Proposition 4.6(4), the unique Σ-homomorphism from $Free(\Sigma, A)$ to Ini maps $t^{Free(\Sigma,A)}(a)$ to $t^{Ini}(a)$. Hence $t^{Free(\Sigma,A)}(a) = u^{Free(\Sigma,A)}(b)$ implies $t^{Ini}(a) = u^{Ini}(b)$. We conclude that abs is well-defined. By Lemma 10.8(3), Ini is Σ-reachable. Hence for all $c \in Ini$ there are $t \in MGen_\Sigma$ and $a \in A$ such that $t^{Ini}(a) = c$ and thus abs is surjective.

$f : dom \to s \in F_1$ is interpreted in Ini as follows. Since Ini is Σ-reachable, f^{Ini} is well-defined by $f^{Ini}(t^{Ini}(a)) = abs((f \circ t)^B(a))$ for all $t : s_0 \to dom \in MGen_\Sigma$ and $a \in s_0^A$. Let $b \in B$. Since B is Σ-reachable, there are $t \in MGen_\Sigma$ and $a \in A$ such that $t^B(a) = b$. Hence abs is compatible with f:

$$abs(f^B(b)) = abs(f^B(t^B(a))) = abs((f \circ t)^B(a)) = f^{Ini}(t^{Ini}(a)) = f^{Ini}(abs(t^B(a))) = f^{Ini}(abs(b)).$$

We conclude that abs is Σ'-homomorphic. For all $r \in R' \setminus R$, $r^{Ini} = abs(r^B)$.

Suppose that $B|_\Sigma \in \mathcal{C}$ and $abs(b) = abs(c)$ for some $b, c \in B$. Then $t^{Ini}(a) = t^{Ini}(a')$ for some $a, a' \in A$ with $t^B(a) = b$ and $t^B(a') = c$. By assumption, there is a unique Σ-homomorphism h from Ini to $B|_\Sigma$. Hence by Proposition 4.6(4), $b = t^B(a) = h(t^{Ini}(a)) = h(t^{Ini}(a')) = t^B(a') = c$ and thus abs is injective.

(2) Let $s \in S$ and $c \in s^{Fin}$. By Theorem 6.4, $Cofree(\Sigma, A)$ is final in $Mod_{EU}(\Sigma, A)$. Let h be the unique Σ-homomorphism from Fin to $Cofree(\Sigma, A)$. Since B is Σ-complete, $obs_s^B(b) = h(c)$ for some $b \in B$. Hence by Proposition 4.6(5), for all $t \in MObs_{\Sigma,s}$,

$$\pi_t(obs_s^B(b)) = \pi_t(h(c)) = t^{Cofree(\Sigma,A)}(h(c)) = t^{Fin}(a) = \pi_t(obs_s^{Fin}(c))$$

and thus $obs_s^B(b) = obs_s^{Fin}(c)$. Suppose that $obs_s^B(b) = obs_s^B(c)$ for some $b, c \in s^B$. Since B is Σ-observable, $b = c$. We conclude that rep is well-defined. By Lemma 10.8(4), Fin is Σ-observable. Hence for all $b, c \in Fin$, $obs_s^{Fin}(b) = obs_s^{Fin}(c)$ implies $b = c$, and thus rep is injective.

$f : s \to ran \in F_1$ is interpreted in Fin as follows. Since Fin is Σ-observable, f^{Fin} is well-defined by $t^{Fin}(f^{Fin}(c)) = (t \circ f)^B(rep(c))$ for all $t \in MObs_{\Sigma,ran}$ and $c \in s^{Fin}$. Hence for all $c \in s^{Fin}$,

$$obs_{ran}^B(rep(f^{Fin}(c))) = t^{Fin}(f^{Fin}(c)) = (t \circ f)^B(rep(c)) = obs_{ran}^B(f^B(rep(c)))$$

and thus $rep(f^{Fin}(c)) = f^B(rep(c))$ because B is Σ-observable. We conclude that rep is Σ'-homomorphic. For all $r : s \in R' \setminus R$, $r^{Fin} = \{c \in s^{Fin} \mid rep(c) \in r^B\}$.

Suppose that $B|_\Sigma \in \mathcal{C}$, $s \in S$ and $b \in s^B$. By assumption, there is a unique Σ-homomorphism h from $B|_\Sigma$ to Fin. Hence by Proposition 4.6(5), for all $t \in MObs_{\Sigma,s}$,

$$\pi_t(obs_s^B(b)) = t^B(b) = t^{Fin}(h(b)) = \pi_t(obs_s^{Fin}(h(b)))$$

and thus $obs_s^B(b) = obs_s^{Fin}(h(b))$. Hence $b = rep(h(b))$ and thus rep is surjective. ❏

Theorem 11.2 *Let the assumptions of Def. 10.6 hold true and $F_1 = F' \setminus F$.*

(1) Suppose that F_1 consists of S_1-constructors and B is Σ'-reachable.
B is Σ-reachable iff $reach_\Sigma^B$ is F_1-compatible.

(2) Suppose that F_1 consists of S_1-destructors and B is Σ'-observable.
B is Σ-observable iff obs_Σ^B is F_1-compatible.

Proof. (1) The "only-if"-direction is trivial. Let $s \in S$ and $b \in s^B$. Since B is Σ'-reachable, $b = t^B(a)$ for some $t : dom \to s \in MGen_{\Sigma'}$ and $a \in dom^A$. We show $b \in reach_\Sigma^B$ by induction on (m, n) where m and n are the numbers of occurrences of F_1-symbols resp. F'-symbols in t. We start with two basic cases: (a) t consists of F-symbols and (b) $t = f \circ u$ for some $f \in F_1$ and $u \in MGen_\Sigma$. In case (a), t is a Σ-generator and thus the proof is complete. In case (b), $u^B(a) \in reach^B_\Sigma$. Hence by assumption, $(f \circ u)^B(a) = f^B(u^B(a)) \in reach^B_\Sigma$ and thus $t^B(a) = (f \circ u)^B(a) = w^B(c)$ for some $w \in MGen_\Sigma$ and $c \in dom^A_w$. Again, the proof is complete.

If neither case (a) nor case (b) holds true, then $t = v \circ \langle u_i \rangle_{i \in I}$ for some $\{v\} \cup \{u_i : dom \to s_i\}_{i \in I} \subseteq Gen_{\Sigma'}$ such that for some $k \in I$, u_k contains at least one F_1-symbol. Since u_k is a subterm of t, the induction hypothesis implies $u_k^B(a) \in reach^B_\Sigma$, i.e., $u_k^B(a) = w^B(c)$ for some $w \in MGen_{\Sigma,s_k}$ and $c \in dom^A_w$. Hence

$$\begin{aligned} b = t^B(a) = (v \circ \langle u_i \rangle_{i \in I})^B(a) = v^B(\langle u_i^B \rangle_{i \in I}(a)) = v^B((u_i^B(a))_{i \in I}) = \\ v^B((v_i^B(a,c))_{i \in I}) = v^B(\langle v_i^B \rangle_{i \in I}(a,c)) = (v \circ \langle v_i \rangle_{i \in I})^B(a,c) \end{aligned} \tag{3}$$

where for all $i \in I$,

$$v_i = \begin{cases} w \circ \pi_2 : dom \times dom_w \to s_k & \text{if } i = k, \\ u_i \circ \pi_1 : dom \times dom_w \to s_i & \text{if } i \neq k. \end{cases}$$

Since $v \circ \langle v_i \rangle_{i \in I}$ contains less occurrences of F_1-symbols than t, the induction hypothesis implies $b = (v \circ \langle v_i \rangle_{i \in I})^B(a,c) \in reach^B_\Sigma$. We conclude that B is Σ-reachable.

(2) The "only-if"-direction is trivial. Let $s \in S$ and $a, b \in s^B$ such that $a \neq b$. Since B is Σ'-observable, there are $t : s \to ran \in MObs_{\Sigma'}$ such that $t^B(a) \neq t^B(b)$. We show $(a,b) \notin obs^B_\Sigma$ by induction on (m,n) where m and n are the numbers of occurrences of F_1-symbols resp. F'-symbols in t. We start with two basic cases: (a) t consists of F-symbols and (b) $t = u \circ f$ for some $f \in F_1$ and $u \in MObs_\Sigma$. In case (a), t is a Σ-observer and thus the proof is complete. In case (b), $u^B(f^B(a)) = (u \circ f)^B(a) = t^B(a) \neq t^B(b) = (u \circ f)^B(b) = u^B(f^B(b))$. Hence $u \in MObs_\Sigma$ implies $(f^B(a), f^B(b)) \notin obs^B_\Sigma$ and thus by assumption, $(a,b) \notin obs^B_\Sigma$. Again, the proof is complete.

If neither case (a) nor case (b) holds true, then $t = [u_i]_{i \in I} \circ v$ for some $\{u_i : s_i \to ran\}_{i \in I} \cup \{v\} \subseteq Obs_{\Sigma'}$ such that for some $k \in I$, u_k contains at least one F_1-symbol. Since $t^B(a) \neq t^B(b)$, there are $i, j \in I$, $c \in s_i^B$ and $d \in s_j^B$ such that $ran_v = \coprod_{i \in I} s_i$, $v^B(a) = (c,i)$, $v^B(b) = (d,j)$, $c \neq d$ and $u_i^B(c) \neq u_j^B(d)$.

Case 1: $i = j = k$. Then $u_k^B(c) \neq u_k^B(d)$. Since u_k is a superterm of t, the induction hypothesis implies $(c,d) \notin obs^B_{\Sigma'}$, i.e., $w^B(c) \neq w^B(d)$ for some $w \in MObs_{\Sigma,s_k}$. Hence

$$\begin{aligned} ([v_i]_{i \in I} \circ v)^B(a) = ([v_i^B]_{i \in I})(v^B(a)) = ([v_i^B]_{i \in I})(c,k) = v_k^B(c) = w^B(c) \neq w^B(d) = \\ v_k^B(d) = ([v_i^B]_{i \in I})(d,k) = ([v_i^B]_{i \in I})(v^B(b)) = ([v_i]_{i \in I} \circ v)^B(b) \end{aligned} \tag{4}$$

where for all $i \in I$,

$$v_i = \begin{cases} w : s_k \to ran_w & \text{if } i = k, \\ u_i : s_i \to ran_w & \text{if } i \neq k. \end{cases}$$

Case 2: $i \neq k$ or $j \neq k$. For all $i \in I$, let

$$v_i = \begin{cases} \iota_1 : s_k \to 1 + ran_w & \text{if } i = k, \\ \iota_2 \circ u_i : s_i \to 1 + ran_w & \text{if } i \neq k. \end{cases}$$

Hence

$$\begin{cases} v_i^B(c) = \iota_2(u_i^B(c)) \neq \iota_2(u_j^B(d)) = v_j^B(d) & \text{if } i \neq k \text{ and } j \neq k, \\ v_i^B(c) = \iota_1(c) \neq \iota_2(u_j^B(d)) = v_j^B(d) & \text{if } i = k, \\ v_i^B(c) = \iota_2(u_i^B(c)) \neq \iota_1(d) = v_j^B(d) & \text{if } j = k, \end{cases}$$

and thus

$$\begin{aligned} ([v_i]_{i \in I} \circ v)^B(a) = ([v_i^B]_{i \in I})(v^B(a)) = ([v_i^B]_{i \in I})(c,i) = v_i^B(c) \neq v_j^B(d) = \\ ([v_i^B]_{i \in I})(d,j) = ([v_i^B]_{i \in I})(v^B(b)) = ([v_i]_{i \in I} \circ v)^B(b). \end{aligned} \tag{5}$$

Since, in both cases, $[v_i]_{i\in I} \circ v$ contains less occurrences of F_1-symbols than t, the induction hypothesis implies $(a,b) \not\in obs_\Sigma^B$. We conclude that B is Σ-observable. ❑

Theorem 11.3 *(conservative model extension)*

(1) Let $SP' = (\Sigma', AX')$ be a visible abstraction with base type $SP = (\Sigma, AX)$, Ini be initial in $Mod_{EU}(SP)$, Ini' be initial in $Mod(\Sigma', AX)$, σ be the relation transformer defined by $AX' \setminus AX$ and Φ be the (Ini', σ)-step functor. Ini can be extended to an initial object of $Mod_{EU}(SP')$ if $B = lfp(\Phi)/{\equiv^{lfp(\Phi)}}$ is Σ-consistent and $reach_\Sigma^B$ is F_1-compatible. The converse holds true if SP satisfies 5.1(1).

(2) Let $SP' = (\Sigma', AX')$ be a hidden restriction with base type $SP = (\Sigma, AX)$, Fin be final in $Mod_{EU}(SP)$, Fin' be final in $Mod(\Sigma', AX)$, σ be the relation transformer defined by $AX' \setminus AX$ and Φ be the (Fin', σ)-step functor. Fin can be extended to a final object of $Mod_{EU}(SP')$ if $B = gfp(\Phi)|all^{gfp(\Phi)}$ is Σ-complete and obs_Σ^B is F_1-compatible. The converse holds true if SP satisfies 5.1(2).

Proof. (1) By Theorem 10.9(1), B is initial in $Mod_{EU}(SP')$. Suppose that B is Σ-consistent and $reach_\Sigma^B$ is F_1-compatible. By Lemma 10.8(3), B is Σ'-reachable and thus by Theorem 11.2(1), B is Σ-reachable because $reach_\Sigma^B$ is F_1-compatible. Since $B|_\Sigma \in Mod_{EU}(SP)$, Lemma 11.1(1) implies that Ini can be extended to a Σ'-structure that is Σ'-isomorphic to B and thus initial in $Mod_{EU}(SP')$.

Suppose that SP satisfies 5.1(1) and Ini is initial in $Mod_{EU}(SP')$. Then $Mod_{EU}(SP) = Mod_{EU}(\Sigma, A)$ for some set A. Moreover, B and Ini are Σ'-isomorphic. Since Ini is initial in $Mod_{EU}(SP) = Mod_{EU}(\Sigma, A)$, Lemma 10.8(3) implies that Ini is Σ-reachable and Σ-consistent. Since Ini and B are Σ'-isomorphic, Lemma 10.7(4/5) implies that B is also Σ-consistent and Σ-reachable, and thus $reach_\Sigma^B$ is F_1-compatible.

(2) By Theorem 10.10(1), B is final in $Mod_{EU}(SP')$. Suppose that B is Σ-complete and obs_Σ^B is F_1-compatible. By Lemma 10.8(4), B is Σ'-observable and thus by Theorem 11.2(2), B is Σ-observable because obs_Σ^B is F_1-compatible. Since $B|_\Sigma \in Mod_{EU}(SP)$, Lemma 11.1(2) implies that Fin can be extended to a Σ'-structure that is Σ'-isomorphic to B and thus final in $Mod_{EU}(SP')$.

Suppose that SP satisfies 5.1(2) and Fin is final in $Mod_{EU}(SP')$. Then $Mod_{EU}(SP) = Mod_{EU}(\Sigma, A)$ for some set A. Moreover, B and Fin are Σ'-isomorphic. Since Fin is final in $Mod_{EU}(SP) = Mod_{EU}(\Sigma, A)$, Lemma 10.8(4) implies that Fin is Σ-observable and Σ-complete. Since Fin and B are Σ'-isomorphic, Lemma 10.7(4/5) implies that B is also Σ-complete and Σ-observable, and thus obs_Σ^B is F_1-compatible. ❑

12 The perfect model

Definition 12.1 *(perfect model of a swinging type)* Let $SP = (\Sigma, AX)$ be a swinging type. The **perfect model of** SP, $Per(SP)$, is defined inductively as follows: If $SP = (\emptyset, \emptyset)$, then $Per(SP)$ is the empty Σ-structure. Otherwise

- $Per(SP)$ is the initial object of $Mod_{EU}(SP)$ if SP is visible, but not a visible restriction.
- $Per(SP)$ is the initial object of $OMod_{EU}(SP)$ if SP is a hidden abstraction.
- $Per(SP)$ is the final object of $RMod_{EU}(SP)$ if SP is a visible restriction.
- $Per(SP)$ is the final object of $Mod_{EU}(SP)$ if SP is hidden, but not a hidden abstraction. ❑

Theorems 6.3, 6.4, 7.1, 7.2, 10.9 and 10.10 ensure the existence of the perfect model. The following two lemmas provide conditions under which the quotients in Theorem 10.9 and the substructures in Theorem 10.10 are not proper:

Lemma 12.2 *(trivial quotients) Let $SP' = (\Sigma', AX')$ be a μ-extension of a swinging type $SP = (\Sigma, AX)$. Let Ini be initial in $Mod_{EU}(SP)$.*

(1) If $CONH$ is the set of all axioms of $AX_1 = AX' \setminus AX$ that do not include $\equiv_s$ for some $s \in S_1$, then for all $s \in S_1$, the least SP'-model over Ini interprets $\equiv_s$ as the diagonal of Ini_s^2.

(2) If $INVH \subseteq AX'$, then for all $s \in S_1$, the least (and only) SP'-model over Ini interprets all_s as Ini_s.

Proof. By Theorem 8.14, the least SP'-model B over Ini exists. Let $\Sigma = (S, F, R, S_0, C)$.

(1) Let D be the S-sorted set of all $a \in Ini$ such that $a \equiv^B a$. Since B satisfies $CONH$, $\equiv^B$ is Σ-congruent and thus D is a substructure of Ini: Let $f : s \to s' \in \Sigma$ and $a \in D_s$, i.e., $a \equiv^B a$. Then $f(a) \equiv^B f(a)$, i.e., $f(a) \in D_{s'}$. By Theorem 10.4, $D \in Mod_{EU}(SP)$. Hence by Lemma 6.2(3), $D = Ini$ and thus for all $a \in Ini$, $a \equiv^B a$, i.e., Δ^{Ini} is a subset of $\equiv^B$. Since Δ^{Ini} yields an interpretation of $\equiv$ in Ini that satisfies $CONH$ and $\equiv^B$ is the least one, $\equiv^B$ is a subset Δ^{Ini}. We conclude that both relations are equal, i.e., B interprets $\equiv_s$ as the diagonal of Ini_s^2.

(2) Since $Ini \in Mod_{EU}(SP)$ and B satisfies $INVH$, all^B is Σ-invariant and thus, by Theorem 10.4, can be turned into an SP-model over A with equality and universe such that the inclusion mapping inc from all^B to Ini is Σ-homomorphic. Since Ini is initial in $Mod_{EU}(SP)$, there are unique Σ-homomorphisms $h : Ini \to all^B$ and $h' : Ini \to Ini$. Hence $inc \circ h = id^{Ini}$, i.e., for all $a \in Ini$, $a = h(a) \in all^B$. Of course, all^B is a subset Ini. We conclude that both relations are equal, i.e., B interprets all as Ini. ❏

Lemma 12.3 *(trivial substructures) Let $SP' = (\Sigma', AX')$ be a ν-extension of a swinging type $SP = (\Sigma, AX)$ Let Fin be final in $Mod_{EU}(SP)$.*

(1) If $CONC \subseteq AX'$, then for all $s \in S_1$, the greatest (and only) SP'-model over Fin interprets $\equiv_s$ as the diagonal of Fin_s^2.

(2) If $INVC$ is the set of all axioms of $AX_1 = AX' \setminus AX$ that do not include all_s for some $s \in S_1$, then for all $s \in S_1$, the greatest SP'-model over Fin interprets all_s as Fin_s.

Proof. By Theorem 8.14, the greatest SP'-model B over Fin exists. Let $\Sigma = (S, F, R, S_0, C)$.

(1) Since B satisfies $CONC$, $\equiv^B$ is Σ-congruent. By Lemma 10.2(5), $\equiv^B$ is an R'-compatible equivalence relation. Hence Δ^{Fin} is a subset of $\equiv^B$ and the quotient $D =_{def} Fin/{\equiv^B}$ is well-defined. By Theorem 10.3, $D \in Mod_{EU}(SP)$. Hence by Lemma 6.2(4), $D \cong Fin$, i.e., $\equiv^B$ is contained in Δ^{Fin}. We conclude that both relations are equal, i.e., B interprets $\equiv_s$ as the diagonal of Fin_s^2.

(2) Since $Fin \in Mod_{EU}(SP)$ and B satisfies $INVC$, all^B is Σ-invariant and thus, by Theorem 10.4, can be turned into an SP-model over A with equality and universe such that the inclusion mapping inc from all^B to Fin is Σ-homomorphic. Since Fin is final in $Mod_{EU}(SP)$, there is a unique Σ-homomorphism $h : all^B \to Fin$. Of course, all^B is contained in Fin. Since Fin yields an interpretation of all in Fin that satisfies $INVC$ and all^B is the greatest one, Fin is a subset of all^B. We conclude that both relations are equal, i.e., B interprets all as Fin. ❏

Successive abstractions/restrictions induced by least or greatest congruences/invariants specified by Horn or co-Horn clauses can be combined to a single one:

Lemma 12.4 *(composition of abstractions) Let $SP_1 = (\Sigma_1, AX_1)$ and $SP_2 = (\Sigma_2, AX_1 \cup AX_2)$ be swinging types with base type $SP = (\Sigma, AX)$ resp. $SP_1 = (\Sigma_1, AX_1)$ such that SP_1 and SP_2 are (1) μ-extensions or (2) ν-extensions of SP resp. SP_1. Let σ_1, σ_2 and σ_3 be the relation transformers defined by $AX_1 \setminus AX$, $AX_2 \setminus AX_1$ and $AX_2 \setminus AX$, respectively. Let R be the relations defined by $AX_2 \setminus AX$, A be a $(\Sigma_2 \setminus R)$-structure and for $i = 1, 3$, let Φ_i be the (A, σ_i)-step functor, Φ_2 be the $(B_1/{\equiv^{B_1}}, \sigma_2)$-step functor and for $i = 1, 2, 3$, let (1) $B_i = lfp(\Phi_i)$ or (2) $B_i = gfp(\Phi_i)$.*

(1) If for $i = 1, 2, 3$, Φ_i is continuous, then $B_2/{\equiv^{B_2}}$ and $B_3/{\equiv^{B_3}}$ are Σ_2-isomorphic.

(2) If for $i = 1, 2, 3$, Φ_i is cocontinuous, then $B_2/{\equiv^{B_2}}$ and $B_3/{\equiv^{B_3}}$ are Σ_2-isomorphic.

Proof. (1) Let nat_1, nat_2 and nat_3 be the natural mappings from A to $B_1/{\equiv^{B_1}}$, from $B_1/{\equiv^{B_1}}$ to $B_2/{\equiv^{B_2}}$ and from A to $B_3/{\equiv^{B_3}}$, respectively.

$$\begin{tikzcd} A \arrow[r, two heads, "nat_3"] \arrow[d, two heads, "nat_1"'] \arrow[dr, phantom, "="] & B_3/{\equiv^{B_3}} \\ B_1/{\equiv^{B_1}} \arrow[r, two heads, "nat_2"'] & B_2/{\equiv^{B_2}} \arrow[u, dashed, "h"'] \end{tikzcd}$$

interpretations of a
non-primitive sort $s \in \Sigma$:
$s^{B_1} = s^A$
$s^{B_2} = s^A/{\equiv^{B_1}}$
$s^{B_3} = s^A$

Suppose that the function $h : B_2/{\equiv^{B_2}} \to B_3/{\equiv^{B_3}}$ with $h \circ nat_2 \circ nat_1 =_{def} nat_3$ is bijective. Since $nat_2 \circ nat_1$ is Σ_2-epimorphic and nat_3 is Σ_2-homomorphic, Lemma 4.16(1) implies that h is a Σ_2-isomorphism. For the bijectivity of h we must show that for all $a, b \in A$,

$$[a]_{\equiv^{B_1}} \equiv^{B_2} [b]_{\equiv^B} \quad \text{iff} \quad a \equiv^{B_3} b,$$

or, more generally, for all $r : s \in R$ and $a \in s^A$,

$$[a]_{\equiv^{B_1}} \in r^{B_2} \quad \text{iff} \quad a \in r^{B_3}. \tag{3}$$

Let $i = 1, 2, 3$. Since Φ_i is continuous, Theorem 8.3(2) implies $B_i = \sqcup_{j \in \mathbb{N}} \Phi_i^j(\bot)$. We start with the "only-if"-direction of (3) and show by induction on j that for all $j \in \mathbb{N}$,

$$[a]_{\equiv^{B_1}} \in r^{\Phi_2^j(\bot)} \quad \text{implies} \quad a \in r^{\Phi_3^k(\bot)} \tag{4}$$

for some $k \in \mathbb{N}$. Suppose that

$$a \in r^{B_1} \quad \text{implies} \quad a \in r^{\Phi_3^k(\bot)} \tag{5}$$

for some $k \in \mathbb{N}$. Let $[a]_{\equiv^{B_1}} \in r^{\Phi_2^j(\bot)}$. If $j = 0$, then $[a] \in r^\bot = r^{B_1/\equiv^{B_1}}$ and thus $a \in r^{B_1}$. Hence by (5), $a \in r^{\Phi_3^k(\bot)}$ for some $k \in \mathbb{N}$. If $j > 0$, then by the definition of Φ_2,

$$[a] \in \varphi_{r,AX_2}^{\Phi_2^{j-1}(\bot)}.$$

By induction hypothesis, for all relations q occurring in φ_{r,AX_2}, $[b]_{\equiv^{B_1}} \in q^{\Phi_2^{j-1}(\bot)}$ implies $b \in q^{\Phi_3^k(\bot)}$ for some $k \in \mathbb{N}$. Hence by the monotonicity of Φ_2,

$$a \in \varphi_{r,AX_2}^{\Phi_3^k(\bot)} \subseteq \varphi_{r,AX_3}^{\Phi_3^k(\bot)}$$

for some $k \in \mathbb{N}$ because $AX_2 \subseteq AX_3$ and thus for all Σ_2-structures C, $\varphi_{r,AX_2}^C \subseteq \varphi_{r,AX_3}^C$. By the definition of Φ_3, we conclude that $a \in r^{\Phi_3^{k+1}(\bot)}$. This finishes the proof of (4). It remains to show (5), i.e., that for all $j \in \mathbb{N}$,

$$a \in r^{\Phi_1^j(\bot)} \quad \text{implies} \quad a \in r^{\Phi_3^k(\bot)} \tag{6}$$

for some $k \in \mathbb{N}$. Let $a \in r^{\Phi_1^j(\bot)}$. If $j = 0$, then $a \in r^\bot = r^A = r^{\Phi_3^j(\bot)}$. If $j > 0$, then by the definition of Φ_1,

$$a \in \varphi_{r,AX_1}^{\Phi_1^{j-1}(\bot)}.$$

By induction hypothesis, for all relations q occurring in φ_{r,AX_1}, $b \in q^{\Phi_1^{j-1}(\bot)}$ implies $b \in q^{\Phi_3^k(\bot)}$ for some $k \in \mathbb{N}$. Hence by the monotonicity of Φ_1,

$$a \in \varphi_{r,AX_1}^{\Phi_3^k(\bot)} \subseteq \varphi_{r,AX_3}^{\Phi_3^k(\bot)}$$

for some $k \in \mathbb{N}$ because $AX_1 \subseteq AX_3$ and thus for all Σ_2-structures C, $\varphi_{r,AX_1}^C \subseteq \varphi_{r,AX_3}^C$. By the definition of Φ_3, we conclude that $a \in r^{\Phi_3^{k+1}(\bot)}$. This finishes the proof of (6) and thus of (5).

We proceed with the "if"-direction of (3) and show by induction on j that for all $j \in \mathbb{N}$,

$$a \in r^{\Phi_3^j(\bot)} \quad \text{implies} \quad [a]_{\equiv^{B_1}} \in r^{\Phi_2^k(\bot)} \tag{7}$$

for some $k \in \mathbb{N}$. Let $a \in r^{\Phi_3^j(\bot)}$. If $j = 0$, then $a \in r^\bot = r^A \subseteq r^{B_1}$ and thus $[a] \in r^{B_1/\equiv^{B_1}} = r^{\Phi_2^j(\bot)}$. If $j > 0$, then by the definition of Φ_3,

$$a \in \varphi_{r,AX_3}^{\Phi_3^{j-1}(\bot)}.$$

Since $AX_3 = AX_1 \cup AX_2$, there are two cases: (i) $a \in \varphi_{r,AX_1}^{\Phi_3^{j-1}(\bot)}$ and (ii) $a \in \varphi_{r,AX_2}^{\Phi_3^{j-1}(\bot)}$.

(i) By induction hypothesis, for all relations q occurring in φ_{r,AX_1}, $b \in q^{\Phi_3^{j-1}(\bot)}$ implies $[b] \in q^{\Phi_2^k(\bot)}$ for some $k \in \mathbb{N}$. Hence by the monotonicity of Φ_1, $[a] \in \varphi_{r,AX_1}^{\Phi_2^k(\bot)}$ and thus $[a] \in r^{\Phi_2^k(\bot)}$ because by Theorem 10.3, $B_1 \models r \Leftarrow \varphi_{r,AX_1}$ implies $\bot = B_1/\equiv^{B_1} \models r \Leftarrow \varphi_{r,AX_1}$ and thus $\Phi_2^k(\bot) \models r \Leftarrow \varphi_{r,AX_1}$.

(ii) By induction hypothesis, for all relations q occurring in φ_{r,AX_2}, $b \in q^{\Phi_3^{j-1}(\bot)}$ implies $[b] \in q^{\Phi_2^k(\bot)}$ for some $k \in \mathbb{N}$. Hence by the monotonicity of Φ_2, $[a] \in \varphi_{r,AX_2}^{\Phi_2^k(\bot)}$ for some $k \in \mathbb{N}$. By the definition of Φ_2, we conclude that $[a] \in r^{\Phi_2^{k+1}(\bot)}$. This finishes the proof of (7).

(2) can be shown analogously. ❏

Lemma 12.5 *(composition of restrictions) Let $SP_1 = (\Sigma_1, AX_1)$ and $SP_2 = (\Sigma_2, AX_1 \cup AX_2)$ be swinging types with base type $SP = (\Sigma, AX)$ resp. $SP_1 = (\Sigma_1, AX_1)$ such that SP_1 and SP_2 are (1) ν-extensions or (2) μ-extensions of SP resp. SP_1. Let σ_1, σ_2 and σ_3 be the relation transformers defined by $AX_1 \setminus AX$, $AX_2 \setminus AX_1$ and $AX_2 \setminus AX$, respectively. Let R be the relations defined by $AX_2 \setminus AX$ and A be a $(\Sigma_2 \setminus R)$-structure. For $i = 1, 3$, let Φ_i be the (A, σ_i)-step functor, Φ_2 be the $(B_1|all^{B_1}, \sigma_2)$-step functor and for $i = 1, 2, 3$, let (1) $B_i = gfp(\Phi_i)$ or (2) $B_i = lfp(\Phi_i)$.*

(1) If for $i = 1, 2, 3$, Φ_i is cocontinuous, then $B_2|all^{B_2}$ and $B_3|all^{B_3}$ are Σ_2-isomorphic.

(2) If for $i = 1, 2, 3$, Φ_i is continuous, then $B_2|all^{B_2}$ and $B_3|all^{B_3}$ are Σ_2-isomorphic.

Proof. (1) Let inc_1, inc_2 and inc_3 be the inclusion mappings from $B_1|all^{B_1}$ to A, from $B_2|all^{B_2}$ to $B_1|all^{B_1}$ and from $B_3|all^{B_3}$ to A, respectively.

```tikzcd
A \arrow[dr, phantom, "="] & B_3|all^{B_3} \arrow[l, "inc_3"'] \arrow[d, dashed, "h"] \\
B_1|all^{B_1} \arrow[u, "inc_1"] & B_2|all^{B_2} \arrow[l, "inc_2"]
```

interpretations of a non-primitive sort $s \in \Sigma$:
$s^{B_1} = s^A$
$s^{B_2} = s^A \cap all^{B_1}$
$s^{B_3} = s^A$

Suppose that the function $h : B_3|all^{B_3} \to B_2|all^{B_2}$ with $inc_1 \circ inc_2 \circ h = inc_3$ is bijective. Since $inc_1 \circ inc_2$ is Σ_2-monomorphic and inc_3 is Σ_2-homomorphic, Lemma 4.16(2) implies that h is a Σ_2-isomorphism. For the bijectivity of h we must show $all^{B_2} = all^{B_3}$ or, more generally, for all $r : s \in R$,

$$r^{B_2} \cap all^{B_2} \;=\; r^{B_3} \cap all^{B_3}. \tag{8}$$

Let $i = 1, 2, 3$. Since Φ_i is cocontinuous, Theorem 8.3(2) implies $B_i = \sqcap_{j \in \mathbb{N}} \Phi_i^j(\top)$. We start with the left-to-right inclusion of (8) and show by induction on j that for all $j \in \mathbb{N}$,

$$a \in s^A \setminus r^{\Phi_3^j(\top)} \quad \text{implies} \quad a \in s^A \setminus r^{\Phi_2^k(\top)} \tag{9}$$

for some $k \in \mathbb{N}$. Let $a \in s^A \setminus r^{\Phi_3^j(\top)}$. $j = 0$ implies

$$s^A \setminus r^{\Phi_3^j(\top)} = s^A \setminus r^\top = s^A \setminus r^A = s^A \setminus s^A = \emptyset.$$

Hence $j > 0$ and thus by the definition of Φ_3,

$$a \in s^A \setminus \varphi_{r,AX_3}^{\Phi_3^{j-1}(\top)}.$$

Since $AX_3 = AX_1 \cup AX_2$, there are two cases: (i) $a \in s^A \setminus \varphi_{r,AX_1}^{\Phi_3^{j-1}(\top)}$ and (ii) $a \in s^A \setminus \varphi_{r,AX_2}^{\Phi_1^{j-1}(\top)}$.

(i) By induction hypothesis, for all relations $q : s'$ occurring in φ_{r,AX_1}, $b \in (s')^A \setminus q^{\Phi_3^{j-1}(\top)}$ implies $b \in (s')^A \setminus q^{\Phi_2^k(\top)}$ for some $k \in \mathbb{N}$. Hence by the monotonicity of Φ_3, $a \in s^A \setminus \varphi_{r,AX_1}^{\Phi_2^k(\top)}$ and thus $a \in s^A \setminus r^{\Phi_2^k(\top)}$ for some $k \in \mathbb{N}$ because by Theorem 9.4, $B_1 \models r \Rightarrow \varphi_{r,AX_1}$ implies $\top = B_1|all^{B_1} \models r \Rightarrow \varphi_{r,AX_1}$ and thus $\Phi_2^k(\top) \models r \Rightarrow \varphi_{r,AX_1}$.

(ii) By induction hypothesis, for all relations $q : s'$ occurring in φ_{r,AX_2}, $b \in (s')^A \setminus q^{\Phi_3^{j-1}(\top)}$ implies $b \in (s')^A \setminus q^{\Phi_2^k(\top)}$ for some $k \in \mathbb{N}$. Hence by the monotonicity of Φ_3, $a \in s^A \setminus \varphi_{r,AX_2}^{\Phi_2^k(\top)}$ for some $k \in \mathbb{N}$. By the definition of Φ_2, we conclude that $a \in s^A \setminus r^{\Phi_2^{k+1}(\top)}$. This finishes the proof of (9).

We proceed with the right-to-left inclusion of (8) and show by induction on j that for all $j \in \mathbb{N}$,

$$a \in s^A \setminus r^{\Phi_2^j(\top)} \quad \text{implies} \quad a \in s^A \setminus r^{\Phi_3^k(\top)} \tag{10}$$

for some $k \in \mathbb{N}$. Suppose that

$$a \in s^A \setminus r^{B_1|all^{B_1}} \quad \text{implies} \quad a \in s^A \setminus r^{\Phi_3^k(\top)} \tag{11}$$

for some $k \in \mathbb{N}$. Let $a \in s^A \setminus r^{\Phi_2^j(\top)}$. If $j = 0$ then $a \in s^A \setminus r^\top = s^A \setminus r^{B_1|all^{B_1}}$. Hence by (11), $a \in s^A \setminus r^{\Phi_3^k(\top)}$ for some $k \in \mathbb{N}$. If $j > 0$, then by the definition of Φ_2,

$$a \in s^A \setminus \varphi_{r,AX_2}^{\Phi_2^j(\top)}.$$

By induction hypothesis, for all relations $q : s'$ occurring in φ_{r,AX_2}, $b \in (s')^A \setminus q^{\Phi_2^{j-1}(\top)}$ implies $b \in (s')^A \setminus q^{\Phi_3^k(\top)}$ for some $k \in \mathbb{N}$. Hence by the monotonicity of Φ_2,

$$a \in s^A \setminus \varphi_{r,AX_2}^{\Phi_3^k(\top)} \subseteq \varphi_{r,AX_3}^{\Phi_3^k(\perp)}$$

for some $k \in \mathbb{N}$ because $AX_2 \subseteq AX_3$ and thus for all Σ_2-structures C, $\varphi_{r,AX_3}^C \subseteq \varphi_{r,AX_2}^C$. By the definition of Φ_3, we conclude that $a \in s^A \setminus r^{\Phi_3^{k+1}(\top)}$. This finishes the proof of (10). It remains to show (11), i.e., that for all $j \in \mathbb{N}$,

$$a \in s^A \setminus r^{\Phi_1^j(\top)} \quad \text{implies} \quad a \in s^A \setminus r^{\Phi_3^k(\top)} \tag{12}$$

for some $k \in \mathbb{N}$. Let $a \in s^A \setminus r^{\Phi_1^j(\top)}$. If $j = 0$, then $a \in s^A \setminus r^\top = r^A = r^{\Phi_3^j(\top)}$. If $j > 0$, then by the definition of Φ_1,

$$a \in s^A \setminus \varphi_{r,AX_1}^{\Phi_1^{j-1}(\top)}.$$

By induction hypothesis, for all relations $q : s'$ occurring in φ_{r,AX_1}, $b \in (s')^A \setminus q^{\Phi_1^{j-1}(\top)}$ implies $b \in (s')^A \setminus q^{\Phi_3^k(\top)}$ for some $k \in \mathbb{N}$. Hence by the monotonicity of Φ_1,

$$a \in s^A \setminus \varphi_{r,AX_1}^{\Phi_3^k(\top)} \subseteq s^A \setminus \varphi_{r,AX_3}^{\Phi_3^k(\top)}$$

for some $k \in \mathbb{N}$ because $AX_1 \subseteq AX_3$ and thus for all Σ_2-structures C, $\varphi_{r,AX_3}^C \subseteq \varphi_{r,AX_1}^C$. By the definition of Φ_3, we conclude that $a \in s^A \setminus r^{\Phi_3^{k+1}(\top)}$. This finishes the proof of (12) and thus of (11).

(2) can be shown analogously. ❏

By Theorems 6.3, 6.4, 7.1, 7.2, 10.9 and 10.10 and Lemmas 12.4 and 12.5, the perfect model of a swinging type (Σ, AX) is always a (maybe trivial) quotient of a free Σ-structure or a (maybe trivial) substructure of a cofree Σ-structure:

Theorem 12.6 *Let $SP = (\Sigma, AX)$ be a swinging type with primitive sort set S_0 and sort-building predecessor SP_1.*

(1) *Let SP be an abstraction. Then there are a unique S_0-sorted set A and a μ-extension SP_2 of SP_1 such that SP is a ν-extension of SP_2 and*

$$Per(SP) \quad = \quad (Free(\Sigma, A)/{\equiv^{lfp(\Phi_1)}})/{\equiv^{gfp(\Phi_2)}}$$

where Φ_i, $i = 1, 2$, is the (A, σ_i)-step functor and σ_1, σ_2 are the relation transformers defined by $AX_2 \setminus AX_1$ and $AX \setminus AX_2$, respectively. If SP is a hidden abstraction, then $SP_2 = SP$.

(2) *Let SP be a restriction. Then there are a unique S_0-sorted set A and a ν-extension SP_2 of SP_1 such that SP is a μ-extension of SP_2 and*

$$Per(SP) \quad = \quad (Cofree(\Sigma, A)|all^{gfp(\Phi_1)})|all^{lfp(\Phi_2)}$$

where Φ_i, $i = 1, 2$, is the (A, σ_i)-step functor and σ_1, σ_2 are the relation transformers defined by $AX_2 \setminus AX_1$ and $AX \setminus AX_2$, respectively. If SP is a visible restriction, then $SP_2 = SP$. ❏

Theorem 12.6 allows us to characterize conservative model extensions in terms of congruences on free structures and invariants on cofree structures, respectively:

Lemma 12.7 *Let $SP' = (\Sigma', AX')$ be a swinging type with base type $SP = (\Sigma, AX)$, F_1 be the set of function symbols of $\Sigma' \setminus \Sigma$ and $A = prim(SP')$.*

(1) *Let SP' be a visible abstraction and $\sim = \equiv^{lfp(\Phi(SP'))}$,*

  (i) *for all $f : s \to s' \in F_1$, $t \in MGen_{\Sigma,s}$ and $a \in dom_t^A$ there are $\iota_u(b) \in Free(\Sigma', A)$ such that $\iota_{f \circ t}(a) \sim \iota_u(b)$,*

  (ii) *for all $\iota_t(a), \iota_u(b) \in Free(\Sigma', A)$, $\iota_t(a) \sim \iota_u(b)$ implies $t = u$ and $a = b$.*

  Then $Per(SP')$ is Σ-reachable and Σ-consistent.

(2) *Let SP' be a hidden restriction, $inv = all^{gfp(\Phi(SP'))}$,*

  (iii) *for all $f : s \to s' \in F_1$ and $a, b \in s^{inv}$, if $\pi_t(a) = \pi_t(b)$ for all $t \in MObs_{\Sigma,s}$, then $\pi_{t \circ f}(a) = \pi_{t \circ f}(b)$ for all $t \in MObs_{\Sigma,s'}$,*

  (iv) *for all sorts $s \in \Sigma$ and $a \in s^{Cofree(\Sigma',A)}$ there is $b \in s^inv$ such that for all $t \in MObs_{\Sigma,s}$, $\pi_t(a) = \pi_t(b)$.*

  Then $Per(SP')$ is Σ-observable and Σ-complete.

Proof. (1) By Theorem 12.6(1), $B =_{def} Per(SP') = Free(\Sigma', A)/{\sim}$. By Theorem 6.3, for all $t \in MGen_{\Sigma'}$, $t^{Free(\Sigma',A)} = \iota_t$. Hence by (ii), B is Σ-consistent. Since by Lemma 10.7(3), B is Σ-reachable, Theorem 11.2(1) implies that B is Σ-reachable if $reach_\Sigma^B$ is F_1-compatible. So let $f : s \to s' \in F_1$, $t \in MGen_{\Sigma',s}$ and $a \in dom_t^A$ such that

$$t^B(a) = nat(t^{Free(\Sigma',A)}(a)) = nat(\iota_t(a)) \in reach_\Sigma^B.$$

Then there are $u \in MGen_\Sigma$ and $b \in dom_u^A$ such that

$$nat(\iota_t(a)) = t^B(a) = u^B(b) = nat(u^{Free(\Sigma',A)}(b)) = nat(\iota_u(b))$$

and thus $\iota_t(a) \sim \iota_u(b)$. By (i), there are $v \in MGen_\Sigma$ and $c \in dom_v^A$ such that $\iota_{f \circ u}(b) \sim \iota_v(c)$. Hence

$$\begin{aligned} f^B(nat(\iota_t(a))) &= f^B(nat(\iota_u(b))) = nat(f^{Free(\Sigma',A)}(\iota_u(b))) = nat(f^{Free(\Sigma',A)}(u^{Free(\Sigma',A)}(b))) \\ &= nat((f \circ u)^{Free(\Sigma',A)}(b)) = nat(\iota_{f \circ u}(b)) = nat(\iota_v(c)) \in reach_\Sigma^B. \end{aligned}$$

We conclude that $reach_\Sigma^B$ is F_1-compatible.

(2) By Theorem 12.6(2), $B =_{def} Per(SP') = Cofree(\Sigma', A)|inv$. By Theorem 6.4, for all $t \in MObs_{\Sigma'}$, $t^{Cofree(\Sigma',A)} = \pi_t$. Hence by (iv), B is Σ-complete. Since by Lemma 10.7(4), B is Σ-observable, Theorem 11.2(2) implies that B is Σ-observable if obs_Σ^B is F_1-compatible. So let $f : s \to s' \in F_1$ and $a, b \in s^B \subseteq s^{inv}$ such

that $(a,b) \in obs^B_\Sigma$, i.e., for all $t \in MObs_{\Sigma,s}$, $\pi_t(a) = \pi_t(b)$. Then by (iii), for all $t \in MObs_{\Sigma,s'}$, $\pi_{t\circ f}(a) = \pi_{t\circ f}(b)$. Hence

$$\pi_t(f^B(a)) = \pi_t(inc(f^B(a))) = \pi_t(f^{Cofree(\Sigma',A)}(inc(a))) = t^{Cofree(\Sigma',A)}(f^{Cofree(\Sigma',A)}(a))$$
$$= (t\circ f)^{Cofree(\Sigma',A)}(a) = \pi_{t\circ f}(a) = \pi_{t\circ f}(b) = (t\circ f)^{Cofree(\Sigma',A)}(b) = t^{Cofree(\Sigma',A)}(f^{Cofree(\Sigma',A)}(b))$$
$$= \pi_t(f^{Cofree(\Sigma',A)}(inc(b))) = \pi_t(inc(f^B(b))) = \pi_t(f^B(b)),$$

i.e., $(f^B(a), f^B(b)) \in obs^B_\Sigma$. We conclude that obs^B_Σ is F_1-compatible. ❏

Definition 12.8 *(covering, defining)* A set T of Σ-generators is $MGen_\Sigma$**-covering** if for all $t \in MGen_\Sigma$ some term of T is a superterm of t. A set T of Σ-observers is is $MObs_\Sigma$**-covering** if for all $t \in MObs_\Sigma$ some term of T is a subterm of t.

Given a set $F_1 = \{f_1,\dots,f_n\}$ of S_1-constructors such that $F_1 \cap F = \emptyset$, a set AX of Horn clauses

$$\{f_i \circ c_{ij} \equiv c'_{ij} \Leftarrow \varphi_{ij} \wedge \bigwedge_{k=1}^{n_{ij}} f_{ijk} \circ c_{ijk} \equiv c'_{ijk} \mid 1 \le i \le n, 1 \le j \le n_i\}$$

for F_1 is F_1**-defining** if $\{c_{ij} \mid 1 \le i \le n, 1 \le j \le n_i\}$ is an $MGen_\Sigma$-covering set and there is a well-founded ordering $>$ such that for all $1 \le i \le n$, $1 \le j \le n_i$ and $1 \le k \le n_{ij}$, $f_{ijk} \in F$ and c'_{ij}, c_{ijk} and c'_{ijk} are Σ-generators, $(c_{ij}, c'_{ij}) > (c_{ijk}, c'_{ijk})$ and neither the terms $c_{ij}, c'_{ij}, c_{ijk}, c'_{ijk}$ nor the formula φ_{ij} contain symbols of F.

Given a set $F_1 = \{f_1,\dots,f_n\}$ of S_1-destructors such that $F_1 \cap F = \emptyset$, a set AX of Horn clauses

$$\{d_{ij} \circ f_i \equiv c_{ij} \Rightarrow \varphi_{ij} \wedge \bigwedge_{k=1}^{n_{ij}} d_{ijk} \circ f_{ijk} \equiv c_{ijk} \mid 1 \le i \le n, 1 \le j \le n_i\}$$

for F_1 is a F_1**-defining** if $\{d_{ij} \mid 1 \le i \le n, 1 \le j \le n_i\}$ is an $MObs_\Sigma$-covering set and there is a well-founded ordering $>$ such that for all $1 \le i \le n$, $1 \le j \le n_i$ and $1 \le i \le n_{ij}$, $f_{ijk} \in F$, c_{ij} and c_{ijk} are Σ-generators and d_{ijk} is a Σ-observer, $(d_{ij}, c_{ij}) > (d_{ijk}, c_{ijk})$ and neither the terms $d_{ij}, c_{ij}, d_{ijk}, c_{ijk}$ nor the formula φ_{ij} contain symbols of F. ❏

Theorem 12.9 *****

(1) $F_1 = \{f_1,\dots,f_n\}$ S_1-constructors. AX F_1-defining implies Lemma 12.7(i/ii).

(1) $F_1 = \{f_1,\dots,f_n\}$ S_1-destructors. AX F_1-defining implies Lemma 12.7(iii/iv).

13 Deductive semantics

Given a swinging type $SP = (\Sigma, AX)$, Theorem 8.14 implies that the least/greatest SP-model over $B = Free/Cofree(\Sigma, A)$ agrees with the least/greatest fixpoint of the (B,σ)-step functor where σ is the relation transformer defined by AX. We show that these models can also be characterized in terms of a sequent calculus based on AX. It is a variant of Gentzen's system LK [26]. We formulate it as a system of rules for inferring implications $\varphi \Rightarrow \psi$ and admit applications of Boolean laws in order to turn Σ-formulas into matching rule redices.

The calculus contains rules with infinitely many premises ($\wedge$- and $\forall$-introduction). Hence we must employ ordinal numbers and transfinite induction for defining the length of a proof via the calculus (see section 7). Ordinal numbers for measuring proofs rules have been used in, e.g., [106], §20, and [98], Section 1.3.

Definition 13.1 Let the assumptions of Def. 8.1 hold true, $\Sigma_1 = (S_0, S, F', R)$ and A be a Σ_1-structure. The **sequent calculus for** (A, SP') is given by the following rules for deriving (implications between) Σ'-formulas. Let I be a nonempty set.

base rule $\dfrac{}{\varphi \implies \varphi}$ for all $\varphi \in Form_{\Sigma'(A)}$ (see Def. **??**)

axiom rule $\dfrac{}{\varphi \implies \psi}$ for all $\varphi \Rightarrow \psi \in AX'(A)$ (see Def. **??**)

$\wedge$-introduction $\dfrac{\varphi \implies \psi}{\varphi \wedge \vartheta \implies \psi}$ $\dfrac{\varphi \implies \psi \vee \vartheta, \quad \varphi' \implies \psi' \vee \vartheta'}{\varphi \wedge \varphi' \implies (\psi \wedge \psi') \vee \vartheta \vee \vartheta'}$

$\vee$-introduction $\dfrac{\varphi \implies \psi}{\varphi \implies \psi \vee \vartheta}$ $\dfrac{\varphi \wedge \psi \implies \vartheta, \quad \varphi' \wedge \psi' \implies \vartheta'}{(\varphi \vee \varphi') \wedge \psi \wedge \psi' \implies \vartheta \vee \vartheta'}$

$\neg$-introduction $\dfrac{\varphi \implies \psi \vee \vartheta}{\varphi \wedge \neg\psi \implies \vartheta}$ $\dfrac{\varphi \wedge \psi \implies \vartheta}{\varphi \implies \neg\psi \vee \vartheta}$

$\Rightarrow$-introduction $\dfrac{\varphi \wedge \vartheta \implies \vartheta' \vee \psi}{\varphi \implies (\vartheta \Rightarrow \vartheta') \vee \psi}$ $\dfrac{\varphi \implies \psi \vee \vartheta, \quad \varphi' \wedge \psi' \implies \vartheta'}{(\psi \Rightarrow \varphi') \wedge \varphi \wedge \psi' \implies \vartheta \vee \vartheta'}$

$\forall$-introduction $\dfrac{\varphi \wedge \psi \odot_k a \implies \vartheta}{\varphi \wedge \forall k\psi \implies \vartheta}$ for all $a \in s_k^A$ and $k \in I$ (see Def. 3.15)

$\dfrac{\{\varphi \implies \psi \odot_k a \vee \vartheta \mid a \in s_k^A\}}{\varphi \implies \forall k\psi \vee \vartheta}$ for all $k \in I$

$\exists$-introduction $\dfrac{\varphi \implies \psi \odot_k a \vee \vartheta}{\varphi \implies \exists k\psi \vee \vartheta}$ for all $a \in s_k^A$ and $k \in I$

$\dfrac{\{\varphi \wedge \psi \odot_k a \implies \vartheta \mid a \in s_k^A\}}{\varphi \wedge \exists k\psi \implies \vartheta}$ for all $k \in I$

*** $\dfrac{\varphi}{t \odot \varphi} \Downarrow$ for all $t : s \to s' \in T_{\Sigma'}$ and $\varphi : s \in Form_\Sigma$ such that t^A is surjective

*** $\dfrac{t \odot \varphi}{\varphi} \Downarrow$ for all $t : s \to s' \in T_{\Sigma'}$ and $\varphi : s \in Form_\Sigma$ such that t^A is injective

*** $\dfrac{\{t \circ u \equiv t \circ v \mid t \in MObs_{\Sigma,s}\}}{u \equiv v} \Downarrow$ for all $s \in S$ and $u, v : dom \to s \in T_\Sigma$

Let α be an ordinal number. The set $\vdash^\alpha_{A,SP'}$ of Σ'-formulas **derivable** with the sequent calculus for (A, SP') is defined inductively as follows:

- Let $\{\varphi_i\}_{i\in I}$ be the premises and ψ be the conclusion of (an instance of) a rule of the sequent calculus for (A, SP') modulo Boolean equivalences including $\varphi \times True \Leftrightarrow \varphi$ and $\varphi + False \Leftrightarrow \varphi$. If for all $i \in I$ there is $\alpha_i < \alpha$ such that $\vdash^{\alpha_i}_{A,SP'} \varphi_i$, then $\vdash^\alpha_{A,SP'} \psi$.

The **length of a derivation** of $\varphi \in Form_{\Sigma'}$ via the sequent calculus for (A, SP') is the least ordinal α such that $\vdash^\alpha_{A,SP'} \varphi$. We write $\vdash_{A,SP'} \varphi$ if there is an ordinal α such that $\vdash^\alpha_{A,SP'} \varphi$. ❑

Definition 13.2 *(deductive model μ- and ν-extensions)* Let the assumptions of Def. 8.1 hold true, $\Sigma_1 = (S_0, S, F', R)$ and A be a Σ_1-structure. The **deductive** (A, SP')**-model**, $Ded(A, SP')$, is defined as follows.

- $Ded(A, SP')|_{\Sigma' \setminus R_1} = A|_{\Sigma' \setminus R_1}$,
- If SP' is a μ-extension of SP, then for all $r : s \in R_1$ and $t : 1 \to s \in T_{\Sigma'}$,

$$t^A \in r^{Ded(A,SP')} \quad \text{iff} \quad \vdash_{A,SP'} True \Rightarrow r(t).$$

- If SP' is a ν-extension of SP, then for all $r : s \in R_1$ and $t : 1 \to s \in T_{\Sigma'}$,

$$t^A \in r^{Ded(A,SP')} \quad \text{iff} \quad \not\vdash_{A,SP'} r(t) \Rightarrow False. \qquad ❑$$

Lemma 13.3 *(correctness and completeness of $\vdash_{A,SP'}$ wrt A) Let the assumptions of Def. 8.1 hold true, $\Sigma_1 = (S_0, S, F', R)$ and A be a Σ_1-structure. Then for all Σ'-formulas $\varphi : 1$,* *(1) if SP' is a μ-extension of SP, then $Ded(A, SP') \models \varphi$ iff $\vdash_{A,SP'}$ True $\Rightarrow \varphi$,*

(2) if SP' is a ν-extension of SP, then $Ded(A, SP') \not\models \varphi$ iff $\vdash_{A,SP'} \varphi \Rightarrow$ False.

Proof. (1) The "if"-direction (correctness) is shown by induction on the length of derivations via the sequent calculus for (A, SP'). As usually, this follows from the correctness of each rule $\frac{\{\varphi_i:s_i\}_{i\in I}}{\psi:s}$ of the calculus in the sense that for all $B \in Mod(A, SP')$;

$$\forall\, i \in I : s_i^B \subseteq \varphi_i^B \quad \text{implies} \quad s^B \subseteq \psi^B.$$

The correctness of the base rule, the axiom rule and $\wedge$- and $\vee$-introduction and -elimination is trivial. For the instance rule, suppose that $\prod_{i\in I} s_i^A \subseteq \varphi^A$.

The "only-if"-direction (completeness) is shown by induction on the size of Σ'-formulas.

(2) ❏

Theorem 13.4 *Let the assumptions of Def. 8.1 hold true, $\Sigma_1 = (S_0, S, F', R)$ and A be a Σ_1-structure.*

(1) If SP' is a μ-extension of SP, then $Ded(A, SP') = lfp(\Phi_{A,\sigma})$.

(2) If SP' is a ν-extension of SP, then $Ded(A, SP') = gfp(\Phi_{A,\sigma})$.

Proof. (1) Suppose that $B = Ded(A, SP')$ satisfies AX_1 and $r \in R_1$. Since $lfp(\Phi_{A,\sigma})$ satisfies AX_1, the cut calculus for SP' is correct w.r.t. $lfp(\Phi_{A,\sigma})$. Hence for all Σ'-atoms $r(t) : s$, $True_s \vdash_{A,SP'} r(t)$ implies $lfp(\Phi_{A,\sigma}) \models r(t)$ and thus r^B is a subset of $r^{lfp(\Phi_{A,\sigma})}$. Conversely, $r^{lfp(\Phi_{A,\sigma})}$ is a subset of r^B because $lfp(\Phi_{A,\sigma})$ is the least SP'-model over A. It remains to show that B satisfies AX_1.

Let $r(t) \Leftarrow \varphi \in AX_1$ and $B \models \varphi$. By Lemma 13.3(1), $True \vdash_{A,SP'} \varphi$ and thus $True \vdash_{A,SP'} r(t)$ by the definition of r^B. Hence $B \models r(t)$ and thus $B \models r(t) \Leftarrow \varphi$.

(2) Suppose that $B = Ded(A, SP')$ satisfies AX_1 and $r \in R_1$. Since $gfp(\Phi_{A,\sigma})$ satisfies AX_1, the cut calculus for SP' is correct w.r.t. $gfp(\Phi_{A,\sigma})$. Hence for all Σ'-atoms $r(t) : s$, $r(t) \vdash_{A,SP'} False_s$ implies $gfp(\Phi_{A,\sigma}) \models \neg r(t)$ and thus $r^{gfp(\Phi_{A,\sigma})}$ is a subset of r^B. Conversely, r^B is a subset of $r^{gfp(\Phi_{A,\sigma})}$ because $gfp(\Phi_{A,\sigma})$ is the greatest SP'-model over A. It remains to show that B satisfies AX_1.

Let $r(t) \Rightarrow \varphi \in AX_1$ and $B \models r(t)$. By the definition of r^B, $r(t) \not\vdash_{A,SP'} False$. Hence $\varphi \not\vdash_{A,SP'} False$ and thus $B \models \varphi$ by Lemma 13.3(2). Hence $B \models r(t) \Rightarrow \varphi$. ❏

14 Constructor-based algebras

Let $SP = (\Sigma, AX)$ be a visible swinging type with sort set S, constructor set CO, base type $baseSP = (base\Sigma, baseAX)$ and extension (Σ', AX').

A functor $F : Set^S \to Set^S$ is defined as follows: for all $A \in Set^S$ and $s \in S$,

$$F(A)_s \;=\; \begin{cases} s^B & \text{if } s \in base\Sigma, \\ \coprod_{f:w\to s\in CO} A_w & \text{otherwise.} \end{cases}$$

By Theorem 21.3, F is continuous and thus by Theorem 21.2, $Alg(F)$ has an initial object $ini : F(Ini(F)) \to Ini(F)$. $Ini(F)$ can be represented as the algebra $T_{B\cup CO}$ of finite ground terms over $B \cup CO$.

Proof!

The free F-algebra over an S-sorted set X is given by the algebra $T_{B\cup CO}(X)$ of finite terms over $B \cup CO$ with variables in X. This complies with Theorem 21.8 because X just forms a set of additional constants, in other words: $T_{B\cup CO\cup X}$ coincides with $T_{B\cup CO}(X)$.

Σ-structures are F-algebras: $\alpha : F(A) \to A$ is obtained from a Σ-structure A by combining the interpretations in A of all constructors $f : w \to s \in CO$ into a single morphism α such that for all $A \in Set^S$ and $a \in A_w$,

$$\alpha(\iota_f(a)) =_{def} f^A(a)$$

where ι_f is the injection from A_w to $\coprod_{f:w\to s\in CO} A_w$. Conversely, decomposing an F-algebra α into an interpretation of CO defines a Σ-algebra: for all $a \in A_w$,

$$f^A(a) =_{def} \alpha(\iota_f(a)).$$

Example 14.1

NAT

`vissorts`	nat
`constructs`	$0 :\to nat$
	$suc : nat \to nat$
`defuncts`	$pred : nat \to 1 + nat$
	$1 : 1 \to nat$
	$\_ + \_, \_ - \_ : nat \times nat \to nat$
	$min, max : nat \times nat \to nat$
`preds`	$\_ \leq \_ : nat \times nat$
	$\_ \not\equiv \_ : nat \times nat$
`vars`	$x, y : nat$
`axioms`	$pred(0) \equiv \iota_1$
	$pred(suc(x)) \equiv \iota_2(x)$
	$1 \equiv suc(0)$
	$0 + x \equiv x$
	$suc(x) + y \equiv suc(x + y)$
	$0 - x \equiv 0$
	$suc(x) - y \equiv suc(x - y)$
	$min(0, x) \equiv 0$
	$min(suc(x), 0) \equiv 0$
	$min(suc(x), suc(y)) \equiv min(x, y)$
	$max(0, x) \equiv x$
	$max(suc(x), 0) \equiv suc(x)$
	$min(suc(x), suc(y)) \equiv max(x, y)$
	$0 \leq x$
	$suc(x) \leq suc(y) \quad \Leftarrow \quad x \leq y$
	$0 \not\equiv suc(x)$
	$suc(x) \not\equiv 0$
	$suc(x) \not\equiv suc(y) \quad \Leftarrow \quad x \not\equiv y$

In terms of Def. 5.1, SP is empty, $S' = \{nat\}$ and $F' = \{0, suc\}$. By Theorem 6.3,

$$nat^{Ini} \;=\; \{suc^n(0) \mid n \in \mathbb{N}\}.$$

SUPERTYPE! By ???, $F(A)_{nat} = 1 + A_{nat}$. The initial NAT-model is isomorphic to the initial F-algebra (see Def. 3.3). In particular, α_{nat} is the unique sum extension of $0^{Ini(F)}$ and $suc^{Ini(F)}$. ❑

Example 14.2 The following specification of finite lists is a parameterized ST that extends the parameter type TRIV(s) (see Example 5.4):

$\mathrm{LIST}[\mathrm{TRIV}(s)[\mathrm{BOOL}]]$ `where` $\mathrm{LIST} = \mathrm{NAT}$ `and`

`vissorts`	$list(s)$
`constructs`	$[] : 1 \to list(s)$
	$\_ : \_ : s \times list(s) \to list(s)$
	$\lambda y.\_(x, y) : ((s \times s) \to bool) \to (s \to bool)$
`recfuncts`	$head : list(s) \to 1 + s$
	$tail : list(s) \to 1 + list(s)$
	$length : list(s) \to nat$
`defuncts`	$[\_] : s \to list(s)$
	$take, drop : nat \times list(s) \to list(s)$
	$reverse : list(s) \to list(s)$
	$odds, evens : list(s) \to list(s)$
	$\_ {+\!\!+} \_, zip : list(s) \times list(s) \to list(s)$
	$length' : list(s) \to nat$
	$nth : list(s) \to 1 + s$
	$exists, forall : (s \to bool) \times list(s) \to bool$
	$filter : (s \to bool) \times list(s) \to list(s)$
	$remove : s \times list(s) \to list(s)$
	$\$: ((s \to bool) \times s) \to bool$
`preds`	$\_ \not\equiv \_ : list(s) \times list(s)$
`vars`	$x, y : s \quad L, L' : list(s) \quad n : nat \quad f : s \to bool \quad g : s \times s \to bool$
`axioms`	$head \circ [] \equiv \iota_1$
	$head \circ (:) \equiv \iota_2 \circ \pi_1$
	$tail \circ [] \equiv \iota_1$
	$tail \circ (:) \equiv \iota_2 \circ \pi_2$
	$length \circ [] \equiv 0$
	$length \circ (:) \equiv suc \circ \pi_2 \circ (id_s \times length)$
	$[x] \equiv x : []$
	$take(0, L) \equiv []$
	$take(suc(n), []) \equiv []$
	$take(suc(n), x : L) \equiv x : take(n, L)$
	$drop(0, L) \equiv L$
	$drop(suc(n), []) \equiv []$
	$drop(suc(n), x : L) \equiv drop(n, L)$
	$reverse([]) \equiv []$
	$reverse(x : L) \equiv reverse(L) {+\!\!+} [x]$
	$odds([]) \equiv []$
	$odds(x : L) \equiv x : evens(L)$
	$evens([]) \equiv []$
	$evens(x : L) \equiv odds(L)$
	$[] {+\!\!+} L \equiv L$
	$(x : L) {+\!\!+} L' \equiv x : (L {+\!\!+} L')$
	$zip([], L) \equiv L$
	$zip(L, []) \equiv L$
	$zip(x : L, y : L') \equiv x : y : zip(L, L')$
	$length'([]) \equiv 0$
	$length'(x : L) \equiv length'(L) + 1$
	$nth(n, []) \equiv \iota_1$

$$\begin{array}{l}
nth(0, x : L) \equiv \iota_2(x) \\
nth(suc(n), x : L) \equiv nth(n, L) \\
exists(f, []) \equiv false \\
exists(f, x : L) \equiv f(x) \text{ or } exists(f, L) \\
forall(f, []) \equiv true \\
forall(f, x : L) \equiv f(x) \text{ and } forall(f, L) \\
filter(f, []) \equiv [] \\
filter(f, x : L) \equiv x : filter(f, L) \;\Leftarrow\; f(x) \equiv true \\
filter(f, x : L) \equiv filter(f, L) \;\Leftarrow\; f(x) \equiv false \\
remove(x, L) \equiv filter(\lambda y.not(eq(x, y)), L) \\
\$(\lambda y.g(x, y), y) \equiv g(x, y) \\
[] \not\equiv x : L \\
x : L \not\equiv [] \\
x : L \not\equiv y : L' \quad\Leftarrow\quad x \not\equiv y \vee L \not\equiv L'
\end{array}$$

Note that a λ-abstraction is declared as a constructor mapping from the sorts of its free variables to a sort representing functions in its bounded variables. λ-abstractions as well as other higher-order functions are applied to arguments by suitable *apply* functions (denoted by \$), which are—usually implicitly—declared as defined functions (see Section 2.2).

In terms of Def. 5.1, SP=TRIV$(s)\cup$NAT, $S' \setminus S = \{list(s)\}$ and $F' \setminus F = \{[], \_ : \_\}$. Let B be a parameter model of TRIV(s). Hence $T_{B\cup CO,list(s)}$ is the set of finite lists with entries in s^B and

$$F(A)_{list(s)} = 1 + s^B \times A_{list(s)}.$$

The initial LIST(B)-model is isomorphic to the initial F-algebra $\alpha : F(Ini(F)) \to Ini(F)$ (see Def. 3.3). In particular, $\alpha_{list(s)}$ is the unique sum extension of $[]^{Ini(F)}$ and $(\_ : \_)^{Ini(F)}$. ❏

Definition 14.3 Let $SP = (\Sigma, AX)$ and

$$PSP = SP[PAR_1, \dots, PAR_k, PAR_{k+1}, \dots, PAR_n],\; PSP_1,\; \dots,\; PSP_k$$

be parameterized types. For all $1 \leq i \leq k$ let $P\Sigma_i$ be the set of signature symbols of PAR_i that do not belong to a constant subtype of PAR_i (see Def. 5.3). Let $\sigma_i : P\Sigma_i \to \Sigma(PSP_i)$ be a signature morphism. The parameterized type

$$SP_{\sigma_1,\dots,\sigma_k}[PSP_1, \dots, PSP_k][PAR_{k+1}, \dots, PAR_n] \;=_{def}\; (\Sigma', AX')[PAR_{k+1}, \dots, PAR_n]$$

with

$$\Sigma' \;=_{def}\; \bigcup_{i=1}^{k}(\Sigma(PSP_i) \cup \sigma_i(\Sigma)) \;\text{ und }\; AX' \;=_{def}\; \bigcup_{i=1}^{k}(AX(PSP_i) \cup \sigma_i(AX))$$

is called the **amalgamation** of PSP with $PSP_1 \dots, PSP_k$ along $\sigma_1 \dots, \sigma_k$ where σ_i, $1 \leq i \leq k$, is extended to Σ as follows:

- $\sigma_i(s) = s$ for all unstructured sorts $s \in \Sigma$,
- $\sigma_i(s(s_1, \dots, s_n)) = s(\sigma_i(s_1), \dots, \sigma_i(s_n))$ for all structured sorts $s \in \Sigma$,
- $\sigma_i(f : w \to s) = f : \sigma_i(w) \to \sigma_i(s)$ for all functions $f \in \Sigma$,
- $\sigma_i(r : w) = r : \sigma_i(w)$ for alle relations $r \in \Sigma$. ❏

Example 14.4 The following specification of finite binary trees is a further parameterized swinging type that extends the parameter type TRIV(s):

BINTREE[TRIV(s)[BOOL]] `where` BINTREE = NAT `and` LIST$_{bool/s}$[BOOL] ??? `and`

`vissorts`	$bintree(s)$
`constructs`	$mt : 1 \to bintree(s)$
	$\_\#\_\#\_ : bintree(s) \times s \times bintree \to bintree(s)$
	$\lambda y.\_(x,y) : ((s \times s) \to bool) \to (s \to bool)$
`defuncts`	$size : bintree(s) \to nat$
	$< \_ >: s \to bintree(s)$
	$mirror : bintree(s) \to bintree(s)$
	$subtree : bintree(s) \times list(bool) \to 1 + bintree(s)$
	$exists, forall : (s \to bool) \times bintree(s) \to bool$
	$\_ \in \_ : s \times bintree(s) \to bool$
`preds`	$\_ \not\equiv \_ : bintree(s) \times bintree(s)$
`vars`	$x, y : s \quad T, T' : bintree(s) \quad b : bool \quad bL : list(bool) \quad f : s \to bool \quad g : s \times s \to bool$
`axioms`	$size(mt) \equiv 0$
	$size(T\#x\#T') \equiv size(T) + size(T') + 1$
	$< x >\equiv mt\#x\#mt$
	$mirror(mt)) \equiv mt$
	$mirror(T\#x\#T') \equiv mirror(T')\#x\#mirror(T)$
	$subtree(T, []) \equiv \iota_2(T)$
	$subtree(mt, bL) \equiv \iota_1$
	$subtree(T\#x\#T', true : bL) \equiv subtree(T, bL)$
	$subtree(T\#x\#T', false : bL) \equiv subtree(T', bL)$
	$exists(f, mt) \equiv false$
	$exists(f, T\#x\#T') \equiv f(x) \text{ or } exists(f, T) \text{ or } exists(f, T')$
	$forall(f, mt) \equiv true$
	$forall(f, T\#x\#T') \equiv f(x) \text{ and } forall(f, T) \text{ and } forall(f, T')$
	$x \in T \equiv exists(\lambda y.eq(x, y), T)$
	$\$(\lambda y.g(x, y), y) \equiv g(x, y)$
	$mt \not\equiv T\#x\#T'$
	$T\#x\#T' \not\equiv mt$
	$T\#x\#T' \not\equiv T_1\#y\#T_2 \quad \Leftarrow \quad x \not\equiv y \vee T \not\equiv T_1 \vee T' \not\equiv T_2$

In terms of Def. 5.1, SP=TRIV(s)∪NAT∪LIST(BOOL), $S' \setminus S = \{bintree(s)\}$ and $F' \setminus F\{mt, \_\#\_\#\_\}$. Let B be a parameter model of TRIV(s). Hence $T_{B \cup CO, bintree(s)}$ is the set of finite binary trees with entries in s^B and

$$F(A)_{bintree(s)} = 1 + A_{bintree(s)} \times s^B \times A_{bintree(s)}.$$

The initial BINTREE(B)-model is isomorphic to the initial F-algebra (see Def. 3.3). In particular, $\alpha_{bintree(s)}$ is the unique sum extension of $mt^{Ini(F)}$ and $(\_\#\_\#\_)^{Ini(F)}$. ❏

Example 14.5 The following specification of finite trees is a parameterized swinging type with two new (visible) sorts:

TREE[TRIV(s)[BOOL]] `where` TREE = NAT `and` LIST$_{nat/s}$[NAT] ??? `and`

`vissorts`	$tree(s) \quad trees(s)$
`constructs`	$\_\&\_ : s \times trees(s) \to tree(s)$
	$[] : 1 \to trees(s)$
	$\_ : \_ : tree(s) \times trees(s) \to trees(s)$
	$\lambda y.\_(x,y) : ((s \times s) \to bool) \to (s \to bool)$

`destructs`	$rs : tree(s) \to s \times trees(s)$	
	$ht : trees(s) \to 1 + s \times trees(s)$	
`defuncts`	$size : tree(s) \to nat$	
	$sizeL : trees(s) \to nat$	
	$<\_>: s \to tree(s)$	
	$subtree : tree(s) \times list(bool) \to 1 + tree(s)$	
	$subtreeL : trees(s) \times list(bool) \to 1 + tree(s)$	
	$exists, forall : (s \to bool) \times tree(s) \to bool$	
	$\_ \in \_ : s \times tree(s) \to bool$	
	$\$: ((s \to bool) \times s) \to bool$	
`preds`	$\_ \not\equiv \_ : tree(s) \times tree(s)$	
	$\_ \not\equiv \_ : trees(s) \times trees(s)$	
`vars`	$x, y : s \quad T : tree(s) \quad TL : trees(s) \quad n : nat \quad nL : list(nat)$	
	$f : s \to bool \quad g : s \times s \to bool$	
`axioms`	$rs(x\&TL) \equiv (x, TL)$	(1)
	$ht([]) \equiv \iota_1$	(2)
	$ht(T : TL) \equiv \iota_2(T, TL)$	(3)
	$size(x\&TL) \equiv sizeL(TL) + 1$	
	$sizeL([]) \equiv 0$	
	$sizeL(T : TL) \equiv size(T) + sizeL(TL)$	
	$< x > \equiv x\&[]$	
	$subtree(T, []) \equiv \iota_2(T)$	
	$subtree(x\&TL, n : nL) \equiv subtreeL(TL, n : nL)$	
	$subtreeL([], nl) \equiv \iota_1$	
	$subtreeL(T : TL, []) \equiv \iota_1$	
	$subtreeL(T : TL, 0 : nL) \equiv subtree(T, nL)$	
	$subtreeL(T : TL, suc(n) : nL) \equiv subtreeL(TL, n : nL)$	
	$exists(f, x\&TL) \equiv f(x)\ or\ existsL(f, TL)$	
	$existsL(f, []) \equiv false$	
	$existsL(f, T : TL) \equiv exists(f, T)\ or\ existsL(f, TL)$	
	$forall(f, x\&TL) \equiv f(x)\ and\ forallL(f, TL)$	
	$forallL(f, []) \equiv true$	
	$forallL(f, T : TL) \equiv forall(f, T)\ and\ forallL(f, TL)$	
	$x \in T \equiv exists(\lambda y.eq(x, y), T)$	
	$\$(\lambda y.g(x, y), y) \equiv g(x, y)$	
	$x\&TL \not\equiv y\&TL' \quad \Leftarrow \quad x \not\equiv y \vee TL \not\equiv TL'$	
	$[] \not\equiv T : TL$	
	$T : TL \not\equiv []$	
	$T : TL \not\equiv T' : TL' \quad \Leftarrow \quad T \not\equiv T' \vee TL \not\equiv TL'$	

Functional programmers, don't cry because of the introduction of a list version for each function on trees! Of course, an implementation would avoid the list versions by using the well-known map function that applies a function on s to each element of an s-list. However, this does not help when properties of a tree function f shall be *proved*. Then one needs has to find and prove corresponding properties of $map \circ f$. To this end, an explicit definition of $map \circ f$ will be needed anyway.

In terms of Assumption **??**, SP_i=TRIV(s)∪NAT∪LIST(NAT), $extS = \{tree(s), trees(s)\}$ and $CO = \{\_\&\_, [], \_ : \_\}$. Let B be a parameter model of TRIV(s). Hence $T_{B \cup CO, tree(s)}$ is the set of nonempty finite trees with entries in s^B and finite node degree, $T_{B\cup CO,trees(s)}$ is the set of finite forests with entries in s^B and

$$F(A)_{tree(s)} = s^B \times A_{trees(s)},$$
$$F(A)_{trees(s)} = 1 + A_{tree(s)} \times A_{trees(s)}.$$

The initial TREE(B)-model is isomorphic to the initial F-algebra (see Def. 3.3). In particular, $\alpha_{tree(s)} = (\_\&\_)^{Ini(F)}$, $\alpha_{trees(s)}$ is the unique sum extension of $[]^{Ini(F)}$ and $(\_:\_)^{Ini(F)}$, $\alpha^{-1}_{tree(s)} = rs^{Ini(F)}$ and $\alpha^{-1}_{trees(s)} = ht^{Ini(F)}$. Hence rs ("root and successors") and ht ("head and tail") are called destructors. In terms of Def. 5.1, they are defined functions (by axioms (1)-(3)). ❑

15 Constructor-based coalgebras

Let $SP = (\Sigma, AX)$ be a hidden swinging type with sort set S, constructor set CO, base type $baseSP = (base\Sigma, baseAX)$ and extension (Σ', AX').

By Thm. 21.3, F is also cocontinuous. Hence by Thm. 21.2, $coAlg(F)$ has a final object $fin : Fin(F) \to F(Fin(F))$. $Fin(F)$ can be represented as the algebra $T^{\infty}_{B\cup CO}$ of finite or infinite ground terms over $B \cup CO$. The elements of $T^{\infty}_{B\cup CO}$ are usually represented as the partial functions $t : \mathbb{N}^* \to B\cup CO$ whose domain $dom(t)$ satisfies the following conditions: for all $w \in \mathbb{N}^*$ and $i \in \mathbb{N}$,

$$wi \in dom(t) \quad \Rightarrow \quad w \in dom(t),$$
$$w(i+1) \in dom(t) \quad \Rightarrow \quad wi \in dom(t).$$

$f : w \to s \in CO$ is defined in $Fin(F)$ as in $Ini(F)$ just by placing the symbol f on top of the argument terms: for all $t \in Fin(F)_w$, $f^{Fin(F)}(t) =_{def} f(t)$.

Proof!

The cofree F-coalgebra over an S-sorted set X is given by the product $T^{\infty}_{B\cup CO} \times X$. Hence each element can be represented as a finite or infinite ground term over $B \cup CO$ whose root is *colored* by an element of X.

$Ini(F) = T_{B\cup CO}$ is the least fixpoint, $Fin(F) = T^{\infty}_{B\cup CO}$ is the greatest fixpoint of the same functor F.[15] But F-algebras and F-coalgebras are usually different from each other: given an S-sorted set A, an F-algebra α maps $F(A)$ to A, while an F-coalgebra β maps A to $F(A)$. Since $F(A)$ is a sum of sets (see §4.2), α is a sum of maps that can be decomposed into several functions, one for each constructor of CO. Conversely, β is a single function d^A that maps A to a sum of sets. Since the initial α and the final β are isomorphisms, $Ini(F)$ is also an F-*coalgebra* and $Fin(F)$ is also an F-*algebra*, i.e., $d^{Fin(F)}$ is the inverse of the sum of constructor interpretations mapping $F(Fin(F))$ to $Fin(F)$. Hence $d^{Fin(F)}$ is the interpretation of the S-sorted *destructor*

$$d_s : s \;\to\; \begin{cases} 1 & \text{if } s \in baseS, \\ \coprod_{f:w\to s\in CO} w & \text{otherwise} \end{cases}$$

such that for all $f : w \to s \in CO$ and $t \in Fin(F)_w$, $d_s^{Fin(F)}(f(t)) =_{def} \iota_f(t)$ where ι_f is the injection from $Fin(F)_w$ to $\coprod_{f:w\to s\in CO} Fin(F)_w$.

Hence for final F-coalgebras, constructors are as essential as they are for initial F-algebras. Syntactically, their different interpretation is indicated by the different mode of the sorts of *extS*: if *extS* consists of visible sorts, then they are interpreted as carriers of the initial F-algebra; if *extS* consists of hidden sorts, they are interpreted as carriers of the final F-coalgebra.

[15] By [7], Thm. 3.2, the fixpoints of other both continuous and cocontinuous functors F on *Set* (and thus on Set^S) are related to each other in the same way: the final F-coalgebra is the *Cauchy completion* of the initial F-algebra.

For all $s \in extS$, behavioral s-equality may be specified either in terms of CO_s or in terms of d_s. In fact, the behavior axioms B1/B2 and B5/B6 (see Def. 5.1(2)) are equivalent in $Fin(F)$ (but of course not in all F-coalgebras!):

$$x \sim_s y \quad \Rightarrow \quad d_s(x) \sim d_s(y)$$

holds true in $Fin(F)$ iff for all $f, g \in CO_s$,

$$f(x) \sim_s g(y) \quad \Rightarrow \quad d_s(f(x)) \sim d_s(g(y))$$

holds true in $Fin(F)$ iff for all $f, g \in CO_s$,

$$f(x) \sim_s g(y) \quad \Rightarrow \quad \iota_f(x) \sim \iota_g(y)$$

holds true in $Fin(F)$ iff for all $f, g \in CO_s$ with $f \neq g$,

$$\begin{aligned} f(x) \sim_s f(y) \quad &\Rightarrow \quad \iota_f(x) \sim \iota_f(y), \\ f(x) \sim_s g(y) \quad &\Rightarrow \quad \iota_f(x) \sim \iota_g(y) \end{aligned}$$

hold true in $Fin(F)$ iff for all $f, g \in CO_s$ with $f \neq g$,

$$\begin{aligned} f(x) \sim_s f(y) \quad &\Rightarrow \quad x \sim y, \\ f(x) \sim_s g(y) \quad &\Rightarrow \quad \mathit{False} \end{aligned}$$

holds true in $Fin(F)$. In all subsequent examples, we specify behavioral equalities in terms of B5/B6.

Example 15.1

CONAT

`hidsorts`	$cnat$	
`destructs`	$pred : cnat \to 1 + cnat$	
`cofuncts`	$0, 1, \infty : 1 \to cnat$	
	$suc : cnat \to cnat$	
	$\_ + \_, \_ - \_ : cnat \times cnat \to cnat$	
	$min, max : cnat \times cnat \to cnat$	
`preds`	$\_ \not\sim \_ : cnat \times cnat$	
`copreds`	$\_ \leq \_ : cnat \times cnat$	
	$\_ \sim \_ : cnat \times cnat$	
`vars`	$x, y, x', y' : cnat$	
`axioms`	$pred(0) \equiv \iota_1$	(1)
	$pred(suc(x)) \equiv \iota_2(x)$	(2)
	$pred(1) \equiv \iota_2(0)$	
	$pred(\infty) \equiv \iota_2(\infty)$	
	$pred(x + y) \equiv pred(y) \quad \Leftarrow \quad pred(x) \equiv \iota_1$	
	$pred(x + y) \equiv \iota_2(x' + y) \quad \Leftarrow \quad pred(x) \equiv \iota_2(x')$	
	$pred(x - y) \equiv \iota_1 \quad \Leftarrow \quad pred(x) \equiv \iota_1$	
	$pred(x - y) \equiv \iota_2(x' - y) \quad \Leftarrow \quad pred(x) \equiv \iota_2(x')$	
	$x \leq y \quad \Rightarrow \quad pred(x) \equiv \iota_1 \vee (pred(x) \equiv \iota_2(x') \wedge pred(y) \equiv \iota_2(y') \wedge x' \leq y')$	
	$0 \sim suc(x) \quad \Rightarrow \quad \mathit{False}$	(3)
	$suc(x) \sim 0 \quad \Rightarrow \quad \mathit{False}$	(4)
	$suc(x) \sim suc(y) \quad \Rightarrow \quad x \sim y$	(5)
	$0 \not\sim suc(x)$	(6)
	$suc(x) \not\sim 0$	(7)
	$suc(x) \not\sim suc(y) \quad \Leftarrow \quad x \not\sim y$	(8)

In terms of Assumption **??**, SP_i is empty, $extS = \{cnat\}$, $CO = \{0, suc\}$ and $d_{cnat} = pred$. Hence

$$T_{CO}^{\infty} \;=\; \{suc^n(0) \mid n \in \mathbb{N} \cup \{\infty\}\}$$

and $F(A)_{cnat} = 1 + A_{cnat}$.[16] Note that equations (1)-(8) present the general definition of $d_s^{Fin(F)}$ (see above) and the behavior axioms (see 5.1(2)) for $s = cnat$, respectively, and may thus be dropped. ❑

Example 15.2 The following specification of finite or infinite lists is a parameterized ST that extends the parameter type TRIV(s) (see Example 5.4):

COLIST[TRIV(s)] = NAT `and` LIST[TRIV(s)] `and` CONAT `then`

`hidsorts`	$clist(s)$	
`constructs`	$[] : 1 \to clist(s)$	
	$\_ : \_ : s \times clist(s) \to clist(s)$	
	$\lambda y.\_(x, y) : ((s \times s) \to bool) \to (s \to bool)$	
`destructs`	$ht : clist(s) \to 1 + s \times clist(s)$	
	$length : clist(s) \to 1 + nat$	
`corecfuncts`	$length' : clist(s) \to cnat$	
	$min : (s \to bool) \times clist(s) \to 1 + nat$	
`cofuncts`	$[\_] : s \to clist(s)$	
	$take : nat \times clist(s) \to list(s)$	
	$drop : nat \times clist(s) \to clist(s)$	
	$odds, evens : clist(s) \to clist(s)$	
	$\_ {+\!\!+} \_, zip : clist(s) \times clist(s) \to clist(s)$	
	$nth : nat \times clist(s) \to 1 + s$	
	$filter : (s \to bool) \times clist(s) \to clist(s)$	
`defuncts`	$\_ \in \_ : s \times clist(s) \to bool$	
	$remove : s \times clist(s) \to clist(s)$	
	$\$: ((s \to bool) \times s) \to bool$	
`preds`	$exists : (s \to bool) \times clist(s)$	
	$finite : clist(s)$	
`copreds`	$forall : (s \to bool) \times clist(s)$	
	$infinite : clist(s)$	
	$fair : (s \to bool) \times clist(s)$	
	$\_ \sim \_ : clist(s) \times clist(s)$	
`vars`	$x, y : s \quad L, L', L'' : clist(s) \quad m, n : nat \quad f : s \to bool \quad g : s \times s \to bool$	
`axioms`	$ht([]) \equiv \iota_1$	(1)
	$ht(x : L) \equiv \iota_2(x, L)$	(2)
	$take(0, L) \equiv []$	
	$take(suc(n), L) \equiv [] \quad \Leftarrow \quad ht(L) \equiv \iota_1$	
	$take(suc(n), L) \equiv x : take(n, L') \quad \Leftarrow \quad ht(L) \equiv \iota_2(x, L')$	
	$ht(drop(0, L)) \equiv ht(L)$	
	$ht(drop(n+1, L) \equiv \iota_1 \quad \Leftarrow \quad ht(L) \equiv \iota_1$	
	$ht(drop(n+1, L) \equiv ht(drop(n, L')) \quad \Leftarrow \quad ht(L) \equiv \iota_2(x, L')$	
	$ht(odds(L)) \equiv \iota_2(x, evens(L')) \quad \Leftarrow \quad ht(L) \equiv \iota_2(x, L')$	
	$ht(evens(L)) \equiv odds(L') \quad \Leftarrow \quad ht(L) \equiv \iota_(x, L')$	
	$ht(L {+\!\!+} L') \equiv \iota_1 \quad \Leftarrow \quad ht(L) \equiv \iota_1 \wedge ht(L') \equiv \iota_1$	
	$ht(L {+\!\!+} L') \equiv \iota_2(x, L'') \quad \Leftarrow \quad ht(L) \equiv \iota_1 \wedge ht(L') \equiv \iota_2(x, L'')$	

[16] $suc^{\infty}(0)$ denotes the infinite term t with $dom(t) = 0^*$ and $t(w) = suc$ for all $w \in dom(t)$.

$$
\begin{array}{ll}
ht(L +\!\!+ L') \equiv \iota_2(x, L'' +\!\!+ L') \quad \Leftarrow \quad ht(L) \equiv \iota_2(x, L'') & \\
ht(zip(L, L')) \equiv \iota_1 \quad \Leftarrow \quad ht(L) \equiv \iota_1 \wedge ht(L') \equiv \iota_1 & \\
ht(zip(L, L')) \equiv \iota_2(x, L'') \quad \Leftarrow \quad ht(L) \equiv \iota_1 \wedge ht(L') \equiv \iota_2(x, L'') & \\
ht(zip(L, L')) \equiv \iota_2(x, zip(L', L'')) \quad \Leftarrow \quad ht(L) \equiv \iota_2(x, L'') & \\
nth(n, L) \equiv \iota_1 \quad \Leftarrow \quad ht(L) \equiv \iota_1 & \\
nth(0, L) \equiv \iota_2(x) \quad \Leftarrow \quad ht(L) \equiv \iota_2(x, L') & \\
nth(n+1, L) \equiv nth(n, L') \quad \Leftarrow \quad ht(L) \equiv \iota_2(x, L') & \\
length(L) \equiv \iota_1 \quad \Leftarrow / \Rightarrow \quad \exists x, L' : (ht(L) \equiv \iota_2(x, L') \wedge length(L') \equiv \iota_1) & \\
length(L) \equiv \iota_2(0) \quad \Leftarrow / \Rightarrow \quad ht(L) \equiv \iota_1 & \\
length(L) \equiv \iota_2(suc(n)) \quad \Leftarrow / \Rightarrow \quad \exists x, L' : (ht(L) \equiv \iota_2(x, L') \wedge length(L') \equiv \iota_2(n)) & \\
pred \circ length' \equiv (id + (length' \circ \pi_2)) \circ ht & \\
min(f, L) \equiv \iota_1 \quad \Rightarrow \quad f(nth(n, L)) \equiv false & \\
min(f, L) \equiv \iota_2(n) \quad \Rightarrow \quad f(nth(n, L)) \equiv true \wedge (n \leq m \vee f(nth(m, L)) \equiv false & \\
ht(filter(f, L)) \equiv \iota_1 \quad \Leftarrow \quad min(f, L) \equiv \iota_1 & \\
ht(filter(f, L)) \equiv \iota_2(x, filter(f, L')) & \\
\quad \Leftarrow \quad min(f, L) \equiv \iota_2(n) \wedge nth(n, L) \equiv \iota_2(x) \wedge drop(n+1, L) \equiv L' & \\
x \in L \equiv exists(\lambda y.eq(x, y), L) & \\
remove(x, L) \equiv filter(\lambda y.not(eq(x, y)), L) & \\
\$(\lambda y.g(x, y), y) \equiv g(x, y) & \\
exists(f, L) \quad \Leftarrow \quad ht(L) \equiv \iota_2(x, L') \wedge (f(x) \equiv true \vee exists(f, L')) & \\
finite(L) \quad \Leftarrow \quad ht(L) \equiv \iota_1 \vee \exists x, L' : (ht(L) \equiv \iota_2(x, L') \wedge finite(L')) & \\
forall(f, L) & \\
\quad \Rightarrow \quad ht(L) \equiv \iota_1 \vee \exists x, L' : (ht(L) \equiv \iota_2(x, L') \wedge f(x) \equiv true \wedge forall(f, L')) & \\
infinite(L) \quad \Rightarrow \quad \exists x, L' : (ht(L) \equiv \iota_2(x, L') \wedge infinite(L')) & \\
fair(f, L) \quad \Rightarrow \quad ht(L) \equiv \iota_1 \vee (exists(f, L) \wedge ht(L) \equiv \iota_2(x, L') \wedge fair(f, L')) & \\
[] \sim x : L \quad \Rightarrow \quad False & (3) \\
x : L \sim [] \quad \Rightarrow \quad False & (4) \\
x : L \sim y : L' \quad \Rightarrow \quad x \equiv y \wedge L \sim L' & (5) \\
[] \not\sim x : L & (6) \\
x : L \not\sim [] & (7) \\
x : L \not\sim y : L' \quad \Leftarrow \quad x \not\equiv y \vee L \not\sim L' & (8)
\end{array}
$$

In terms of Assumption **??**, SP_i=TRIV(s)∪NAT, $extS = \{clist(s)\}$, $CO = \{[], \_ : \_\}$ and $d_{clist(s)} = ht$. Let B be a parameter model of TRIV(s). Hence $T^\infty_{B \cup CO, clist(s)}$ is the set of finite or infinite lists with entries in s^B and

$$F(A)_{clist(s)} = 1 + s^B \times A_{clist(s)}.$$

F coincides with the functor F of Example 14.2. Note that equations (1)-(8) present the general definition of $d_s^{Fin(F)}$ (see above) and the behavior axioms (see 5.1(2)), respectively, for $s = clist(s)$ and may thus be dropped.

Here is an alternative specification of *exists* and *forall* as Boolean functions, analogously to Example 14.2:

$$
\begin{array}{ll}
\texttt{destructs} & exists, forall : (s \to bool) \times clist(s) \to bool \\
\texttt{axioms} & exists(f, L) \equiv true \\
& \quad \Rightarrow \quad \exists x, L' : (ht(L) \equiv \iota_2(x, L') \wedge f(x) \text{ or } exists(f, L') \equiv true) \\
& exists(f, L) \equiv false \\
& \quad \Rightarrow \quad ht(L) \equiv \iota_1 \vee \exists x, L' : (ht(L) \equiv \iota_2(x, L') \wedge f(x) \text{ and } exists(f, L') \equiv false) \\
& forall(f, L) \equiv true \\
& \quad \Rightarrow \quad ht(L) \equiv \iota_1 \vee \exists x, L' : (ht(L) \equiv \iota_2(x, L') \wedge f(x) \text{ and } forall(f, L') \equiv true)
\end{array}
$$

$$forall(f,L) \equiv false$$
$$\Rightarrow \quad \exists x, L' : (ht(L) \equiv \iota_2(x,L') \wedge f(x) \textit{ or } forall(f,L') \equiv false)$$

The reason why *exists*, *forall* and the functions *length* and *min* of COLIST are declared as destructors and the other non-constructor functions as codefined functions will be given later.

The following ST specifies infinite number sequences:

COLIST' = COLIST$_{nat/s}$[NAT] `then`

`cofuncts`	$blink :\to clist(nat)$
	$nats : nat \to clist(nat)$
`vars`	$n : nat$
`axioms`	$ht(blink) \equiv \iota_2(0, 1 : blink)$
	$ht(nats(n)) \equiv \iota_2(n, nats(n+1))$❑

Example 15.3 The following specification of finite or infinite binary trees is a further parameterized swinging type that extends the parameter type TRIV(s):

COBINTREE[TRIV(s)] = NAT `and` LIST$_{bool/s}$[BOOL] `then`

`hidsorts`	$cbintree(s)$
`constructs`	$undef : 1 \to cbintree(s)$
	$\_\#\_\#\_ : cbintree(s) \times s \times cbintree(s) \to cbintree(s)$
	$\lambda y.\_(x,y) : (s \times s) \to bool) \to (s \to bool)$
`destructs`	$ler : cbintree(s) \to 1 + cbintree(s) \times s \times cbintree(s)$
	$size : cbintree(s) \to 1 + nat$
`cofuncts`	$<\_>: s \to cbintree(s)$
	$mirror : cbintree(s) \to cbintree(s)$
	$subtree : cbintree(s) \times list(bool) \to cbintree(s)$
`defuncts`	$\_ \in \_ : s \times cbintree(s) \to bool$
	$\$: ((s \to bool) \times s) \to bool$
`preds`	$exists : (s \to bool) \times cbintree(s)$
	$finite : cbintree(s)$
`copreds`	$forall : (s \to bool) \times cbintree(s)$
	$infinite : cbintree(s)$
	$\_ \sim \_ : cbintree(s) \times cbintree(s)$
	$\_ \sim \_ : (cbintree(s) \times cbintree(s)) \times (cbintree(s) \times cbintree(s))$
`vars`	$x,y : s \quad T,T',U,U' : cbintree(s) \quad b : bool \quad bL : list(bool) \quad f : s \to bool \quad g : s \times s \to bool$
	$k,m,n : nat$
`axioms`	$ler(T\#x\#T') \equiv \iota_2(T,x,T')$
	$ler(undef) \equiv \iota_3$
	$size(T) \equiv \iota_1$
	$\quad \Rightarrow \quad \exists x, T_1, T_2 : (ler(T) \equiv \iota_2(T_1,x,T_2) \wedge (size(T_1) \equiv \iota_1 \vee size(T_2) \equiv \iota_1))$
	$size(T) \equiv \iota_2(n)$
	$\quad \Rightarrow \quad \exists x, T_1, T_2 : (ler(T) \equiv \iota_2(T_1,x,T_2) \wedge size(T_1) \equiv \iota_2(k) \wedge size(T_2) \equiv \iota_2(m) \wedge k+m+1 \equiv n)$
	$ler(< x >) \equiv \iota_2(mt, x, mt)$
	$ler(mirror(T)) \equiv \iota_1 \quad \Leftarrow \quad ler(T) \equiv \iota_1$
	$ler(mirror(T)) \equiv \iota_2(mirror(U'), x, mirror(U)) \quad \Leftarrow \quad ler(T) \equiv \iota_2(U,x,U')$
	$subtree(T, []) \equiv \ T$
	$subtree(T, b : bL)) \equiv \iota_1 \quad \Leftarrow \quad ler(T) \equiv \iota_1$

$$
\begin{array}{ll}
subtree(T,true:bL)\equiv subtree(U,bL)) \;\;\Leftarrow\;\; ler(T)\equiv\iota_2(U,x,U') & \\
subtree(T,false:bL)\equiv subtree(U',bL) \;\;\Leftarrow\;\; ler(T)\equiv\iota_2(U,x,U') & \\
x\in T\equiv exists(\lambda y.eq(x,y),T) & \\
\$(\lambda y.g(x,y),y)\equiv g(x,y) & \\
exists(f,T) \;\;\Leftarrow\;\; ler(T)\equiv\iota_2(U,x,U')\wedge(exists(f,U)\vee f(x)\equiv true\vee exists(f,U')) & \\
finite(T) \;\;\Leftarrow\;\; ler(T)\equiv\iota_1\vee(ler(T)\equiv\iota_2(U,x,U')\wedge finite(U)\wedge finite(U')) & \\
forall(f,T) \;\;\Rightarrow\;\; ler(T)\equiv\iota_1\vee\exists x,U,U':(ler(T)\equiv\iota_2(U,x,U')\wedge forall(f,U)\;\wedge & \\
\qquad\qquad\qquad\qquad f(x)\equiv true\wedge forall(f,U'))\vee ler(T)\equiv\iota_3 & \\
infinite(T) \;\;\Rightarrow\;\; \exists x,U,U':(ler(T)\equiv\iota_2(U,x,U')\wedge(infinite(U)\vee infinite(U'))) & \\
T\equiv T' \;\;\Rightarrow\;\; ler(T)\equiv ler(T') & (3)\\
\iota_2(T,x,T')\sim\iota_2(U,y,U') \;\;\Rightarrow\;\; T\equiv T'\wedge x\equiv y\wedge U\equiv U' & (4)\\
\iota_2(T,x,T')\sim\iota_1 \;\;\Rightarrow\;\; False & (4)\\
\iota_1\sim\iota_2(T,x,T') \;\;\Rightarrow\;\; False & (4)
\end{array}
$$

In terms of Assumption **??**, SP_i=TRIV(s)∪NAT∪LIST(BOOL), $extS=\{cbintree(s),cbintree_1(s)\}$, $CO=\{mt,\_\#\_\#\_,def,undef\}$, $d_{cbintree(s)}=ler$ and $d_{cbintree_1(s)}=switch$. Let B be a parameter model of TRIV(s). Hence $T^\infty_{B\cup CO,cbintree(s)}$ is the set of finite or infinite binary trees with entries in s^B, $T^\infty_{B\cup CO,cbintree_1(s)}=1+T^\infty_{B\cup CO,cbintree(s)}$ and

$$
\begin{array}{l}
F(A)_{cbintree(s)}=1+A_{cbintree(s)}\times s^B\times A_{cbintree(s)},\\
F(A)_{cbintree_1(s)}=1+A_{cbintree(s)}.
\end{array}
$$

F coincides with the functor F of Example 14.4. Note that equations (1)-(14) present the general definition of $d_s^{Fin(F)}$ (see above) and the behavior axioms (see 5.1(2)), respectively, for $s\in\{cbintree(s),cbintree_1(s)\}$ and may thus be dropped.

Here is an alternative specification of *exists* and *forall* as Boolean functions, analogously to Example 14.4:

$$
\begin{array}{ll}
\texttt{destructs} & exists,forall:(s\to bool)\times cbintree(s)\to bool\\
\texttt{axioms} & exists(f,T)\equiv true\\
& \quad\Rightarrow\;\; \exists x,T_1,T_2:(ler(T)\equiv\iota_2(T_1,x,T_2)\wedge exists(f,T_1)\ or\ f(x)\ or\ exists(f,T_2)\equiv true)\\
& exists(f,T)\equiv false\\
& \quad\Rightarrow\;\; ler(T)\equiv\iota_1\vee\exists x,T_1,T_2:(ler(T)\equiv\iota_2(T_1,x,T_2)\;\wedge\\
& \qquad\qquad\qquad exists(f,T_1)\ and\ f(x)\ and\ exists(f,T_2)\equiv false)\\
& forall(f,T)\equiv true\\
& \quad\Rightarrow\;\; ler(T)\equiv\iota_1\vee\exists x,T_1,T_2:(ler(T)\equiv\iota_2(T_1,x,T_2)\;\wedge\\
& \qquad\qquad\qquad forall(f,T_1)\ and\ f(x)\ and\ forall(f,T_2)\equiv true)\\
& forall(f,T)\equiv false\\
& \quad\Rightarrow\;\; \exists x,T_1,T_2:(ler(T)\equiv\iota_2(T_1,x,T_2)\wedge forall(f,T_1)\ or\ f(x)\ or\ forall(f,T_2)\equiv false)
\end{array}
$$

The reason why *exists*, *forall* and the function *size* of COBINTREE are declared as destructors and the other non-constructor functions as codefined ones will be given later. ❏

Example 15.4 The following specification of finite or infinite trees is a parameterized swinging type with three new (hidden) sorts:

$$
\begin{array}{ll}
\multicolumn{2}{l}{\text{COTREE[TRIV}(s)] = \text{NAT } \texttt{and} \text{ LIST}_{nat/s}\text{[NAT] } \texttt{then}}\\
\quad\texttt{hidsorts} & ctree(s)\quad ctrees(s)\quad ctree_1(s)\\
\quad\texttt{constructs} & \_\&\_:s\times ctrees(s)\to ctree(s)
\end{array}
$$

	$[] :\to ctrees(s)$
	$\_ : \_ : ctree(s) \times ctrees(s) \to ctrees(s)$
	$undef : 1 \to ctree_1(s)$
	$def : ctree(s) \to ctree_1(s)$
	$\lambda y.\_(x,y) : (s \times s) \to bool) \to (s \to bool)$
`destructs`	$rs : ctree(s) \to s \times ctrees(s)$
	$ht : ctrees(s) \to 1 + s \times ctrees(s)$
	$switch : ctree_1(s) \to 1 + ctree(s)$
	$size : ctree(s) \to 1 + nat$
	$sizeL : ctrees(s) \to 1 + nat$
`cofuncts`	$<\_>: s \to ctree(s)$
	$subtree : ctree(s) \times list(bool) \to ctree_1(s)$
	$subtreeL : ctrees(s) \times list(bool) \to ctree_1(s)$
	$exists, forall : (s \to bool) \times bintree(s) \to bool$
`defuncts`	$\_ \in \_ : s \times bintree(s) \to bool$
	$\$: ((s \to bool) \times s) \to bool$
`preds`	$exists : (s \to bool) \times ctree(s)$
	$existsL : (s \to bool) \times ctrees(s)$
	$finite : ctree(s)$
	$finiteL, finiteB : ctrees(s)$
`copreds`	$forall : (s \to bool) \times ctree(s)$
	$forallL : (s \to bool) \times ctrees(s)$
	$infinite : ctree(s)$
	$infiniteL : ctrees(s)$
	$\_ \sim \_ : ctree(s) \times ctree(s)$
	$\_ \sim \_ : ctrees(s) \times ctrees(s)$
	$\_ \sim \_ : ctree_1(s) \times ctree_1(s)$
`vars`	$x, y : s \quad T, T' : ctree(s) \quad TL, TL' : ctrees(s) \quad m, n : nat \quad nL : list(nat)$
	$f : s \to bool \quad g : s \times s \to bool$
`axioms`	$rs(x\&TL) \equiv (x, TL)$ (1)
	$ht([]) \equiv \iota_1$ (2)
	$ht(T : TL) \equiv \iota_2(T, TL)$ (3)
	$switch(undef) \equiv \iota_1$ (4)
	$switch(def(T)) \equiv \iota_2(T)$ (5)
	$size(T) \equiv \iota_1 \quad \Rightarrow \quad rs(T) \equiv \iota_2(x, TL) \wedge sizeL(TL) \equiv \iota_1$
	$size(T) \equiv \iota_2(n)$
	$\quad \Rightarrow \quad rs(T) \equiv \iota_2(x, TL) \wedge sizeL(TL) \equiv \iota_2(m) \wedge m + 1 \equiv n$
	$sizeL(TL) \equiv \iota_1$
	$\quad \Rightarrow \quad ht(TL) \equiv \iota_2(T, TL') \wedge (size(T) \equiv \iota_1 \vee sizeL(TL') \equiv \iota_1)$
	$sizeL(TL) \equiv \iota_2(0) \quad \Rightarrow \quad ht(TL) \equiv \iota_1$
	$sizeL(TL) \equiv \iota_2(suc(n))$
	$\quad \Rightarrow \quad ht(TL) \equiv \iota_2(T, TL') \wedge size(T) \equiv \iota_2(k) \wedge sizeL(TL) \equiv \iota_2(m) \wedge k + m \equiv n$
	$rs(< x >) \equiv (x, [])$
	$switch(subtree(T, [])) \equiv \iota_2(T)$
	$switch(subtree(T, n : nL)) \equiv switch(subtreeL(TL, n : nL)) \quad \Leftarrow \quad rs(T) \equiv (x, TL)$
	$switch(subtreeL([], nl)) \equiv \iota_1$
	$switch(subtreeL(T : TL, [])) \equiv \iota_1$
	$switch(subtreeL(T : TL, 0 : nL)) \equiv switch(subtree(T, nL))$

$$
\begin{array}{ll}
switch(subtreeL(T:TL,suc(n):nL)) \equiv switch(subtreeL(TL,n:nL)) & \\
x \in T \equiv exists(\lambda y.eq(x,y),T) & \\
\$(\lambda y.g(x,y),y) \equiv g(x,y) & \\
exists(f,T) \quad \Leftarrow \quad rs(T) \equiv (x,TL) \wedge (f(x) \equiv true \vee existsL(f,TL)) & \\
existsL(f,TL) \quad \Leftarrow \quad ht(TL) \equiv \iota_2(T,TL') \wedge (exists(f,T) \vee existsL(f,TL')) & \\
finite(T) \quad \Leftarrow \quad rs(T) \equiv (x,TL) \wedge finiteL(TL) & \\
finiteL(TL) \quad \Leftarrow \quad ht(TL) \equiv \iota_1 \vee (ht(TL) \equiv \iota_2(T,TL') \wedge finite(T) \wedge finiteL(TL')) & \\
finiteB(TL) \quad \Leftarrow \quad ht(TL) \equiv \iota_1 \vee (ht(TL) \equiv \iota_2(T,TL') \wedge finiteB(TL')) & \\
forall(f,T) \quad \Rightarrow \quad \exists x,TL:(rs(T) \equiv (x,TL) \wedge f(x) \equiv true \wedge forallL(f,TL)) & \\
forallL(f,TL) & \\
\quad \Rightarrow \quad ht(TL) \equiv \iota_1 \vee \exists T,TL':(ht(TL) \equiv \iota_2(T,TL') \wedge forall(f,T) \wedge forallL(f,TL')) & \\
infinite(T) \quad \Rightarrow \quad \exists x,TL:(rs(T) \equiv (x,TL) \wedge infiniteL(TL)) & \\
infiniteL(TL) & \\
\quad \Rightarrow \quad finiteB(TL) \wedge \exists x,TL':(ht(TL) \equiv \iota_2(T,TL') \wedge (infinite(T) \vee infiniteL(TL'))) & \\
x\&TL \sim y\&TL' \quad \Rightarrow \quad x \equiv y \wedge TL \sim TL' & (6) \\
x\&TL \not\sim y\&TL' \quad \Leftarrow \quad x \not\equiv y \vee TL \not\sim TL' & (7) \\
[] \sim T:TL \quad \Rightarrow \quad False & (8) \\
T:TL \sim [] \quad \Rightarrow \quad False & (9) \\
T:TL \sim T':TL' \quad \Rightarrow \quad T \sim T' \wedge TL \sim TL' & (10) \\
[] \not\sim T:TL & (11) \\
T:TL \not\sim [] & (12) \\
T:TL \not\sim T':TL' \quad \Leftarrow \quad T \not\sim T' \vee TL \not\sim TL' & (13) \\
def(T) \sim undef \quad \Rightarrow \quad False & (14) \\
undef \sim def(T') \quad \Rightarrow \quad False & (15) \\
def(T) \sim def(T') \quad \Rightarrow \quad T \sim T' & (16) \\
def(T) \not\sim undef & (17) \\
undef \not\sim def(T') & (18) \\
def(T) \not\sim def(T') \quad \Leftarrow \quad T \not\sim T' & (19)
\end{array}
$$

In terms of Assumption **??**, SP_i=TRIV(s)∪NAT∪LIST(NAT), $extS = \{ctree(s), ctrees(s), ctree_1(s)\}$ and $CO = \{\_\&\_, [], \_:\_, def, undef\}$, $d_{ctree(s)} = rs$, $d_{ctrees(s)} = ht$ and $d_{ctree_1(s)} = switch$. Hence $T^{\infty}_{B\cup CO,ctree(s)}$ is the set of finite or infinite trees with entries in s^B and finite or infinite node degree, $T_{B\cup CO,trees(s)}$ is the set of finite or infinite forests with entries in s^B, $T^{\infty}_{B\cup CO,ctree_1(s)} = 1 + T^{\infty}_{B\cup CO,ctree(s)}$ and

$$
\begin{aligned}
F(A)_{ctree(s)} &= s^B \times A_{ctree(s)}, \\
F(A)_{ctrees(s)} &= 1 + A_{ctree(s)} \times A_{ctrees(s)}, \\
F(A)_{ctree_1(s)} &= 1 + A_{ctree(s)}.
\end{aligned}
$$

F coincides with the functor F of Example 14.5. Note that equations (1)-(19) present the general definition of $d_s^{Fin(F)}$ (see above) and the behavior axioms (see 5.1(2)), respectively, for $s \in \{ctree(s), ctrees(s), ctree_1(s)\}$ and may thus be dropped. The reason why the functions $size$ and $sizeL$ of COTREE are declared as destructors and the other non-constructor functions as codefined functions will be given later. ❑

$Fin(F)$ is also the initial **continuous** $(B \cup CO)$-algebra whose carriers are cpos and whose functions are continuous w.r.t. the cpo structure [38]. Given variables $x_1 \in X_{s_1}, \dots, x_n \in X_{s_n}$ and $f_1, \dots, f_n \in B \cup CO$, the set

$$E \;=\; \{x_1 \equiv f_1(x_{11}, \dots, x_{1k_1}), \dots, x_n \equiv f_n(x_{n1}, \dots, x_{nk_n})\}$$

of **regular** or (f_i-)**guarded** equations has a unique solution in $Fin(F)$ ([38], Theorem 5.2). E defines the infinite trees $t_1, \dots, t_n$ of $Fin(F)$ that arise from unfolding the graph presented by E. As part of a swinging type SP, E would be written as:

$$d_{s_1}(x_1) \equiv \iota_{f_1}(x_{11}, \ldots, x_{1k_1}), \ldots, d_{s_n}(x_n) \equiv \iota_{f_n}(x_{n1}, \ldots, x_{nk_n})$$

where d is the S-sorted destructor defined at the top of this section. The unique solvability of E in $Fin(F)$ can be shown easily: Suppose that $\{t_i\}_{i=1}^n$ and $\{u_i\}_{i=1}^n$ are two solutions of E. Let $R = \{(t_1, u_1), \ldots, (t_n, u_n)\}$. Since the greatest binary relation $\sim$ on $Fin(F)$ that satisfies the behavior axioms of SP (see Def. 5.1(2)) coincides with the diagonal of $Fin(F)^2$ (see Theorem ????), we only need to show that R also satisfies these axioms, which read here as follows: For all $1 \leq i \leq n$,

$$x \sim y \quad \Rightarrow \quad d_{s_i}(x) \sim d_{s_i}(y).$$

So let $(t_i, u_i) \in R$. Then

$$d_{s_i}(t_i) \equiv \iota_{f_i}(t_{i1}, \ldots, t_{ik_i}) \quad \text{and} \quad d_{s_i}(u_i) \equiv \iota_{f_i}(u_{i1}, \ldots, u_{ik_i}).$$

Since for all $1 \leq j \leq k_i$, $(t_{ij}, u_{ij}) \in R$, the proof is complete.

16 Destructor-based coalgebras

F is built up of coproducts and finite products. The following functor $G : Set^S \to Set^S$ involves also infinite products, i.e., function spaces. In contrast to F, G covers hidden data types with *parameterized* destructors. G is cocontinuous, but not necessarily continuous. While F is induced by a signature Σ of constructors with hidden range sort, G is associated with a signature Δ of destructors with a unique hidden argument. Hence w.l.o.g. for all $f : sw \to s' \in \Delta$, $s \in hidS$. Given a $visS$-sorted set C, $G : Set^S \to Set^S$ is defined as follows: for all $A \in Set^S$ and $s \in S$,

$$G(A)_s \;=\; \begin{cases} s^C & \text{if } s \in visS, \\ \prod_{f:sw \to ran \in \Delta}[w^C \to ran^A] & \text{if } s \in hidS. \end{cases}$$

The **Nerode** or **contextual SP-equivalence**, $\sim_{SP}^{Ner}$ is the set of all ground term pairs (t, t') such that for all Σ-contexts $c : sw \to s'$ and $u \in T_{\Sigma,w}$, $c(t, u) \equiv_{SP} c(t', u)$.

As Δ agrees with the set $Obs(\Xi)$ of observations of a (Ω, Ξ)-signature ([61], Def. 6.1), so CT_Σ coincides with the set $Cont(\Xi)$ of Ξ-contexts constructed from $Obs(\Xi)$ ([61], Def. 6.2).

If A is a Herbrand structure, then we write c for c^A (cf. Section 2). The values of contexts determine the contextual equivalence of terms (see below). The quotient of T_Σ by $\sim_{SP}^{Ner}$ is also called the **final realization of the behavior of** C (cf. [78], Section 5).

Since G is cocontinuous, Thm. 21.2 implies that $coAlg(G)$ has a final object $fin : Fin(G) \to G(Fin(G))$. $Fin(G)$ can be represented as a product of function spaces: for all $s \in S$,

$$Fin(G)_s \;=\; \begin{cases} s^C & \text{if } s \in visS, \\ \prod_{c:sw \to s' \in CT_\Sigma}[C_w \to C_{s'}] & \text{if } s \in hidS. \end{cases}$$

CoCASL [83] also admits parameterized destructors. However, the semantics of the hidden data types is not given by the final coalgebra $Fin(G)$, but a—probably isomorphic—*behavior algebra* $Beh_\Sigma(C)$, which generalizes the infinite-term algebra $T_{\Sigma \cup C}^\infty$ of the previous section to parameterized destructors. The trees of $Beh_\Sigma(C)$ may not only have infinite paths, but also infinite outdegree. Inner nodes of $Beh_\Sigma(C)$ are labelled with hidden sorts. Let n be an inner node labelled with s. Then for each destructor $d : sw \to s'$ and each $c \in C_w$, n has a direct successor with label s' and a direct successor with label c. Hence the out degree of n is infinite if C_w is infinite.

If Δ lacks destructors with visible range sort, then CT_Σ is empty and thus $Fin(G)$ is a one-element set! In Section 3.2, a signature Σ consisting of constructors with hidden range sort was associated with a functor F. In Section 3.3, a cosignature Δ of unary destructors was derived from Σ.

The set of instances of a class *cl* declared in an object-oriented program agrees with the product $\prod_{a\in Att} Val_a$ where Att is the set of attributes of *cl* and Val_a is the set of possible values of a. If considered as a set of unary functions, Att is the set of contexts and hence usually finite. However, if, for instance, some attributes denote (references to) other object of class *cl*, we obtain infinitely many contexts and, as a final G-coalgebra, the semantics of *cl* is an infinite product.

Example 16.1 Infinite sequences are specified as follows.

STREAM = LIST `and` ENTRY($entry'$) `then`

`hidsorts`	$stream = stream(entry)$ $stream' = stream(entry')$	
`destructs`	$head : stream \to entry$	
	$tail : stream \to stream$	
`constructs`	$\_\&\_ : entry \times stream \to stream'$	
	$blink : \to stream(nat)$	
	$nats : nat \to stream(nat)$	
	$zip : stream \times stream \to stream$	
	$map : (entry \to entry') \times stream \to stream'$	
`defuncts`	$\_\#\_ : list \times stream \to stream$	
	$nth : nat \times stream \to entry$	
	$nthtail : nat \times stream \to stream$	
`static preds`	$exists : (entry \to bool) \times stream$	
`copreds`	$forall : (entry \to bool) \times stream$	
	$fair : (entry \to bool) \times stream$	
`vars`	$n : nat \quad x, y : entry \quad L : list \quad s, s' : stream$	
	$f : entry \to entry' \quad g : entry \to bool$	
`axioms`	$head(x\&s) \equiv x$	$tail(x\&s) \equiv s$
	$head(blink) \equiv 0$	$tail(blink) \equiv 1\&blink$
	$head(nats(n)) \equiv n$	$tail(nats(n)) \equiv nats(n+1)$
	$head(zip(s, s')) \equiv head(s)$	$tail(zip(s, s')) \equiv zip(s', tail(s))$
	$head(map(f, s)) \equiv f(s)$	$tail(map(f, s)) \equiv map(f, tail(s))$
	$[]\#s \equiv s$	
	$(x : L)\#s \equiv x\&(L\#s)$	
	$nth(0, s) \equiv head(s)$	
	$nth(n+1, s) \equiv nth(n, tail(s))$	
	$nthtail(0, s) \equiv s$	
	$nthtail(n+1, s) \equiv nthtail(n, tail(s))$	
	$exists(g, s) \quad \Leftarrow \quad g(head(s)) \equiv true$	
	$exists(g, s) \quad \Leftarrow \quad exists(g, tail(s))$	
	$forall(g, s) \quad \Rightarrow \quad g(head(s)) \equiv true \wedge forall(g, tail(s))$	
	$fair(g, s) \;\Rightarrow\; exists(g, s) \wedge fair(g, tail(s))$	

& appends an entry to a stream. *blink* denotes a stream whose elements alternate between zeros and ones. $nats(n)$ generates the stream of all numbers starting from n. *zip* merges two streams into a single stream by alternatively appending an element of one stream to an element of the other stream. # concatenates a list and a stream into a stream. $fair(g, s)$ holds true iff s contains infinitely many elements satisfying g.

Let SP = STREAM, $visS = \{entry\}$, $hidS = \{stream\}$, $\Sigma = \{\&\}$, $\Delta = \{(head, tail)\}$, $C = Ini(SP)$ and $F(A)_{stream} = C_{entry} \times A_{stream}$. $Fin(SP)_{stream}$ is embedded in

$$T^{\infty}_{\Sigma\cup C,stream} \;=\; \{a_1\&a_2\&\dots \mid a_1, a_2, \dots \in C_{entry}\}.❏$$

Example 16.2 The final models of the following types are embedded in final G-coalgebras:

SET = LIST `then`

`hidsorts`	$set = set(entry)$	
`destructs`	$in : entry \times set \to bool$	
`constructs`	$\emptyset, all :\to set$	
	$\{\_\} : entry \to set$	
	$\_\cup\_ : set \times set \to set$	
	$\_\setminus\_ : set \times set \to set$	
	$compr : (entry \to bool) \to set$	set comprehension
`vars`	$x, y : entry \quad s, s' : set \quad g : entry \to bool$	
`axioms`	$in(x, \emptyset) \;\equiv\; false$	
	$in(x, all) \;\equiv\; true$	
	$in(x, \{y\}) \;\equiv\; eq(x, y)$	
	$in(x, s \cup s') \;\equiv\; in(x, s) \ or \ in(x, s')$	
	$in(x, s \setminus s') \;\equiv\; in(x, s) \ and \ not(in(x, s'))$	
	$in(x, compr(g)) \;\equiv\; g(x)$	

BAG = LIST `then`

`hidsorts`	$bag = bag(entry)$
`destructs`	$card : bag \times entry \to nat$
`constructs`	$empty :\to bag$
	$[\_] : entry \to bag$
	$\_+\_ : bag \times bag \to bag$
	$\_-\_ : bag \times bag \to bag$
`vars`	$x, y : entry \quad b, b' : bag$
`axioms`	$card(empty, x) \;\equiv\; 0$
	$card([x], x) \;\equiv\; 1$
	$card([x], y) \;\equiv\; 0 \quad \Leftarrow \quad x \not\equiv y$
	$card(b + b', x) \;\equiv\; card(b, x) + card(b', x)$
	$card(b - b', x) \;\equiv\; card(b, x) - card(b', x)$

WSET = LIST `and` INT[17]`then`

`hidsorts`	$wset = wset(entry)$
`destructs`	$weight : wset \times entry \to int$
`constructs`	$empty :\to wset$
	$[\_] : entry \to wset$
	$\_+\_ : wset \times wset \to wset$
	$\_-\_ : wset \to wset$
`vars`	$x, y : entry \quad V, W : wset$
`axioms`	$weight(empty, x) \equiv 0$
	$weight([x], x) \equiv 1$
	$weight([x], y) \equiv 0 \quad \Leftarrow \quad x \not\equiv y$
	$weight(V + W, x) \equiv weight(V, x) + weight(W, x)$

$$weight(V - W, x) \equiv weight(V, x) - weight(W, x)$$

MAP = ENTRY($domain$) `and` ENTRY($range$) `then`

`hidsorts`	$map = map(domain, range)$
`destructs`	$get : map \times domain \to range$
`constructs`	$new : range \to map$
	$upd : domain \times range \times map \to map$
`vars`	$i, j : domain \quad x : range \quad f : map$
`axioms`	$get(new(x), i) \equiv x$
	$get(upd(i, x, f), i) \equiv x$
	$get(upd(i, x, f), j) \equiv get(f, j) \quad \Leftarrow \quad i \not\equiv j$

Let SP = SET, $visS = \{entry, bool\}$, $hidS = \{set\}$, $\Delta = \{in\}$, $C = Ini(SP)$, $G(A)_{set} = [C_{entry} \to C_{bool}]$.

Let SP = BAG, $visS = \{entry, nat\}$, $hidS = \{bag\}$, $\Delta = \{card\}$, $C = Ini(SP)$, $C_{nat} = \mathbb{N}$ and $G(A)_{bag} = [C_{entry} \to \mathbb{N}]$.

Let SP = WSET, $visS = \{entry, int\}$, $hidS = \{wset\}$, $\Delta = \{weight\}$, $C = Ini(SP)$ and $G(A)_{wset} = [C_{entry} \to \mathbb{Z}]$.

Let SP = MAP, $visS = \{domain, range\}$, $hidS = \{map\}$, $\Delta = \{get\}$, $C = Ini(SP)$ and $G(A)_{map} = [C_{domain} \to C_{range}]$. ❑

Example 16.3 The classical examples of G-coalgebras are deterministic automata, which realize (partial) functions from the set of words over some fixed set C_{in} of inputs into some fixed set C_{out} of outputs. Here Δ, G and $Fin(G)$ read as follows: Let $A \in Set^S$.

- *Moore-automata.* $\Delta = \{\delta : state \times in \to state,\ \beta : state \to out\}$,
  $G(A)_{state} = [C_{in} \to A_{state}] \times C_{out}$, $Fin(G)_{state} \cong [C_{in}^* \to C_{out}]$.
- *Mealy-automata.* $\Delta = \{\delta : state \times in \to state,\ \beta : state \times in \to out\}$,
  $G(A)_{state} = [C_{in} \to A_{state}] \times [C_{in} \to C_{out}]$, $Fin(G)_{state} \cong [C_{in}^+ \to C_{out}]$. ❑

Lemma 16.4 *Let SP be a continuous and behaviorally consistent specification without hidden constructors and logical predicates. Let Δ be the set of destructors of SP. Based on $C = Ini(SP)$, define G as above. Then there is an injective Σ-homomorphism from $Fin(SP)$ to $Fin(G)$.*

Proof. Let $SP = (\Sigma, AX)$ and $A = Fin(G)$. For all $t \in T_\Sigma$, $[t]$ denotes the $\equiv_{SP}$-equivalence class of t. For all $c : w \to s \in CT_\Sigma$, π_c denotes the projection of A to $[C_w \to C_s]$. A function $h : Her(SP) \to A$ is defined as follows: for all $s \in visS$ and $t \in T_{\Sigma,s}$, $h(t) = [t]$, while for all $s \in hidS$, $t \in T_{\Sigma,s}$, $c : sw \to s' \in CT_\Sigma$ and $u \in T_{\Sigma,w}$,

$$\pi_c(h(t))([u]) \ =_{def} \ [c(t, u)].$$

We show that the equivalence kernel $\sim_h$ of h coincides with $\sim_{SP}$. Let $s \in S$ and $t, t' \in T_{\Sigma,s}$. $h(t) = h(t')$ holds true iff $t \sim_{SP}^{Ner} t'$. Analogously to the proof of Lemma 18.3 (see below) one may show that behavioral SP-equivalence coincides with contextual SP-equivalence and thus with $\sim_h$.

Since SP is behaviorally consistent and $\sim_{SP} = \sim_h$, $\sim_h$ is a Σ-congruence. Hence $h(Her(SP))$ becomes a Σ-structure and h a Σ-homomorphism if one defines $f^{h(Her(SP))}(h(t)) = h(f(t))$ for all $f : w \to s \in \Sigma$ and $t \in T_{\Sigma,w}$. Consequently, h induces an injective Σ-homomorphism $h^* : Her(SP)/\sim_h \to A$. Hence $Fin(SP) = Her(SP)/\sim_{SP} = Her(SP)/\sim_h$ is embedded in $A = Fin(G)$. ❑

Data types often involve several hidden sorts whose meaning distributes over initial F-algebras, final F-coalgebras and final G-coalgebras. For arbitrary endofunctors F and G on the same category $\mathcal{K}$, F, G**-objects**

[17] For a specification of integer numbers, see Example 19.1.

[68], F, G-**bialgebras** [24] and F, G-**structures** [49, 61][18] deal with pairs of an F-algebra and a G-coalgebra. [25] also considers F, G-**dialgebras**, i.e., morphisms from $F(A)$ to $G(A)$ for some $A \in \mathcal{K}$. Here each hidden sort is interpreted as the object of either the initial F-algebra or the final G-algebra.

In most applications, including those discussed in the above-mentioned papers, F and G are instances of the schemas presented in the previous two sections. What goes beyond are non-deterministic structures where behavioral equality is determined by transition predicates, such as labelled transition systems (LTS). In the coalgebraic setting, these structures are modelled as solutions of domain equations involving powerset functors. Unfortunately, the construction of the corresponding models becomes much less elegant and useful as a semantical basis of specification if one proceeds from polynomial to powerset functors (cf., e.g., [105, 104, 19]. Given an LTS with a term-generated set of states, one may keep to presenting the LTS as a relation, in terms of swinging types: as a dynamic predicate. A relational presentation is also adequate and does not cause model- or proof-theoretical problems if the LTS is not used as a transition predicate, i.e., if it does not determine a behavioral equality (cf. Def. 5.1). As one may conclude from Section 5, only if the state set is uncountable *and* the LTS determines a behavioral equality, the swinging type approach enforces us to replace the relational presentation of an LTS. In this case one should start out from a functional simulation of the LTS, say $\rightarrow$: $state \times label \times state$, and specify a destructor $sucs : state \times label \rightarrow state^*$ such that for all states $s, s_1, \ldots, s_n$ and labels x,

$$sucs(s, x) = (s_1, \ldots, s_n) \quad \text{implies} \quad \{s_1, \ldots, s_n\} = \{s' \mid s \xrightarrow{x} s'\}.$$

Of course, $sucs$ induces a finer behavioral equivalence than $\rightarrow$ would do if $\rightarrow$ were declared as a transition predicate. But even the desired equivalence can be achieved if one extends the dialgebraic swinging type whose behavioral equality is induced by $sucs$ to an algebraic ST:

LTS = SUCS `then`

`hidsorts`	$state'$	
`constructs`	$mkstate' : state \rightarrow state'$	
`transpreds`	$\_ \xrightarrow{\_} \_ : state' \times label \times state'$	
`vars`	$s, s_1, \ldots, s_n : state \quad x : label$	
`axioms`	$mkstate'(s) \xrightarrow{x} mkstate'(s_i) \quad \Leftarrow \quad sucs(s, x) \equiv (s_1, \ldots, s_n)$	for all $1 \leq i \leq n$

This step from a destructor to a transition predicate somewhat reflects the *natural transformation* from a tree constructing functor to a powerset functor given in [105], Section 3.4. It suggests that category-theoretical "weapons" like natural transformations need not be employed in order to accomplish an adequate model of a specification whose behavioral equivalence is determined by an LTS.

17 More on final coalgebras

Definition 17.1 (final coalgebra) Let $SP = (\Sigma, AX)$ be a coalgebraic type with cosignature $\Delta = (visS, hidS, FD)$ and visible subtype $visSP$ and C be a $visSP$-model with equality. The following S-sorted set P collects the "observations" or "measurements" on hidden objects in a product:

$$P_s \;=\; \begin{cases} \prod_{d:s\rightarrow(w\rightarrow s')\in CT_\Delta}[C_w \rightarrow C_{s'}] & \text{if } s \in hidS, \\ C_s & \text{if } s \in visS. \end{cases}$$

For all contexts $d : s \rightarrow (w \rightarrow s') \in CT_\Delta$, π_d denotes the projection from P_s to $[C_w \rightarrow C_{s'}]$. d defines an "experiment", which, if performed on a under the "initial condition" $c \in C_w$, returns the visible result $\pi_d(a)(c) \in C_{s'}$.

[18] In terms of [49, 61], $F = \Omega$ and $G = \Xi$.

Projections for individual contexts can be extended to sums of contexts: Let $\{e_i : s_i \to (w_i \to s'_i)\}_{i\in I} \subseteq CT_\Delta \cup ID$ and $e = \amalg_{i\in I} e_i$. Then

$$\pi_e : \amalg_{i\in I} P_{s_i} \to \amalg_{i\in I}[C_{w_i} \to C_{s'_i}]$$

is defined as follows: for all $i \in I$,

$$\pi_e \circ \iota_i \;=\; \begin{cases} \iota_i \circ \pi_{e_i} & \text{if } e_i \in CT_\Delta, \\ \iota_i & \text{if } e_i \in ID. \end{cases}$$

Let $\mathcal{P}(P)$ be the set of S-sorted subsets of P. A functional $\Phi : \mathcal{P}(P) \to \mathcal{P}(P)$ is defined as follows: for all $s \in S$ and $A \subseteq P$,

$$\Phi(A)_s \;=\; \begin{cases} \{a \in A_s \mid \begin{array}{l} \forall\ s' = \amalg_{i\in I} s_i \in hidS,\ d : s \to (v \to s') \in FD,\ c \in C_v \\ \exists\ i \in I,\ b \in s_i^A\ \forall\ e : s' \to (w \to s'') \in CT_\Delta,\ c' \in C_w : \\ \pi_e(\iota_i(b))(c') = \pi_{d\cdot e}(a)(c,c') \qquad (1) \end{array} \} & \text{if } s \in hidS, \\ C_s & \text{if } s \in visS. \end{cases}$$

Since Φ is monotone, Φ has a greatest fixpoint $gfp(\Phi)$ (cf. Thm. 8.3). The **final** (SP,C)**-coalgebra** $Fin = Fin(SP,C)$ is the (SP,C)-coalgebra such that for all $s \in hidS$,

- $Fin_s = gfp(\Phi)_s$,
- for all $d : s \to (w \to s') \in FD$, $a \in Fin_s$ and $c \in C_w$,

$$d^{Fin}(a)(c) \;=\; \begin{cases} \iota_i(b) \text{ such that (1) holds true} & \text{if } s' \in hidS, \\ \pi_d(a)(c) & \text{if } s' \in visS. \end{cases}$$

Suppose that (1) has two solutions in i and b. Then (1) implies that both solutions are identical. Hence d^{Fin} is well-defined. ❏

$$\pi_e(d^{Fin}(a)) =_{def} \pi_{d\cdot e}(a)$$

Beispiel: finaler Automat: $\delta(f,x)(w) = f(xw)$

Condition 17.1(1) selects those tuples a among the elements of the product P_s such that for each two contexts $e_1 : s \to s_1, e_2 : s \to s_2$ containing the same path p that is taken when a is observed by e_1, then p is also taken when a is observed by e_2, and both observations lead to the same result.

Contexts are dual to terms, in particular to normal forms. Contexts are composed of destructors. Normal forms are composed of constructors, while contexts are composed of destructors. The structure of a context is determined by the coarities of the destructors it is built of. The structure of a normal form is determined by the arities of the constructors it is composed of. Hence both contexts and normal forms are trees whose nodes are labelled with function symbols: destructors and constructors, respectively. In the first case, the outdegree of a node is the coarity of the node label, in the second case, it is the arity. Except for [15], contextual characterizations of final coalgebras have only been given for cosignatures where all destructors are linear (cf. [100, 34, 67, 49, 61]). Only [15] handles the important generalization to sum sorts and works out its impact on coalgebraic specifications.

If all destructors are linear, then the final (Δ, C)-coalgebra is the entire product of context ranges:

Proposition 17.2 *Let SP, Δ and C be as in Def. 17.1 such that all destructors are linear. Then for all $s \in hidS$, $Fin(SP,C)_s = P_s$.*

Proof. Let $d : s \to (s \to s') \in FD$. Since s' has coarity 1, we may identify b with $\iota_i(b)$. Hence (1) amounts to a definition of b: for all $e : s' \to (w \to s'') \in CT_\Delta$ and $c' \in C_w$, $\pi_e(b)(c') =_{def} \pi_{d\cdot e}(a)(c,c')$. Therefore, $gfp(\Phi)_s = P_s$. ❏

The restriction to the linear case excludes many "properly coalgebraic" types such as COLIST (Ex. 15.2), finite and infinite trees (cf. Ex. 19.5), regular graphs ([91], Section 2.2), processes ([91], Section 4.4) and class diagrams involving inheritance relationships ([91], Section 6). The algebraic counterpart of final coalgebras with linear destructors are initial algebras with unary constructors only, i.e., sets of *words*. As the expressiveness of words is limited, so is the expressiveness of linear destructors.

Contexts and normal forms are also complementary with respect to the traversal of their tree representations. A normal form is evaluated bottom-up, starting from the leaves, along *all* paths up to the root where the value *of the object itself* is returned. A context is evaluated top-down, starting from the root, along a *single* path consisting of destructors until a leaf is reached where the result *of an observation* is delivered (cf. Fig. **??**). Hence, from an operational point of view, contexts are flowcharts or transition systems rather than normal forms representing static objects. The evaluations of a context encompass *possible* worlds (states, movements, interferences, etc.). But the various possibilities can never be viewed in parallel. Once an object is observed, its *multiverse* contracts to a universe (cf. [13]).

A normal form *is* the object it denotes. An evaluation of a context only represents the *behavior* of an actually invisible object in a certain *state*. Hence a final coalgebra can be seen as a Kripke structure that may interpret the same constants, function symbols and predicates differently in different states. Just regard the set of contexts as a signature Σ. Then a Σ-algebra is both the interpretation of a state in the corresponding Kripke model and an *element* of a final coalgebra.

Theorem 17.3 *Given the assumptions of Def. 17.1, $Fin(SP,C)$ is final in $coAlg(SP,C)$. Moreover, $Fin(SP,C)$ interprets behavioral equalities as identities.*

Proof. Let $Fin = Fin(SP,C)$ and B be an (SP,C)-coalgebra. A function $fin : B \to Fin$ is defined as follows: for all $s \in S$ and $b \in s^B$,

$$\begin{array}{rcllll} fin(b) & = & b & & \text{if } s \in visS, \\ \pi_e(fin(b)) & = & e^B(b) & \text{for all } e : s \to s' \in CT_\Delta & \text{if } s \in hidS, \\ fin(\iota_i(b)) & = & \iota_i(fin(b)) & & \text{if } s = w_i \text{ for a sum sort } \amalg_{i\in I} w_i. \end{array}$$

fin is a Σ-homomorphism: Let $s' \in visS$, $d : s \to (w \to s') \in FD$, $b \in s^B$ and $c \in C_w$. Hence

$$fin(d^B(b)(c)) \;=\; d^B(b)(c) \;=\; \pi_d(fin(b))(c) \;=\; d^{Fin}(fin(b))(c). \tag{2}$$

Let $s' = \amalg_{i\in I} s_i \in hidS$, $d : s \to (v \to s') \in FD$, $e = \amalg_{i\in I} e_i : s' \to (w \to s'') \in CT_\Delta$, $b \in s^B$, $c \in C_v$ and $c' \in C_w$. $d^B(b)(c) = \iota_i(a)$ implies

$$\begin{array}{rl} & \pi_e(fin(d^B(b)(c)))(c') \;=\; \pi_e(fin(\iota_i(a)))(c') \;=\; \pi_e(\iota_i(fin(a)))(c') \;=\; \iota_i(\pi_{e_i}(fin(a)))(c') \\ = & \iota_i(e_i^B(a))(c') \;=\; e^B(\iota_i(a))(c') \;=\; e^B(d^B(b)(c))(c') \;=\; (d\cdot e)^B(b)(c,c') \;=\; \pi_{d\cdot e}(fin(b))(c,c') \\ & \stackrel{Def.\ 17.1}{=} \; \pi_e(d^{Fin}(fin(b))(c))(c'). \end{array} \tag{3}$$

Hence $fin(d^B(b)(c)) = d^{Fin}(fin(b))(c)$. A permutation of the terms in (2) and (3) shows that fin is the only (Δ, C)-homomorphism from B to Fin.

Let BE be the set of behavioral equalities if Σ' and $\mathcal{C}(A)$ be the class of Σ-structures B such that $B|_{\Sigma\setminus BE} = A =_{def} Fin|_{\Sigma\setminus BE}$. Since Fin is canonical, the fixpoint theorem of Knaster and Tarski (Thm. 8.3) implies that for all $s \in hidS$,

$$\sim_s^A \;=\; \bigcup\{\sim_s^B \subseteq A_w \mid B \in \mathcal{C}(A),\ \sim_s^B \subseteq \sim_s^{\Phi(B)}\}.$$

Hence the diagonal of A^2 is a subset of $\sim_s^A$. Conversely, let $a \sim_s^A b$. Then there is $B \in \mathcal{C}(A)$ such that $(a,b) \in \sim_s^B \subseteq \sim_s^{\Phi(B)}$. Hence for all $d : sw \to s' \in des$ and $c \in B_w = C_w$, $d^B(a,c) \sim_{s'}^B d^B(b,c)$. Consequently, for all

$e : s \to (w \to s') \in CT_\Delta$ and $c \in C_w$, $e^B(a)(c) \sim^B_{s'} e^B(b)(c)$ and thus $\pi_e(a)(c) = e^B(a)(c) = e^B(b)(c) = \pi_e(b)(c)$ because s' is visible and $B|_{vis\Sigma} = C$ is a structure with $\sim$-equality. Hence $a = b$. ❑

We recapitulate the definition of the functor $G : Set^S \to Set^S$ given in Section 3.4 using the notation of Def. **??**: for all $A \in Set^S$ and $s \in S$,

$$G(A)_s = \begin{cases} \prod_{d:s\to(w\to s')\in FD}[C_w \to s'^A] & \text{if } s \in hidS, \\ C_s & \text{if } s \in visS. \end{cases}$$

Each (Δ, C)-coalgebra B in the sense of Def. **??** induces a G-coalgebra $\beta_B : B \to G(B)$: for all $s \in S$ and $b \in s^B$,

$$\begin{array}{rcll} \pi_d(\beta(b)) & = & d^B(b) \quad \text{for all } d : s \to (w \to s') \in FD & \text{if } s \in hidS, \\ \beta(b) & = & b & \text{if } s \in visS. \end{array}$$

Conversely, a G-coalgebra $\beta : B \to G(B)$ equips B with an interpretation of Δ: for all $d : s \to (w \to s') \in FD$ and $b \in s^B$, $d^B(b) =_{def} \pi_d(\beta(b))$. Hence G-coalgebras and (Δ, C)-coalgebras correspond to each other. Since G is cocontinuous, the limit of the chain

$$1 \leftarrow G(1) \leftarrow G^2(1) \leftarrow \ldots$$

is a final G-coalgebra $Fin(G)$ (cf. Thm. 21.2). In fact, $\beta = fin$. The proof proceeds analogously to the proof of Thm. 17.3.

The functor $F : Set^S \to Set^S$ of Section 3.2 can also be adapted to the assumptions of Def. **??**: for all $A \in Set^S$ and $s \in S$,

$$F(A)_s = \begin{cases} \coprod_{f:w\to s\in\Sigma} A_w & \text{if } s \in hidS, \\ C_s & \text{if } s \in visS. \end{cases}$$

The initial F-algebra $Ini(F)$ is given by the algebra of ground $(\Sigma \cup C)$-terms (cf. Section 3.2). Both initial F-algebras and final G-coalgebras are constructed from sums and products, but in opposite order: for all $s \in hidS$, $Ini(F)_s$ can be represented by the sum

$$\coprod_{f:w\to s\in\Sigma} T_{\Sigma\cup C,w}$$

of sets of term tuples, while $Fin(G)_s$ is (a subset of) the product

$$\prod_{d:s\to(w\to s')\in CT_\Delta} [C_w \to C_{s'}]$$

of context ranges (cf. Def. 17.1).

Definition 17.4 (coinductive, coequational type and inductive type) Let $SP = (\Sigma, AX)$ be a swinging type satisfying 5.1(5) such that the base type $baseSP$ of SP is coalgebraic and has cosignature Δ and visible subtype $visSP$. Then SP is a **coinductive type** with cosignature Δ and visible subtype $visSP$. ❑

Theorem 17.5 *Let $SP = (\Sigma, AX)$ be a coinductive type with base type $baseSP = (base\Sigma, baseAX)$, extension (Σ', AX'), cosignature $\Delta = (visS, hidS, FD)$ and visible subtype $visSP = (vis\Sigma, visAX)$. Let C be a $visSP$-model with equality such that $A = Fin(baseSP, C)$ is a canonical $baseSP$-model. Moreover, suppose that*

- *for all $(d(f(x))(t) \equiv u \Leftarrow \varphi) \in AX'$ and $f, g \in \Sigma'$, u does not contain a subterm of the form $f(\ldots, g(\ldots), \ldots)$,*
- *for all $d : s \to (v \to s') \in DS$, $f : w \to s \in \Sigma'$, $a \in A$ and $b : X \to A$ there is exactly one $(d(f(x))(t) \equiv u \Leftarrow \varphi) \in AX'$ such that $a = b^*(t)$ and $A \models_b \varphi$.*

Then there is a canonical SP-model $Fin(SP, C)$ with $Fin(SP, C)|_{base\Sigma} = A$.

Proof. Σ' consists of coinductive functions. We interpret Σ' on A and show that the axioms for Σ' are valid in A. For this purpose, we construct an $(baseSP, C)$-coalgebra B such that the final morphism $fin : B \to A$

yields an interpretation of Σ' in A. Let $hidS'$ be the set of all $s \in hidS$ such that there is $f : w \to s \in \Sigma'$. B is defined as follows: for all $s \in S$,

$$s^B = \begin{cases} \coprod_{f:w\to s\in\Sigma'} A_w & \text{if } s \in hidS', \\ s^A & \text{otherwise.} \end{cases}$$

We interpret each $f : w \to s \in \Sigma'$ in B by the injection $\iota_f : A_w \to s^B$. Let $T_\Sigma(X)'$ be the set of all Σ-terms t such that t does not contain a subterm of the form $f(\ldots, g(\ldots), \ldots)$. Let $d : s \to (v \to s') \in DS$, $f : w \to s \in \Sigma'$, $a \in A$, $b : X \to A$ and $c : X \to B$ such that for all $x \in X$, $b(x) = c(x)$. By assumption, there is exactly one $(d(f(x))(t) \equiv u \Leftarrow \varphi) \in AX'$ such that $a = b^*(t)$ and $A \models_b \varphi$. Hence d^B is defined uniquely as follows:

$$\begin{array}{ll} d^B(\iota_f(bx))(a) =_{def} c^*(u) & \text{if } s \in hidS', \\ d^B =_{def} d^A & \text{otherwise.} \end{array}$$

It is crucial that $u \in T_\Sigma(X)'$. Otherwise $c^*(u)$ were not defined because for some $f : w \to s \in \Sigma'$ occurring in u, $A_w \neq B_w$ and thus $f^B = \iota_f$ would not be a function from B_w to s^B.

Of course, B is a $(baseSP, C)$-coalgebra. Hence by Thm. 17.3 there is a unique $base\Sigma$-homomorphism $fin : B \to A$. Hence A may interpret $f : w \to s \in \Sigma'$ as follows: for all $a \in A_w$,

$$f^A(a) =_{def} fin_s(\iota_f(a)). \tag{1}$$

With this interpretation of coinductive functions, A satisfies AX': Let $(d(f(x))(t) \equiv u \Leftarrow \varphi) \in AX'$, $a \in A$ and $b : X \to A$ such that $a = b^*(t)$ and $A \models_b \varphi$. Hence by (1), the definition of d^B, since fin is compatible with DF and since both $fin_{s'} \circ c^*$ and b^* are Σ-homomorphic extensions of b to $T_\Sigma(X)'$,

$$\begin{array}{l} b^*(d(f(x))(t)) = d^A(f^A(bx))(b^*(t)) = d^A(f^A(bx))(a) = d^A(fin_s(\iota_f(bx)))(a) \\ = fin_{s'}(d^B(\iota_f(bx))(a)) = fin_{s'}(c^*(u)) = b^*(u), \end{array}$$

i.e., $A \models_b d(f(x))(t) \equiv u$. ❏

Besides the set of destructors that is introduced by a cosignature Δ, a cospecification CSP with cosignature Δ allows us to axiomatize functions inductively on the structure of visible data (aux), greatest relations (coP), coinductively defined functions (coF) and subdomains of the final (Δ, C)-coalgebra by assertions. The latter occur in a similar form in CCSL specifications [54]. Jacobs introduced assertions in [52] as axioms for unary predicates—we call them **coequalities** (cf. Def. 17.4)—, which are invariant with respect to the application of destructors and thus equip the subcarriers of coalgebra elements satisfying the assertions with their own coalgebra structure. CCSL assertions are arbitrary first-order formulas over the given cosignature. Sometimes rather complicated closure constructions are necessary for ensuring their invariance with respect to destructors [52]. Our assertions are confined to co-Horn clauses and combined with invariance axioms (see below), which ensures that a final coalgebra satisfying the assertions always exists (cf. Thm. **??**(3)).

The terminal constraints of [101], the destructor specifications of [15] and the (Ω, Ξ)-specifications of [49] represent classes of cospecifications. [15] admits only certain conditional equations, called coequations, as axioms and forbids functional codomains of destructors. The axioms of an (Ω, Ξ)-specification may be arbitrary first-order formulas, but the destructors must be linear, though they may have functional codomains. Such codomains occur also in other approaches (cf., e.g., [18, 34]). They allow us to formalize *parameterized* observations and come quite naturally with the "cofreeness" of final coalgebras. [15] restricts the codomains to non-nested sums, but seems to be the only paper so far where final coalgebras with non-linear destructors are characterized in terms of context interpretations.

other notions of coequations in [42], [62], [114] ??

Instead of presenting a language like CCSL [54] for specifying everything coalgebraically we show in the following section how to incorporate a cospecification CSP into a swinging type SP such that the final model of CSP is isomorphic to the final model of the *domain completion* of SP (Def. 18.4).

In terms of object-oriented design, s-assertions are the **invariants** of the class denoted by s. Only coalgebraic specifications allow us to present subdomains described by invariants directly. From an *algebraic* specification SP whose standard model M consists of terms a particular subdomain can be obtained only indirectly, namely by constructing a **reduct** of M. A reduct is determined by a set F of functions of SP: the F-reduct of M contains exactly those elements of M that can be generated by applications of F. On the other hand, reducts of final coalgebras do not make sense because uncountable subdomains could never be specified in this way. Instead, the reduct construction will be integrated into the notion of a dialgebraic swinging type given in the next section.

In closing this section, let us point out a striking duality between algebraic and coalgebraic specifications:

ALGEBRAIC SPECIFICATIONS ARE FUNCTION-ORIENTED.

A SIGNATURE Σ defines functions on objects that are TERMS consisting of CONSTRUCTORS.

☞ *The standard model of Σ is an initial algebra of terms, i.e., a sum of products.*

The axioms of a SPECIFICATION SP with signature Σ IDENTIFY objects and specify functions INDUCTIVELY.

☞ *The standard model of SP is a* QUOTIENT *of the standard model of Σ.*

COALGEBRAIC SPECIFICATIONS ARE OBJECT-ORIENTED.

A COSIGNATURE Δ defines object states as interpretations of CONTEXTS consisting of DESTRUCTORS.

☞ *The standard model of Δ is a final coalgebra of object states, i.e., a product of sums.*

The axioms of a COSPECIFICATION CSP with cosignature Δ SELECT objects and specify functions COINDUCTIVELY.

☞ *The standard model of CSP is a* SUBCOALGEBRA *of the standard model of Δ.*

A closer look at the standard models reveals further dualities between algebraic and coalgebraic specifications:

Σ	signature	Δ	cosignature
$I = Ini(\Sigma)$ consisting of terms	initial Σ-algebra	$F = Fin(\Delta, C)$ consisting of context interpretations (cf. Fig. ??)	final (Δ, C)-coalgebra
μAX	Horn clauses	$coAX$	co-Horn clauses
$SP = (\Sigma, \mu AX \cup \dots)$	algebraic swinging type	$CSP = (\Delta, C, \dots, coAX \cup \dots)$	cospecification
$EMod(SP)$ $= \{A \in Mod(SP) \mid \forall\ s \in S : \equiv_s^A = \Delta_s^A\}$		$Mod(CSP)$ $= \{A \in pMod(CSP) \mid \forall\ s \in hidS : is_s^A = s^A\}$	
$I \overset{ini}{\to} A$ $kernel(ini) \models \mu AX$ $\equiv^I =_{def}$ least subset of $I \times I$ s.t. $I \models \mu AX$		$A \overset{fin}{\to} F$ $image(fin) \models coAX$ $is^F =_{def}$ greatest subset of F s.t. $F \models coAX$	
$\Longrightarrow \quad \equiv^I \subseteq kernel(ini)$		$\Longrightarrow \quad image(fin) \subseteq is^F$	
$\Longleftrightarrow \quad ini$ is compatible with $\equiv$		$\Longleftrightarrow \quad fin$ is compatible with is	
$\Longrightarrow \left\{ \begin{array}{ccccc} I & \overset{nat}{\to} & I/{\equiv^I} & \overset{ini'}{\to} & A \\ & & [t] & \mapsto & ini(t) \end{array} \right.$	ini' is well-defined	$\Longrightarrow \left\{ \begin{array}{ccccc} A & \overset{fin'}{\to} & is^F & \overset{inc}{\to} & F \\ a & \mapsto & fin(a) & & \end{array} \right.$	fin' is well-defined
$\overset{I \in Mod(SP)}{\Longrightarrow} \quad I/{\equiv^I}$ is initial in $EMod(SP)$		$\overset{F \in pMod(CSP)}{\Longrightarrow} \quad is^F$ is final in $Mod(CSP)$	

Another dimension of the duality between initial and final models is the topic of [10]. In the table above, we construct quotients on the left-hand *initial* side and subdomains on the right-hand *final* side, while [10] confronts *initial* reachability (via constructors) with *final* observability (via behavioral equalities). This reflects the more general duality between the *initial* construction of subdomains (via subsignatures) and the *final* construction of quotients (via greatest fixpoints), respectively.

As proof obligations are concerned, the duality leads to trade-offs. While the final specification of a **subdomain** in terms of assertions is automatically consistent, the initial specification in terms of generating functions requires an inductive proof that the generated objects and only these have the subdomain's desired properties. On the other hand, the initial specification of a **quotient** in terms of equational axioms automatically yields a congruence relation (the structural equivalence), while the final specification in terms of destructors requires the proof that the induced behavioral equality is also compatible with the other functions and relations of the specification.

18 Algebraic types with cospecifications

In this section, we combine a swinging type SP with a cospecification CSP. Certain hidden sorts of SP are declared as *destructor sorts*, which thus become the hidden sorts of CSP. SP is then extended by the final CSP-model insofar as all elements of $Fin(CSP)$ are added to SP as additional constructor constants. Without such an extension swinging types can only represent *finitely* generated data domains where each object has a ground term representation. Final models of a cospecification, however, may have even uncountable carriers.

Definition 18.1 (dialgebraic swinging type) Let $SP = (\Sigma, AX)$ be an algebraic swinging type that is not basic Horn and whose components are named as in Def. 5.1. Suppose that all $hidS$-destructors are functional destructors with a single hidden argument and there are no $hidS$-constructors. Let des denote the set of $hidS$-destructors, $\Delta = (vis\Sigma, hidS, des)$, $C = Fin(visSP)$ and

$$CSP = (\Delta, C, aux, coP, coF, CAX)$$

be a cospecification such that the auxiliary functions of CSP are defined functions of SP and the axioms for aux and νP are the same in SP and CSP. Then

$$CST \;=\; (SP, aux, coF, AX_{hidS\cup coF})$$

is a **dialgebraic swinging type**. SP is the **algebraic part** and CSP is the **cospecification of** CST. A CST**-context** is a Δ-context. ❏

The main syntactic differences between an algebraic swinging type SP and a dialgebraic swinging type CST can be summarized as follows:

- CST has no transitional destructors.
- SP has neither assertions nor cofunctions.

For the later use of a dialgebraic ST as the visible subtype of an algebraic ST, the former is transformed into its *domain completion* (cf. Def. 18.4), which is an algebraic ST. While the hidden sorts of an algebraic ST are interpreted as quotients of term sets, the hidden sorts of a dialgebraic type will be interpreted in the final model of the type's cospecification. Each element of the model denotes a set of behaviors or context interpretations (cf. Def. 17.1). From the applications' point of view the requirement that each destructor for a hidden sort has a single hidden argument (cf. Def. 18.1) is not restrictive. For instance, in object-oriented terminology, the sort s of the hidden argument corresponds to the class whose objects are "observed" by the s-destructors. A destructor never observes several objects simultaneously, unless they are tupled, i.e., attached to a (hidden) product sort whose projections provide the destructors into the component sorts (cf. Section 2). Further arguments of a destructor denote observation *parameters* and thus are always part of a visible subdomain.

Definition 18.2 (contextual equivalence) Let SP be an algebraic functional swinging type. The **Nerode** or **contextual** SP**-equivalence** $\sim_{SP}^{Ner}$ is the S-sorted binary relation on T_Σ that is defined as follows: for all $s \in S$ and $t, t' \in T_{\Sigma,s}$,

$$t \sim_{SP}^{Ner} t' \iff_{def} \begin{cases} nf(t) \sim_{visSP} nf(t') & \text{if } s \in visS, \\ \forall\, e : s \to (w \to s') \in CT_\Delta,\ u \in T_{\Sigma,w} : nf(e(t)(u)) \sim_{visSP} nf(e(t')(u)) & \text{if } s \in hidS. \end{cases}$$ ❏

Lemma 18.3 *Let SP be an algebraic functional and continuous swinging type. Then contextual SP-equivalence agrees with behavioral SP-equivalence.*

Proof. Since $\sim_{SP}$ is the greatest relation on T_Σ that satisfies the behavior axioms of SP (cf. Def. 5.1), $\sim_{SP}^{Ner}$ agrees with $\sim_{SP}$ iff $\sim_{SP}^{Ner}$ is the greatest relation $\approx$ on T_Σ such that for all $s \in S$ and $t, t' \in T_{\Sigma,s}$,

$$t \approx t' \quad \text{implies} \quad \left\{ \begin{array}{ll} t \sim_{SP} t' & \text{if } s \in visS, \\ \forall\, d : sw \to s' \in des,\ u \in T_{\Sigma,w} : d(t,u) \approx d(t',u) & \text{if } s \in hidS. \end{array} \right\} \tag{1}$$

$\sim_{SP}^{Ner}$ is a subrelation of $\sim_{SP}$ if (1) is satisfied by $\approx = \sim_{SP}^{Ner}$. This holds true for $s \in visS$ because then $\sim_{SP,s}^{Ner} \subseteq \equiv_{SP} \circ \sim_{visSP} \circ \equiv_{SP} \subseteq \sim_{SP}$. Let $s \in hidS$, $t \sim_{SP,s}^{Ner} t'$, $d : sw \to s' \in des$ and $u \in T_{\Sigma,w}$. Then for all $e : s' \to (v \to s'') \in CT_\Delta$ and $u' \in T_{\Sigma,v}$,

$$nf(e(d(t,u))(u')) \sim_{visSP} nf((e \cdot d)(t)(u,u')) = nf((e \cdot d)(t')(u,u')) = nf(e(d(t',u))(u')).$$

Hence $d(t,u) \sim_{SP}^{Ner} d(t',u)$, and we conclude that $\approx = \sim_{SP}^{Ner}$ satisfies (1).

For the converse, let Φ be the $coAX$-consequence operator on $Her(SP)|_{\Sigma'}$ (cf. Def. 12.1) and U be the $(\Sigma' \cup coP)$-structure with $U|_{\Sigma'} = Her(SP')$ and $r^U = T_{\Sigma,w}$ for all $r : w \in coP$. By assumption, Φ is continuous. Hence by Kleene's fixpoint theorem (cf., e.g., [89], Thm. 4.2), $Her(SP)_{\Sigma' \cup coP} = \cap_{i \in \mathbb{N}} \Phi^i(U)$.

Let $i\in\mathbb{N}$ and CT^i_Δ be the set of Δ-contexts consisting of at most i destructors. We define approximations of contextual equivalence: for all $s\in S$ and $t,t'\in T_{\Sigma,s}$,

$$t\sim^i t' \iff_{def} \begin{cases} nf(t)\sim_{visSP} nf(t') & \text{if } s\in visS,\\ \forall\ e:s\to s'\in CT^i_\Delta,\ u\in T_{\Sigma,w}: nf(e(t)(u))\sim_{visSP} nf(e(t')(u)) & \text{if } s\in hidS.\end{cases}$$

Suppose that for all $i\in\mathbb{N}$, $s\in S$ and $t,t'\in T_{\Sigma,s}$,

$$\Phi^i(U)\models t\sim t' \quad \text{implies} \quad t\sim^i t'. \tag{2}$$

Then $\sim_{SP}=\cap_{i\in\mathbb{N}}\sim^{\Phi^i(U)}\subseteq\cap_{i\in\mathbb{N}}\sim^i=\sim^{Ner}_{SP}$, and the proof is complete. It remains to show (2).

Let $\Phi^i(U)\models t\sim t'$. If $s\in visS$, then $t\sim_{SP} t'$ and thus $nf(t)\equiv_{SP} t\sim_{SP} t'\equiv_{SP} nf(t')$. Hence $nf(t)\sim_{SP} nf(t')$ and thus $nf(t)\sim_{visSP} nf(t')$ because $\sim_{SP}|_{T_{vis\Sigma}}$ satisfies the behavior axioms of $visSP$ and $\sim_{visSP}$ is the greatest solution of the behavior axioms of $visSP$. Hence $t\sim^i t'$.

Let $s\in hidS$. If $i=0$, then $\sim^{\Phi^i(U)}_s=\sim^U_s=T_{\Sigma,s}\times T_{\Sigma,s}$ and $CT^i_\Delta=\emptyset$. Hence (2) holds true. Let $i>0$. By the definition of Φ,

$$\forall\ d:sw\to s'\in des,\ u\in T_{\Sigma,w}:\Phi^{i-1}(U)\quad\models\quad d(t,u)\sim d(t',u). \tag{3}$$

By induction hypothesis, (3) implies $d(t,u)\sim^{i-1} d(t',u)$. Hence for all $e:s'\to(v\to s'')\in CT^{i-1}_\Delta$ and $u'\in T_{\Sigma,v}$,

$$nf((e\cdot d)(t)(u,u'))=nf(e(d(t,u))(u'))\sim_{visSP} nf(e(d(t,u))(u'))=nf((e\cdot d)(t)(u,u')).$$

Hence $t\sim^i t'$, and the proof of (2) is complete. ❏

Let SP' be the extension of SP by constructor constants denoting the elements of $Fin(CSP)$. The final SP'-model that is given by the quotient of the SP'-Herbrand model by behavioral SP'-equivalence (cf. [89], Def. 4.6) will turn out to be isomorphic to $Fin(CSP)$. Hence the final-*coalgebra* semantics of CST, which is given by $Fin(CSP)$, coincides with the final-*algebra* semantics $Fin(SP')$ of a "sufficiently big" extension of SP.

Definition 18.4 (domain completion) Let CST be a dialgebraic swinging type as in Def. 18.1 such that $visSP$ is functional and head complete (cf. Def. 12.1). We regard the elements of $Fin(CSP)$ as additional (nullary) constructors and the cofunctions of CSP as additional defined functions and add the following sets of axioms for destructors and cofunctions, respectively:

$$\begin{array}{rcll} AX_{des} & = & \{d(a,u)\equiv t\mid & d:sw\to s'\in des,\ s'\in visS,\ a\in Fin(CSP)_s,\\ &&& u,t\in NF_{vis\Sigma},\ t \text{ is an object normal form},\\ &&& d^{Fin(CSP)}(a,[u])=[t]\}^{19}\ \cup\\ && \{d(a,u)\equiv b\mid & d:sw\to s'\in des,\ s'\in hidS,\ a\in Fin(CSP)_s,\\ &&& u\in NF_{vis\Sigma},\ d^{Fin(CSP)}(a,[u])=b\}\end{array}$$

$$\begin{array}{rcll} AX_{coF} & = & \{f(t_1,\dots,t_n)\equiv b\mid & f:s_1\dots s_n\to s\in coF,\ \text{for all } 1\le i\le n,\\ &&& s_i\in visS \text{ implies } t_i\in NF_{vis\Sigma,s_i} \text{ and } t'_i=[t_i],\\ &&& s_i\in hidS \text{ implies } t_i=t'_i\in Fin(CSP)_{s_i},\\ &&& f^{Fin(CSP)}(t'_1,\dots,t'_n)=b\}\end{array}$$

Let $\Sigma'=\Sigma\cup(\emptyset,Fin(CSP)\cup coF,\emptyset)$ and $AX'=AX\cup AX_{des\cup coF}$. Then $SP'=(\Sigma',AX')$ is an algebraic swinging type, called the **domain completion of** CST. A **coinductive theorem of** CST is an inductive theorem of SP' (cf. Def. 12.1).

[19] Since $visSP$ is head complete and functional, [94], Lemma 3.7 implies that all behaviorally $visSP$-equivalent object normal forms are equal. Hence the $\sim_{visSP}$-equivalence $d^{Fin(CSP)}(a,[u])$ contains exactly one object normal form t.

CST is **cospec closed** if CST has no assertions or no $hidS$-constructors or for all $s \in hidS$ and $t \in s^{NF_{\Sigma'}}$ there is $a \in Fin(CSP)_s$ such that $t \sim_{SP'} a$. CST is **functional**, **continuous** or **behaviorally consistent**, respectively, if SP' is functional, continuous or behaviorally consistent, respectively. ❑

Note that AX_{des} need not be the only axioms for $hidS$-destructors in SP'. If SP contains hidden constructors, then these must be specified in terms of axioms for destructors in order to make CST complete and, if CST contains assertions, cospec closed.

SP' is coinductive if SP is coinductive ([89], Def. 6.1; [94], Def. 7.1). This confirms the adequacy of this criterion for behavioral consistency, especially concerning the required form of the left-hand side t of a coinductive equational axiom: the leading function symbol of t is a destructor, while all other symbols of t are constructors or variables. Each of these axioms contributes to the specification of a single destructor. Condition 17.1(1) ensures the "context-free" interpretation of each destructor in the final (Δ, C)-coalgebra. Hence it is not necessary to admit axioms that specify destructors in the context of other destructors (see also [89], Section 6).

Each dialgebraic swinging type has an algebraic part. Conversely, one build algebraic on top of dialgebraic types in the course of developing a specification hierarchically. This happens automatically when the visible subtype of an algebraic type coincides with the domain completion of a dialgebraic type (cf. Def. 5.1).

The domain completion SP' of CST may augment the algebraic part SP of CST with infinitely, maybe uncountably many constants, which denote context interpretations, i.e., elements of $Fin(CSP)$. These constants are and occur in the ground terms $Her(SP')$ consists of. Since $Fin(CSP)$ may be uncountable, while the rest of $Her(SP')$ is countable, $Her(SP')$ may be imagined as an iceberg whose main part hides in the sea (cf. Fig. **??**). In contrast to $Fin(CSP)$, the term structure of $Her(SP')$ allows us to employ inference rules whose applicability relies on term unification. In fact, reasoning about (the domain completion of) CST can be reduced to reasoning about the cospecification of CST:

Theorem 18.5 *Let CST be a cospec closed, functional, continuous and behaviorally consistent dialgebraic swinging type as in Def. 18.1 and $SP' = (\Sigma', AX')$ be the domain completion of CST such that $visSP$ is functional and object constructor complete.*

(1) If CST has no assertions, then for all $s \in hidS$ and $t \in T_{\Sigma',s}$ there is $a \in Fin(CSP)_s$ such that $t \sim_{SP'} a$.

(2) For all $s \in hidS$ and $a, b \in Fin(CSP)_s$, $a \sim_{SP'} b$ iff $a = b$.

(3) $Fin(SP')$ and $Fin(CSP)$ are Σ'-isomorphic.

Proof. Let $\approx$ be the S-sorted relation that is defined as follows. For all $s \in visS$, $\approx_s = \sim_{visSP,s}$. For all $s \in hidS$, $\approx_s$ is the least equivalence relation on $T_{\Sigma',s}$ that includes $\equiv_{SP',s}$ and satisfies

$$t \approx_s t' \wedge u \sim_{visSP,w} u' \quad \Rightarrow \quad d(t,u) \approx_{s'} d(t',u)$$

for all $d : sw \to s' \in des$ with $s' \in hidS$. Of course, $\approx$ is a subrelation of $\sim_{SP'}$.

Let $C = Fin(visSP)$. We define a (Δ, C)-coalgebra A:

- For all $s \in visS$, $s^A = C_s = T_{vis\Sigma,s}/\sim_{visSP,s}$.
- For all $s \in hidS$, $s^A = T_{\Sigma',s}/\approx_s$.
- For all $d : sw \to s' \in des$, $t \in T_{\Sigma',s}$ and $u \in T_{vis\Sigma,w}$, $d^A([t],[u]) = [nf(d(t,u))]$.

d^A is well-defined: Let $t \approx t'$ and $u \sim_{visSP} u'$. If $s' \in visS$, then $d(t,u) \sim_{SP'} d(t',u')$ and thus $nf(d(t,u)) \sim_{SP'} nf(d(t',u'))$. Hence $nf(d(t,u)) \sim_{visSP} nf(d(t',u'))$ because $\sim_{SP'}|_{T_{vis\Sigma}}$ satisfies the behavior axioms of $visSP$ and $\sim_{visSP}$ is the greatest solution of the behavior axioms of $visSP$. If $s' \in hidS$, then $d(t,u) \approx d(t',u')$ by the definition of $\approx$. Hence $nf(d(t,u)) \approx nf(d(t',u'))$.

By Thm. 17.3, $Fin(\Delta, C)$ is final in $coAlg(\Delta, C)$. Hence there is a unique (Δ, C)-homomorphism $fin : A \to Fin(\Delta, C)$ that is defined as follows (see the proof of Thm. 17.3): for all $s \in S$ and $t \in T_{\Sigma',s}$,

$$\begin{array}{rcll} fin([t]) & = & [t] & \text{if } s \in visS, \\ \pi_e(fin([t])) & = & e^A([t]) \quad \text{for all } e : s \to s' \in CT_\Delta & \text{if } s \in hidS. \end{array}$$

Let $nat : NF_{\Sigma'} \to A$ be the natural function that maps each $visS$- resp. $hidS$-sorted ground normal form t to the $\sim_{visSP}$- resp. $\approx$-equivalence class $[t]$. The composition $nat \circ nf$ is compatible with $vis\Sigma$ and des: Let $d : sw \to s' \in des$, $t \in T_{\Sigma',s}$ and $u \in T_{vis\Sigma,w}$. Then

$$(nat \circ nf)(d(t,u)) = [nf(d(t,u))] = d^A([t],[u]) = d^A([nf(t)],[nf(u)]) = d^A((nat \circ nf)(t),(nat \circ nf)(u)).$$

Next we show $fin([a]) = a$ for all $a \in Fin(CSP)$. This holds true if for all $e : s \to (w \to s') \in CT_\Delta$, $\pi_e(fin([a])) = \pi_e(a)$, which is proved by induction on the size of e:

Case 1. $e \in des$. Let $u \in T_{vis\Sigma,w}$. Since $s' \in visS$, there is $t \in NF_{vis\Sigma}$ such that $e^{Fin(CSP)}(a,[u]) = [t]$. Hence $(e(a,u) \equiv t) \in AX'$ and thus

$$\pi_e(fin([a]))([u]) = e^A([a],[u]) = [nf(e(a,u))] = [t] = e^{Fin(CSP)}(a,[u]) = \pi_e(a)([u]).$$

Case 2. $e = f \cdot d$ and $w = vv'$ for some $d : sv \to s'' \in des$ and $f : s'' \to (v' \to s') \in CT_\Delta$. Let $u \in T_{vis\Sigma,v}$ and $u' \in T_{vis\Sigma,v'}$. Since $s' \in hidS$, there is $b \in Fin(\Delta, C)$ such that $d^{Fin(CSP)}(a,[u]) = b$. Hence $(d(a,u) \equiv b) \in AX'$ and thus

$$\begin{aligned} &\pi_e(fin([a]))([u],[u']) = e^A([a])([u],[u']) = f^A(d^A([a],[u]))([u']) = f^A([nf(d(a,u))])([u']) \\ &= f^A([b])([u']) = \pi_f(fin([b]))([u']) \overset{ind.hyp.}{=} \pi_f(b)([u']) = \pi_f(d^{Fin(CSP)}(a,[u]))([u']) \\ &= \pi_e(a)([u],[u']) \end{aligned}$$

where the last equation follows from the interpretation of d in $Fin(\Delta, C)$ (cf. Def. 17.1).

We show that the equivalence kernel of $h =_{def} fin \circ nat \circ nf : Her(SP') \to Fin(\Delta, C)$ agrees with contextual SP'-equivalence (cf. Def.18.2). Let $s \in S$ and $t, t' \in T_{\Sigma',s}$. If $s \in visS$, then

$$h(t) = h(t') \iff nf(t) \sim_{visSP} nf(t') \iff t \sim^{Ner}_{SP'} t'.$$

Let $s \in hidS$. Then $h(t) = h(t')$ iff for all $e : s \to (w \to s') \in CT_\Delta$,

$$e^A([t]) = e^A([nf(t)]) = \pi_e(fin([nf(t)])) = \pi_e(fin([nf(t')])) = e^A([nf(t')]) = e^A([t']). \tag{4}$$

We show that for all $s \in hidS$, $t \in T_{\Sigma',s}$, $e : s \to (w \to s') \in CT_\Delta$ and $u \in T_{vis\Sigma,w}$,

$$e^A([t]) = e^A([t']) \iff nf(e(t)(u)) \sim_{visSP} nf(e(t')(u)), \tag{5}$$

by induction on the size of e. If $e \in des$, then $s' \in visS$ and (5) follows from the definition of e^A (see above). Otherwise there are $v, v' \in visS^*$, $s'' \in hidS$, $d : sv \to s'' \in des$ and $f : s'' \to (v' \to s') \in CT_\Delta$ with $e = f \cdot d$ and $w = vv'$. Suppose that $e^A([t]) = e^A([t'])$ for all $u \in T_{vis\Sigma,u}$ and $u' \in T_{vis\Sigma,v'}$. Then

$$\begin{aligned} &f^A([nf(d(t,u))])([u']) = f^A(d^A([t],[u]))([u']) = e^A([t])([u],[u']) = e^A([t'])([u],[u']) \\ &= f^A(d^A([t'],[u]))([u']) = f^A([nf(d(t',u))])([u']) \end{aligned}$$

and thus by induction hypothesis,

$$nf(e(t)(u,u')) = nf(f(d(t,u))(u')) \sim_{visSP} nf(f(d(t',u))(u')) = nf(e(t)(u,u')).$$

For the converse, suppose that $nf(e(t)(u,u')) \sim_{visSP} nf(e(t')(u,u'))$ for all $u \in T_{vis\Sigma,u}$ and $u' \in T_{vis\Sigma,v'}$. Then

$$nf(f(d(t,u))(u')) = nf(e(t)(u,u')) \sim_{visSP} nf(e(t')(u,u')) = nf(f(d(t',u))(u'))$$

and thus by induction hypothesis,

$$\begin{array}{l} e^A([t])([u],[u']) = f^A(d^A([t],[u]))([u']) = f^A([nf(d(t,u))])([u']) \\ = f^A([nf(d(t',u))])([u']) = f^A(d^A([t'],[u]))([u']) = e^A([t'])([u],[u']). \end{array}$$

By (4) and (5), for all $e : s \to (w \to s') \in CT_\Delta$ and $u \in T_{vis\Sigma,w}$,

$$\begin{array}{l} h(t) = h(t') \iff fin(nat(nf(t))) = fin(nat(nf(t'))) \\ \iff \forall\, e : s \to (w \to s') \in CT_\Delta,\ u \in T_{vis\Sigma,w} : nf(e(t)(u)) \sim_{visSP} nf(e(t')(u)) \iff t \sim^{Ner}_{SP'} t'. \end{array}$$

This finishes the proof that the equivalence kernel of h agrees with contextual SP'-equivalence.

(1) Suppose that CST has no assertions. Then $Fin(\Delta, C) = Fin(CSP)$. Hence for all $t \in T_{\Sigma',s}$ there is $a \in Fin(CSP)$ such that $h(t) = a = fin([a]) = fin([nf(a)]) = h(a)$. Since the equivalence kernel of h agrees with $\sim_{SP'}$, t and a are behaviorally SP'-equivalent.

(2) Since the equivalence kernel of h agrees with $\sim_{SP'}$, for all $a, b \in Fin(CSP)$, $a \sim_{SP'} b$ iff $a = fin([a]) = fin([nf(a)]) = h(a) = h(b) = fin([nf(b)]) = fin([b]) = b$.

By Lemma **??**, A is a pre-CSP-model. Since for all $s' \in hidS$, $d : sw \to s' \in des$, $a \in Fin(CSP)_s$ and $u \in T_{vis\Sigma,w}$, $d^A([a],[u]) = [nf(d(a,u))] = [b]$ for some $b \in Fin(CSP)_{s'}$, A has a CSP-submodel with $A'_s = \{[a] \in s^A \mid a \in Fin(CSP)_s\}$ for all $s \in hidS$. Since SP' is consistent, for all $s \in hidS$ and $a, b \in Fin(CSP)_s$, $a \equiv_{SP'} b$ implies $a = b$. Hence A' and $Fin(CSP)$ are Σ_0-isomorphic where $\Sigma_0 = vis\Sigma \cup (hidS, des \cup aux, coP)$.

Let φ be an assertion of CSP. We show that A satisfies φ. Let $f : X \to T_{\Sigma'}$ such that for all $x \in X_{visS}$, $f(x) \in T_{vis\Sigma}$. Since CST is complete and cospec closed, there are $g : X \to T_{\Sigma'}$ and $g' : X \to Fin(CSP)$ such that for all $x \in X_{visS}$, $g(x) = f(x)$ and $g'(x) = [f(x)]$, and for all $x \in X_{hidS}$, $f(x) \sim_{SP'} g(x) = g'(x)$. Since $Fin(CSP)$ is a pre-CSP-model, $Fin(CSP)$ satisfies φ and thus $Fin(CSP) \models_{g'} \varphi$. Since $Fin(CSP)$ and A' are Σ_0-isomorphic and φ is a Σ_0-formula, $Fin(CSP) \models_{g'} \varphi$ implies $A' \models_{nat \circ g} \varphi$ and thus $A \models_{nat \circ g} \varphi$ because φ has no universal quantifiers. Since $\approx$ is a weak Σ'-congruence and φ is poly-modal, [89], Thm. 3.9(a) implies $Her(SP') \models_g \varphi$. Since SP' is behaviorally consistent, [89], Thm. 3.8(3) implies $Her(SP') \models_f \varphi$. Since $\approx$ is a weak Σ'-congruence and φ is poly-modal, [89], Thm. 3.9(a) implies $A \models_{nat \circ f} \varphi$.

This finishes the proof that A satisfies the assertions of CSP. Since A interprets the copredicates of CSP as the greatest solutions of $coAX$ (see the proof of Lemma **??**), we conclude that A is a CSP-model.

By Thm. **??**(3), $Fin(CSP)$ is final in $Mod(CSP)$ and the range restriction of $fin : A \to Fin(\Delta, C)$ to $Fin(CSP)$ is the final morphism. Since for all $s \in hidS$ and $a \in Fin(CSP)_s$, $fin([a]) = a$, fin and thus h are surjective.

Since the equivalence kernel of h agrees with contextual SP'-equivalence, Lemma 18.3 implies that it coincides with behavioral SP'-equivalence, which, by assumption, is a weak Σ'-congruence.

Let $\Sigma_1 = vis\Sigma \cup (hidS, des, \emptyset)$. nf, nat, fin and thus h are Σ_1-homomorphisms. Since h is surjective and the equivalence kernel of h is a weak Σ'-congruence, $Fin(CSP)$ becomes a Σ'-structure and h a Σ'-homomorphism if $\Sigma' \setminus \Sigma_1$ is interpreted as follows:

$$\begin{array}{rcll} a^{Fin(CSP)} & = & a\ (= fin([a]) = h(a)\ ) & \text{for all } s \in hidS \text{ and } a \in Fin(CSP)_s, \\ f^{Fin(CSP)}(h(t)) & = & h(f(t)) & \text{for all function symbols } f : w \to s \in \Sigma \setminus \Sigma_1, \\ h(t) \in r^{Fin(CSP)} & \Leftrightarrow & Her(SP') \models r(t) & \text{for all static predicates } r : w \in \Sigma \setminus \Sigma_1, \\ (h(t), h(u)) \in \delta^{Fin(CSP)} & \Leftrightarrow & \exists\, v : Her(SP') \models u \sim v \wedge \delta(t,v) & \text{for all dynamic predicates } \delta : ws \in \Sigma. \end{array}$$

Indeed, the interpretations are well-defined: Let $f : w \to s \in \Sigma \setminus \Sigma_1$ and $t, t' \in T_{\Sigma',w}$ such that $h(t) = h(t')$. Since the equivalence kernel of h is compatible with f, $h(f(t)) = h(f(t'))$. Let $r : w \in \Sigma \setminus \Sigma_1$ be a static predicate and $t, t' \in T_{\Sigma',w}$ such that $h(t) = h(t')$. Since the equivalence kernel of h is compatible with r, $Her(SP') \models r(t)$ iff $Her(SP') \models r(t')$. Let $\delta : ws \in \Sigma$ be a dynamic predicate, $t, t' \in T_{\Sigma',w}$ and $u, u' \in T_{\Sigma',s}$ such that $h(t) = h(t')$ and $h(u) = h(u')$. Hence $t \sim_{SP'} t'$, $u \sim_{SP'} u'$ and thus

$$\begin{aligned}&\exists\, v : Her(SP') \models u \sim v \wedge \delta(t,v) \quad\Longrightarrow\quad \exists\, v, v' : Her(SP') \models u \sim v \wedge \delta(t',v') \wedge v \sim v' \\ &\Longrightarrow\quad \exists\, v' : Her(SP') \models u' \sim v' \wedge \delta(t',v').\end{aligned}$$

(3) We have shown that h is a Σ'-homomorphism. Since the equivalence kernel of h agrees with $\sim_{SP'}$, $h : Her(SP') \to Fin(CSP)$ induces an injective Σ'-homomorphism $h' : Fin(SP') \to Fin(CSP)$. h' is surjective because h is surjective. Hence h' is an isomorphism. ❑

Under the assumptions of Thm. 18.5, the inductive theory of CST agrees with the coinductive theory of CSP (see Definitions 18.4 and **??**):

Corollary 18.6 *Let CST be a cospec closed, functional, continuous and behaviorally consistent dialgebraic swinging type and SP' be the domain completion of CST such that $visSP$ is functional and head complete. Then for all poly-modal Σ'-formulas φ,*

$$Fin(CSP) \models \varphi \iff Fin(SP') \models \varphi \iff Her(SP') \models \varphi.$$

Moreover, $Fin(CSP)$ is final in the class of reachable Σ'-structures that interpret behavioral equalities as weak Σ'-congruences.

Proof. The statements are direct consequence of Thm. 18.5 and [89], Thm. 5.1(2) and (5). ❑

Basic inference rules that are sound with respect to $Her(SP')$ are discussed in [93]. In addition, Corollary 18.6 tells us that the assertions of CST are inductive theorems of CST and thus may be used as lemmas in proofs of further inductive theorems. Corollary 18.6 also implies that the following unfolding rule for a cofunction $f : w \to s$ is correct with respect to $Her(SP')$:

unfolding of $f \in coF$ $\quad \dfrac{\varphi(d(f(t),u))}{\bigvee_{i=1}^{n}(\varphi(t_i[t/x,u/z]) \wedge \varphi_i[t/x,u/z])} \Updownarrow$

if $\{d(f(x),z) \equiv t_1 \Leftarrow \varphi_1,\ \ldots,\ d(f(x),z) \equiv t_n \Leftarrow \varphi_n\}$ is the coinductive axiomatization of f (cf. Def. 17.4)

A number of proof samples can be found in [85, 86, 90, 93, 94, 91].

19 Examples

Dynamic Data Types and *Labelled Transition Logic* [20, 5] incorporate transition systems as relations into specifications and axiomatize them in terms of Horn clauses, which amount to SOS ("structural operational semantics") rules, the classical syntax of transition system specifications. The logic used for reasoning about dynamic data types is a temporal one. Swinging types go a step further and admit to integrate not only transitions systems (in terms of set-valued functions), but also temporal- and modal-logic operators (in terms of predicates or copredicates; see, e.g., [89], Example 2.7). Set-valued functions are also used for expressing association multiplicities of UML class diagrams (see Example 6.6).

By modeling state transitions in terms of set-valued functions one gets rid of the distinction between compatibility and zigzag compatibility of an equivalence relation $\sim$ with *static* and *dynamic* predicates, respectively (see [89]). For instance, zigzag compatibility of a transition relation $\delta : s \times lab \times s$ amounts to compatibility of its functional counterpart $f : s \to (lab \to set(s))$ defined by $b \in f(a)(x) \Leftrightarrow \delta(a, x, b)$. Indeed $\sim$ is *zigzag compatible* with δ or a *bisimulation*, i.e.

$$\begin{aligned} a \sim_s a' \wedge \delta(a, x, b) \quad &\Rightarrow \quad \exists\, b' : \delta(a', x, b') \wedge b \sim_s b', \\ a \sim_s a' \wedge \delta(a', x, b') \quad &\Rightarrow \quad \exists\, b : \delta(a, x, b) \wedge b \sim_s b', \end{aligned}$$

if and only if $\sim$ is compatible with f, i.e.

$$a \sim_s a' \quad \Rightarrow \quad \forall\, x : f(a)(x) \sim_{set(s)} f(a')(x),$$

where the extension of $\sim_s$ to $\sim_{set(s)}$ is defined as above.

A third alternative for specifiying transition systems—besides zigzag compatible relations and set-valued functions—is used in Maude [16] and CafeOBJ [17]: here transition systems are presented as rewrite rules and interpreted in categories consisting of term (classes) as objects and rewrite steps as morphisms.

The presentation

$$SP \quad = \quad SP_1 \ \textsf{and} \ \ldots \ \textsf{and} \ SP_n \ \textsf{then hidden part}$$

of a swinging type SP indicates that $SP'_1 \cup \cdots \cup SP'_n$ is the visible subtype of SP where $SP'_i = SP_i$ if SP_i is algebraic and SP'_i is the domain completion of SP_i is SP_i is dialgebraic (cf. Def. 5.1). SP itself is algebraic (resp. dialgebraic) if, in the hidden part, the declaration of the hidden sorts is followed directly (!) by a declaration of constructors (resp. destructors).

Example 19.1 (integers) The following specification represents integer numbers as terms constructed from 0, 1, + and -. Hence *int* can be declared as the hidden sort of an algebraic swinging type (cf. Def. 5.1). Since the induced structural equivalence would be too fine for representing the equality of integers, we specify this equality as a behavioral one with three destructors: successor, predecessor and a test on zero.

INT = BOOL `then`

`hidsorts`	int
`constructs`	$0, 1 :\to int$
	$\_ + \_ : int \times int \to int$
	$-\_ : int \to int$
`destructs`	$pred, succ : int \to 1 + int$
	$is0 : 1 + int \to bool$
`vars`	$x, y, z : int$
`axioms`	$succ(0) \equiv (1)$
	$succ(1) \equiv (1 + 1)$
	$succ(x + y) \equiv (y) \quad \Leftarrow \quad succ(x) \equiv ()$
	$succ(x + y) \equiv (z + y) \quad \Leftarrow \quad succ(x) \equiv (z)$
	$succ(-x) \equiv () \quad \Leftarrow \quad pred(x) \equiv ()$
	$succ(-x) \equiv (-y) \quad \Leftarrow \quad pred(x) \equiv (y)$
	$pred(0) \equiv (-1)$
	$pred(1) \equiv ()$
	$pred(x + y) \equiv (y) \quad \Leftarrow \quad pred(x) \equiv ()$
	$pred(x + y) \equiv (z + y) \quad \Leftarrow \quad pred(x) \equiv (z)$
	$pred(-x) \equiv () \quad \Leftarrow \quad succ(x) \equiv ()$
	$pred(-x) \equiv (-y) \quad \Leftarrow \quad succ(x) \equiv (y)$
	$is0(()) \equiv true$
	$is0((x)) \equiv false$

The destructor $is0$ cannot be dropped. Otherwise all int-terms were behaviorally equivalent. ❏

Example 19.2 (infinite sequences) The specification STREAM (cf. Ex. 16.1) is a dialgebraic swinging type and the sort $stream$ denotes an uncountable domain whenever $entry$ is interpreted by a carrier with at least two elements.

Let Δ be the cosignature and CSP be the cospecification of STREAM (cf. Def. 18.1). Since $head : stream \to entry$ and $tail : stream \to stream$ are the $stream$-destructors, for each STREAM-context d there is $n \in \mathbb{N}$ such that $d = head \cdot tail^n : stream \to entry$. Hence the $stream$-carrier of the final CSP-model consists of all infinite sequences over C (cf. Def. 18.1):

$$Fin(CSP)_{stream} \;=\; \prod_{d \in CT_\Delta} C_{entry} \;=\; \prod_{n \in \mathbb{N}} C_{entry} \;=\; [\mathbb{N} \to C_{entry}].$$

The domain completion SP' of STREAM contains the following additional axioms for $head$ and $tail$:

$$\begin{array}{ll} head(g) \equiv g(0) & \text{and} \\ tail(g) \equiv \lambda n.g(n+1) & \text{for all } g : \mathbb{N} \to C_{entry}. \end{array}$$

Since STREAM has no assertions, STREAM is cospec closed. [90], Korollar 6.1.5 (or [94]??), and [89], Thms. 5.15 and 6.5 imply that STREAM is functional, continuous and behaviorally consistent, respectively. Hence by Thm. 18.5, $Fin(SP')$ and $Fin(CSP)$ are isomorphic.

Let G be a ground normal form of sort $entry \to bool$. We extend STREAM by the destructor

$$min : (entry \to bool) \times stream \to 1 + nat,$$

regard the defined functions nth and $nthtail$ of STREAM (cf. Ex. 16.1) as auxiliary functions and the axioms for nth and $nthtail$ as inductive axiomatizations and add the assertions

$$\begin{array}{ll} \exists\, n, s' : (min(G, s) \equiv (n) \wedge nthtail(n+1, s) \equiv s'), & (1) \\ min(g, s) \equiv () \;\;\Rightarrow\;\; g(nth(n, s)) \equiv false, & (2) \\ min(g, s) \equiv (n) \;\;\Rightarrow\;\; g(nth(n, s) \equiv true, & (3) \\ (min(g, s) \equiv (n) \wedge m < n) \;\;\Rightarrow\;\; g(nth(m, s) \equiv false. & (4) \end{array}$$

Let FAIR be the resulting dialgebraic swinging type and CSP be the cospecification of FAIR. The $stream$-carrier of $Fin(CSP)$ consists of all infinite sequences s over C_{entry} that are fair with respect to G, i.e., for infinitely many $n \in \mathbb{N}$, $Fin(CSP)$ satisfies $G(nth(n, s)) \equiv true$.[20] This is accomplished by (1). The assertions specify the function min that is used in (1). Since min cannot be defined in terms of an inductive axiomatization, min must be declared as a destructor. The required completeness of FAIR enforces axioms for min that describe the effect of min on normal forms built up of $stream$-constructors. For instance, the axioms

$$\begin{array}{l} min(g, zip(s, s')) \equiv (2*n+1) \;\;\Leftarrow\;\; min(g, s) \equiv () \wedge min(g, s') \equiv (n), \\ min(g, zip(s, s')) \equiv (2*m) \;\;\Leftarrow\;\; min(g, s) \equiv (m) \wedge min(g, s') \equiv (), \\ min(g, zip(s, s')) \equiv (2*m) \;\;\Leftarrow\;\; min(g, s) \equiv (m) \wedge min(g, s') \equiv (n) \wedge 2*m < 2*n+1, \\ min(g, zip(s, s')) \equiv (2*n+1) \;\;\Leftarrow\;\; min(g, s) \equiv (m) \wedge min(g, s') \equiv (n) \wedge 2*m \geq 2*n+1 \end{array}$$

specify the effect of min on normal forms $zip(s, s')$ (cf. [91], Section 4.1). Although FAIR has assertions and hidden constructors, FAIR is cospec complete, i.e., each $t \in stream^{NF_{\Sigma'}}$ is behaviorally SP'-equivalent to some element of $Fin(CSP)_{stream}$. By using the Horn axioms of SP', this can be shown easily by induction on t. ❏

Example 19.3 (streams) The specification COLIST (cf. Ex. 15.2) is a dialgebraic swinging type and the sort $stream$ denotes an uncountable domain whenever $entry$ is interpreted by a carrier with at least two elements.

[20] [101], Example 8, provides a related specification.

Let CSP be the cospecification of COLIST (cf. Def. 18.1). Since $ht : stream \to 1 + (entry \times stream)$ is the only $stream$-destructor, for each $stream$-context c there is $n \in \mathbb{N}$ such that

$$c \;=\; c(n) \;=_{def}\; \boxed{(id+}^{n}(id+\pi_1)\cdot ht\cdot \boxed{\pi_2)\cdot ht}^{n} \;:\; stream \to \boxed{1+(}^{n}1+entry\boxed{)}^{n},$$

i.e., for all $n \in \mathbb{N}$, $c(n+1) = (id + c(n)\cdot\pi_2)\cdot ht$. Hence the $stream$-carrier of $Fin(CSP)$ consists of all finite or infinite sequences over C_{entry}: Let

$$P \;=\; \prod_{n\in\mathbb{N}}(\boxed{1+(}^{n}1+C_{entry}\boxed{)}^{n}).$$

$Fin(CSP)_{stream}$ is the greatest fixpoint of the function $\Phi : \mathcal{P}(P) \to \mathcal{P}(P)$ that is defined as follows: for all $A \subseteq P$,

$$\begin{aligned}
\Phi(A) \;&=\; \{a \in A \mid \exists\, b \in 1 + (C_{entry} \times A)\ \forall\, n \in \mathbb{N} : \pi_{(id+c(n)\cdot\pi_2)\cdot ht}(a) = \pi_{id+c(n)\cdot\pi_2}(b)\}\\
&=\; \{a \in A \mid \exists\, b \in 1 + (C_{entry} \times A)\ \forall\, n \in \mathbb{N} : \pi_{c(n+1)}(a) = \pi_{id+c(n)\cdot\pi_2}(b)\}\\
&=\; \{a \in A \mid \forall\, n \in \mathbb{N} : \pi_{c(n+1)}(a) = () \ \vee\ \exists\, (x, a') \in C_{entry} \times A\ \forall\, n \in \mathbb{N} : \pi_{c(n+1)}(a) = (\pi_{c(n)\cdot\pi_2}(x, a'))\}\\
&=\; \{a \in A \mid \forall\, n \in \mathbb{N} : \pi_{c(n+1)}(a) = () \ \vee\ \exists\, a' \in A\ \forall\, n \in \mathbb{N} : \pi_{c(n+1)}(a) = (\pi_{c(n)}(a'))\}.
\end{aligned}$$

Hence for all $a \in P$,

$$a \in Fin \iff \forall\, n \in \mathbb{N} : \pi_{c(n+1)}(a) = () \ \vee\ \exists\, a' \in Fin\ \forall\, n \in \mathbb{N} : \pi_{c(n+1)}(a) = (\pi_{c(n)}(a'))$$

and thus

$$Fin \;=\; 1 + C_{entry} \times (1 + C_{entry} \times (1 + C_{entry} \times \ldots)) \;=\; C^*_{entry} \cup [\mathbb{N} \to C_{entry}].$$

The domain completion $SP' = (\Sigma', AX')$ of COLIST contains the following additional axioms for ht:

$$\begin{array}{ll}
ht([]) \equiv (), & \\
ht(x : L) \equiv (x, L) & \text{for all } x \in C_{entry} \text{ and } L \in C^*_{entry},\\
ht(g) \equiv (g(0), \lambda n.g(n+1)) & \text{for all } g : \mathbb{N} \to C_{entry}.
\end{array}$$

Since COLIST has no assertions, COLIST is cospec closed. [90], Korollar 6.1.5 (or [94]??), and [89], Thms. 5.15 and 6.5 imply that SP' is functional, continuous and behaviorally consistent, respectively. Hence by Thm. 18.5, $Fin(SP')$ and $Fin(CSP)$ are isomorphic.

If we extend COLIST by one of the assertions

$$\begin{array}{ll}
(1) & \exists\, x, s' : ht(s) \equiv (x, s'),\\
(2) & (ht(s) \equiv () \vee\\
& \quad \exists\, x, s' : (ht(s) \equiv (x, s') \wedge\\
& \qquad\quad (ht(s') \equiv () \vee\\
& \qquad\quad \exists\, s'' : ht(s') \equiv (x, s'')))),\\
(3) & (ht(s) \equiv () \vee\\
& \quad \exists\, x, s' : (ht(s) \equiv (x, s') \wedge\\
& \qquad\quad (ht(s') \equiv () \vee\\
& \qquad\quad \exists\, y, s'' : (ht(s') \equiv (y, s'') \wedge\\
& \qquad\qquad\quad (ht(s'') \equiv () \vee\\
& \qquad\qquad\quad \exists\, s_1 : (ht(s'') \equiv (x, s_1) \wedge\\
& \qquad\qquad\qquad (ht(s_1) \equiv () \vee\\
& \qquad\qquad\qquad \exists\, s_2 : ht(s_1) \equiv (y, s_2)))))))),
\end{array}$$

respectively, we obtain a dialgebraic type with cospecification CSP such that the *stream*-carrier of $Fin(CSP)$ consists of infinite streams (1), constant streams (2) or alternating streams (3). This example was motivated by the *coequational* specifications of these subcoalgebras given in [15], Section 4. ❏

Example 19.4 (stream comprehension) The following extension of COLIST by a specification of stream comprehension ($filter$) is inspired by [101], Example 8. We declare $filter$, as a cofunction and specify $filter$ in terms of a coinductive axiomatization in the sense of Def. 17.4. As in the previous example, some axioms involve an additional destructor min and auxiliary functions nth and $nthtail$. Again, the given axioms for nth and $nthtail$ form inductive axiomatizations, and min is specified in terms of both Horn axioms and assertions.

FILTER1 = COLIST `then`

`destructs`	$min : (entry \to bool) \times stream \to 1 + nat$
`cofuncts`	$filter : (entry \to bool) \times stream \to stream$
`vars`	$x : entry \quad m, n : nat \quad s, t : stream \quad g : entry \to bool$
`axioms`	axioms for min
`assertions`	$min(g, s) \equiv () \quad \Rightarrow \quad g(nth(n, s)) \equiv false$
	$min(g, s) \equiv (n) \quad \Rightarrow \quad g(nth(n, s) \equiv true$
	$(min(g, s) \equiv (n) \wedge m < n) \quad \Rightarrow \quad g(nth(m, s) \equiv false$
`cofaxioms`	$ht(filter(g, s)) \equiv () \quad \Leftarrow \quad min(g, s) \equiv ()$
	$ht(filter(g, s)) \equiv (x, filter(g, t))$
	$\Leftarrow \quad min(g, s) \equiv (n) \wedge nth(n, s) \equiv (x) \wedge nthtail(n + 1, s) \equiv t$

A coinductive axiomatization (Def. 17.4) should not be confused with a set of coinductive axioms ([89], Def. 6.1; [94], Def. 8.2). By [89], Thm. 6.5, coinductive axioms ensure—together with functionality and continuity—behavioral consistency. The axioms for $filter$ both are coinductive and form a coinductive axiomatization. By Thm. 17.5, the latter establishes a functional interpretation of *odds* and $filter$ in the final CSP-model where CSP is the cospecification of FILTER1.

Example 19.5 (binary trees) The following specification generalizes COLIST and specializes FTREE (cf. [91], Section 4.6). It is also a "swinging" version of the running example in [54] where finite and infinite binary trees are presented in CCSL (cf. Section 6). For LIST, see [91], Section 1.2.

BINTREE = ENTRY($entry$) `and` LIST `then`

`hidsorts`	$tree = tree(entry)$	
`destructs`	$root\&sucs : tree \to 1 + (tree \times entry \times tree)$	
	$size : tree \to 1 + nat$	
`constructs`	$mirror : tree \to tree$	
	$leaf, fill : entry \to tree$	
	$mkTree : tree \times entry \times tree \to tree$	
`defuncts`	$subtree : tree \times list(bool) \to 1 + tree$	
`static preds`	$\_ \in \_ : entry \times tree$	
	$finite : tree$	
	$\Diamond r : tree$	for all static predicates $r : tree$
`copreds`	$infinite : tree$	
	$\Box r : tree$	for all static predicates $r : tree$
`vars`	$x : entry \quad T, T', T_1, T_2 : tree \quad m, n : nat \quad b : bool$	
`Horn axioms`	$root(mirror(T)) \equiv root(T)$	
	$sucs(mirror(T)) \equiv () \quad \Leftarrow \quad sucs(T) \equiv ()$	
	$sucs(mirror(T)) \equiv (mirror(T_2), mirror(T_1)) \quad \Leftarrow \quad sucs(T) \equiv (T_1, T_2)$	
	$size(mirror(T)) \equiv size(T)$	

$$\begin{array}{ll}
& root(leaf(x)) \equiv x \\
& sucs(leaf(x)) \equiv () \\
& size(leaf(x)) \equiv (1) \\
& root(fill(x)) \equiv x \\
& sucs(fill(x)) \equiv (fill(x), fill(x)) \\
& size(fill(x)) \equiv () \\
& root(mkTree(T_1,x,T_2)) \equiv x \\
& sucs(mkTree(T_1,x,T_2)) \equiv (T_1,T_2) \\
& size(mkTree(T_1,x,T_2)) \equiv () \quad \Leftarrow \quad size(T_1) \equiv () \\
& size(mkTree(T_1,x,T_2)) \equiv () \quad \Leftarrow \quad size(T_2) \equiv () \\
& size(mkTree(T_1,x,T_2)) \equiv (m+n+1) \quad \Leftarrow \quad size(T_1) \equiv (m) \ \wedge \ size(T_2) \equiv (n) \\
& subtree(T,[]) \equiv (T) \\
& subtree(T,b:L) \equiv () \quad \Leftarrow \quad subtree(T,L) \equiv () \\
& subtree(T,b:L) \equiv () \quad \Leftarrow \quad subtree(T,L) \equiv (T') \wedge sucs(T') \equiv () \\
& subtree(T,0:L) \equiv (T_1) \quad \Leftarrow \quad subtree(T,L) \equiv (T') \wedge sucs(T') \equiv (T_1,T_2) \\
& subtree(T,1:L) \equiv (T_2) \quad \Leftarrow \quad subtree(T,L) \equiv (T') \wedge sucs(T') \equiv (T_1,T_2) \\
& x \in T \quad \Leftarrow \quad root(T) \equiv x \\
& x \in T \quad \Leftarrow \quad sucs(T) \equiv (T_1,T_2) \wedge x \in T_1 \\
& x \in T \quad \Leftarrow \quad sucs(T) \equiv (T_1,T_2) \wedge x \in T_2 \\
& finite(T) \quad \Leftarrow \quad sucs(T) \equiv () \\
& finite(T) \quad \Leftarrow \quad sucs(T) \equiv (T_1,T_2) \wedge finite(T_1) \wedge finite(T_2) \\
& \Diamond r(T) \quad \Leftarrow \quad r(T) \\
& \Diamond r(T) \quad \Leftarrow \quad sucs(T) \equiv (T_1,T_2) \wedge \Diamond r(T_1) \\
& \Diamond r(T) \quad \Leftarrow \quad sucs(T) \equiv (T_1,T_2) \wedge \Diamond r(T_2) \\
\texttt{assertions} & sucs(T) \equiv () \ \Rightarrow \ size(T) \equiv (1) \\
& (sucs(T) \equiv (T_1,T_2) \wedge size(T_1) \equiv ()) \ \Rightarrow \ size(T) \equiv () \\
& (sucs(T) \equiv (T_1,T_2) \wedge size(T_2) \equiv ()) \ \Rightarrow \ size(T) \equiv () \\
& (sucs(T) \equiv (T_1,T_2) \wedge size(T_1) \equiv (m) \wedge size(T_2) \equiv (n)) \ \Rightarrow \ size(T) \equiv (m+n+1) \\
\texttt{axioms} & infinite(T) \quad \Rightarrow \quad (sucs(T) \equiv () \ \Rightarrow \ False) \\
& infinite(T) \quad \Rightarrow \quad (sucs(T) \equiv (T_1,T_2) \ \Rightarrow \ infinite(T_1) \vee infinite(T_2)) \\
& \Box r(T) \quad \Rightarrow \quad r(T) \\
& \Box r(T) \quad \Rightarrow \quad (sucs(T) \equiv (T_1,T_2) \ \Rightarrow \ \Box r(T_1) \wedge \Box r(T_2))
\end{array}$$

The set of BINTREE-contexts is the smallest set CT of coterms such that all $tree$- or $(tree \times tree)$-destructors belong to CT and

$$\begin{array}{rcl}
d : tree \to entry \in CT & \Longrightarrow & d \cdot \pi_1, d \cdot \pi_2 : tree \times tree \to entry \in CT, \\
d : tree \times tree \to entry \in CT & \Longrightarrow & (id + d) \cdot sucs : tree \to 1 + entry \in CT.
\end{array}$$

Let CSP be the cospecification of BINTREE. The $tree$-carrier of $Fin(CSP)$ consists of all finite or infinite binary trees over C_{entry}: Let

$$P \ = \ \prod_{c:tree \to s \in CT} C_s.$$

$Fin(CSP)_{tree}$ is the greatest fixpoint of the function $\Phi : \mathcal{P}(P) \to \mathcal{P}(P)$ that is defined as follows: for all $A \subseteq P$,

$$\Phi(A) \ = \ \{a \in A \mid \exists\, b \in A^2 \ \forall\, d = (id + e) \cdot sucs \in CT : \pi_d(a) = (\pi_e(b)) \ \wedge \ A \models_{a/T} AX_{tree}\text{[21]}\}.$$

[21] AX_{tree} is the set of assertions of BINTREE.

The domain completion $SP' = (\Sigma', AX')$ of BINTREE contains the following additional axioms for each partial function $g : \{0,1\}^* \to C_{entry}$ with a binary-tree domain D_g, i.e., for all $w \in \{0,1\}^*$ and $i \in \{0,1\}^*$, $wi \in D_g$ implies $w, w(i+1 \; mod \; 2) \in D_g$:

$$\begin{aligned} root(g) &\equiv g(\varepsilon), \\ sucs(g) &\equiv \begin{cases} (\lambda w.g(0w), \lambda w.g(1w)) & \text{if } g(0), g(1) \in D_g, \\ () & \text{otherwise,} \end{cases} \\ size(g) &\equiv \begin{cases} (|D_g|) & \text{if } D_g \text{ is finite,} \\ () & \text{otherwise.} \end{cases} \end{aligned}$$

In fact, each g represents exactly one element of $Fin(CSP)_{tree}$.

Although BINTREE has assertions and hidden cofunctions, BINTREE is cospec closed, i.e., each $t \in tree^{NF_{\Sigma'}}$ is behaviorally SP'-equivalent to some element of $Fin(CSP)_{tree}$. By using the Horn axioms of SP', this can be shown easily by induction on t. [90], Korollar 6.1.5 (or [94]??), and [89], Thms. 5.15 and 6.5 imply that SP' is functional, continuous and behaviorally consistent, respectively. Hence by Thm. 18.5, $Fin(SP')$ and $Fin(CSP)$ are isomorphic.

n *tree*-constructor constants $c_1, \dots, c_n :\to tree$ together with the axioms

$$\begin{aligned} root(c_1) \equiv t_1, \quad & sucs(c_1) \equiv (d_1, e_1), \\ & \dots \\ root(c_n) \equiv t_n, \quad & sucs(c_n) \equiv (d_n, e_n) \end{aligned}$$

represent a binary *graph* with n nodes if for all $1 \le i \le n$, $d_i, e_i \in \{c_1, \dots, c_n\}$ and t_i is an *entry*-sorted ground term. c_i denotes a graph with root node t_i, left subgraph d_i and right subgraph e_i (see also Section 3.3).

The following formulas taken from [54] are inductive theorems of BINTREE (cf. Ex. 19.5). They can be derived by employing assertions of BINTREE and inference rules given in Section 2.

$$mirror(mirror(t)) \sim t, \tag{1}$$
$$size(fill(x)) \equiv (), \tag{2}$$
$$x \in fill(y) \;\Rightarrow\; x \equiv y, \tag{3}$$
$$x \in t \;\Leftrightarrow\; x \in mirror(t), \tag{4}$$
$$finite(t) \;\Leftrightarrow\; \exists\, n : size(t) \equiv (n), \tag{5}$$
$$infinite(t) \;\Leftrightarrow\; size(t) \equiv (). \quad ❑ \tag{6}$$

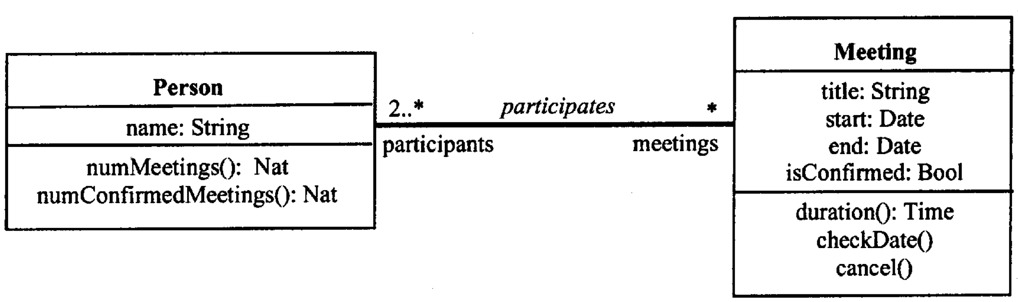


Figure 5. *Two associated classes (Figure 3 of [50]).*

Example 19.6 (UML class specifications) For dealing with object collections like the *meetings* and *participants* in Fig. 5 we refer to an algebraic swinging type FINSET of finite sets:

FINSET

`hidsorts`	$set = set(entry)$
`constructs`	$\emptyset :\to set$
	$\{\_\} : entry \to set$
	$\_ \cup \_ : set \times set \to set$
`destructs`	$in : entry \times set \to bool$
`defuncts`	$remove : entry \times set \to set$
	$mkset : entry^* \to set$
	$filter : (entry \to bool) \times set \to set$
	$\|\_\| : set \to nat$
	$forall : (entry \to bool) \times set \to bool$
	$flatten : set(set) \to set$
`vars`	$x, y, x_1, \dots, x_n : entry \quad s, s' : set \quad f : entry \to entry' \quad g : entry \to bool$
`axioms`	$in(x, \emptyset) \ \equiv \ false$
	$in(x, \{y\}) \ \equiv \ eq(x, y)$
	$in(x, s \cup s') \ \equiv \ in(x, s) \ or \ in(x, s')$
	$remove(x, s) \ \equiv \ s \setminus \{x\}$
	$mkset(()) \ \equiv \ \emptyset$
	$mkset((x_1, \dots, x_n)) \ \equiv \ \{x_1\} \cup \dots \cup \{x_n\}$
	$filter(g, \emptyset) \ \equiv \ \emptyset$
	$filter(g, \{x\}) \ \equiv \ \emptyset \quad \Leftarrow \quad g(x) \equiv false$
	$filter(g, \{x\}) \ \equiv \ \{x\} \quad \Leftarrow \quad g(x) \equiv true$
	$filter(g, s \cup s') \ \equiv \ filter(g, s) \cup filter(g, s')$
	$\|\emptyset\| \ \equiv \ 0$
	$\|\{x\}\| \ \equiv \ 1$
	$\|s \cup s'\| \ \equiv \ \|s \setminus s'\| + \|s' \setminus s\|$
	$forall(g, \emptyset) \ \equiv \ true$
	$forall(g, \{x\}) \ \equiv \ g(x)$
	$forall(g, s \cup s') \ \equiv \ forall(g, s) \ and \ forall(g, s')$
	$flatten(\emptyset) \ \equiv \ \emptyset$
	$flatten(\{s\}) \ \equiv \ s$
	$flatten(s \cup s') \ \equiv \ flatten(s) \cup flatten(s')$

FINSET only specifies the functions used in the class specification given below. For more set functions and also predicates as well as other set specifications, see [91], Section 1.2.2.

The class diagram of Fig. 5 is represented as a dialgebraic swinging type whose axioms cover the multiplicity constraints of Fig. 5 and the following *OCL constraint* [110] taken from [50]:

context Meeting :: checkDate()
  post : isConfirmed =
    self.participants ->
    collect(meetings) ->
    forAll(m | m <> self and m.isConfirmed implies
      (after(self.end,m.start) or (after(m.end,self.start)))

PERSON&MEETING = FINSET `and` STRING `and` DATE&TIME `then`

`hidsorts`	$Person \quad Meeting$
`destructs`	$name : Person \to String$

	$meetings : Person \to Meeting^*$
	$title : Meeting \to String$
	$participants : Meeting \to Person^*$
	$start, end : Meeting \to Date$
	$isConfirmed : Meeting \to bool$
`constructs`	$checkDate : Meeting \to Meeting$
	$cancel : Meeting \to Meeting$
`defuncts`	$Meetings : Person \to set(Meeting)$
	$Participants : Meeting \to set(Person)$
	$numMeetings : Person \to nat$
	$numConfirmedMeetings : Person \to nat$
	$duration : Meeting \to Time$
	$consistent : Meeting \times Meeting \to bool$
`vars`	$p : Person \quad m, m' : Meeting \quad ms : set(Meeting) \quad ps : set(Person)$
`axioms`	$Meetings(p) \equiv mkset(meetings(p))$
	$Participants(m) \equiv mkset(participants(m))$
	$numMeetings(p) \equiv \lvert Meetings(p) \rvert$
	$numConfirmedMeetings(p) \equiv \lvert filter(isConfirmed, Meetings(p)) \rvert$
	$duration(m) \equiv end(m) - start(m)$
	$consistent(m, m') \equiv not(isConfirmed(m'))$
	$\quad or\ end(m) < start(m')\ or\ end(m') < start(m)$
	$isConfirmed(checkDate(m)) \equiv forall(\lambda m'.consistent(m, m'), remove(m, ms))$
	$\quad \Leftarrow Participants(m) \equiv ps \wedge flatten(map(Meetings, ps)) \equiv ms$
	$isConfirmed(cancel(m)) \equiv false$
`assertions`	$\lvert Participants(m) \rvert \geq 2$

Classes come as hidden sorts, attributes and roles as destructors, roles usually as non-linear ones. Basic methods are declared as constructors, derived ones as defined functions. Let CSP be the cospecification of PERSON&MEETING. Similarly to visualizing the elements of of $Fin(\text{BINTREE})_{tree}$ as finite and infinite binary trees (cf. Ex. 19.5), the elements of $Fin(CSP)_{Person}$ and $Fin(CSP)_{Meeting}$ may be regarded as infinite trees whose edges represent the relation between object states that is induced by the *participates* association of Fig. 5. The UML semantics of Fig. 5 requires sets rather than sequences as values of $meetings$ and $participants$. This is reflected by the fact that all axioms of PERSON&MEETING do not use these destructors directly, but only their set versions $Meetings$ and $Participants$.

More details about the presentation of UML classes as swinging types can be found in [91], Section 6.1. ❏

20 Conclusion

Swinging types were introduced in detail in [89]. The paper at hand is devoted to extensions that allow us to use the approach for specifying and reasoning about uncountable data domains, usually given as sets of "infinite" or "lazy" data structures like streams, processes, etc. Traditionally, such structures were treated formally within lattice or order theories. This brought forth many contributions to the model theory of basic computational structures as well as high-level programming language constructs. However, the theory of abstract data types did not gain much from those achievements, probably because the variety of data schemas and appropriate axiomatizations to be considered is much greater here than in classical language theory. Only the evolution of coalgebra theory led to sufficiently general notions for modelling domains with uncountably many elements.

Most of the work on coalgebras and its application to software specification has been done in category-theoretical settings (see, e.g., [105, 104, 56, 10], and the proceedings of already four CMCS workshops[22]) The more restricted notion of a Σ-*coalgebra* as a set of unary functions into a sum sort, now called destructors, already occurs, e.g., in [69]. First approaches to the specification of coalgebras or other behavioral models can be found in [100, 101, 68, 34, 33, 15, 8, 49, 102, 52, 54]. Most of these approaches concentrate on (co)signatures with only linear destructors. Only [15] provides a coterm characterization of final coalgebras with non-linear destructors (see also Section 4). However, the way axiomatic coalgebra presentations are built upon coterms is far from traditional logics. At least, the examples of [15] can also be specified in terms of much-easier-to-comprehend (first-order) assertions (cf. Example 19.3).

Apart from pointing out certain model-theoretic dualities, previous approaches lack an integration of algebraic and coalgebraic types that is suffiently general to cope with "real-world" system models. This is achieved by swinging types, mainly because of their stepwise constructability that allows us to both extend an algebraic basis by coalgebraic components and, conversely, build algebraic structures on top of coalgebraic ones. The following case analysis should provide rough guidelines for the stepwise design of a swinging type:

- **All sorts** s of a type T to be specified **are** *freely generated*, i.e., all s-data can be represented uniquely by ground terms over a finite set of constructors. Then there will be a functional basic Horn specification (cf. Def. 5.1) whose initial and final model coincide with T. *Example:* finite lists of natural numbers (cf. Examples 14.1, 14.2).
- The type is to be **extended by a** *permutative* **sort** s, i.e., all s-data can be represented by ground terms over a finite set of constructors, but the representation is not unique. Then there will be destructors or transition predicates such that the induced behavioral s-equivalence captures the desired identification of data. *Examples:* naturals with ∞ (cf. Ex. 15.1), integers (cf. Ex. 19.1), finite sets, bags or maps (cf. Ex. 16.2).
- The type is to be **extended by an** *infinite* **sort** s, i.e., some s-data may be represented by ground terms over a finite set of constructors, but most s-data will not because they represent "infinite" structures taken from an uncountable domain D. Then there will be destructors and assertions such that the final model of the induced cospecification CSP coincides with D. *Examples:* streams (cf. Exs. 16.1, 15.2, 19.2, 19.3, 19.4), trees (cf. Ex. 19.5). Besides CSP, the resulting dialgebraic ST has an algebraic part for taking into account the term representations of s-data.
- The type is to be **extended by a freely generated sort** s. Then there will be s-object constructors such that the induced behavioral s-equivalence coincides with structural s-equivalence (cf. [94], Lemma 3.8).

Destructors determine the behavioral equivalence. In the case of an infinite sort s, s-destructors also determine the elements of the final model. s-Constructors do so only in the case of a freely generated sort s. In the case of a permutative or an infinite sort s, the decision whether a function should be declared as a constructor or a defined function depends on more or less technical requirements like functionality, coinductivity ([89], Def. 6.1; [94], Def. 8.2) and cospec closedness (Def. 18.4).

Dialgebraic types offer a third possibility, namely to declare a function f as a cofunction. This is suggested at least when f cannot be specified inductively on normal forms. Cofunctions always map into hidden sorts (cf. Ex. 19.4). Non-inductively-specifiable functions on infinite sorts that map into visible sorts must be declared as (additional) destructors and specified in terms of assertions (cf. Exs. 19.4, 19.5). All functions used in assertions must be destructors or auxiliary functions (particular defined functions; cf. Def. 17.4).

If compared with an algebraic ST, the main additional features of a dialgebraic ST are assertions and cofunctions. Assertions are the *invariants* of a dialgebraic type. They allow us to specify subdomains of an uncountable domain. The algebraic counterpart of assertions are *generating functions* that define a subdomain

[22] Coalgebraic Methods in Computer Science 1998-2001, published electronically in Elsevier's ENTCS series.

as the set of all ground terms built up from them. *Such* a specification of a subdomain also works for dialgebraic types, however, the domain is term-generated and thus cannot be uncountable, like, for instance, the set of all alternating streams (cf. Ex. 19.4).

On the other hand, dialgebraic types do not admit transition predicates as destructors. This lack is remedied easily by turning the dialgebraic ST into its (algebraic) domain completion and using this type as the visible subtype of an algebraic type with transition predicates on "copies" of the infinite sorts specified by the dialgebraic ST (see the end of Section 3.4). In this way, we can present uncountable domains equipped with *bisimilarities*, i.e., behavioral equivalences induced by transition predicates.

21 Appendix: Category-theoretical foundations

Algebra may be understandable and applicable without knowing the basics of category theory. Coalgebra and its dual nature in comparison with algebra is rooted in category theory. Hence the knowledge of fundamental constructions and ways of reasoning in category theory are crucial for "getting the point" of dialgebraic modeling (cf., e.g., [65, 4, 70, 97, 104, 56, 41, 60, 2, 55]).

products and sums, equalizers and coequalizers, pullbacks and pushouts, limits and colimits

Definition 21.1 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 $\alpha : F(A) \to A$. $\mathbf{Alg}_F$ denotes the category of F-algebras. An $Alg(F)$**-morphism** h from $\alpha : F(A) \to A$ to $\beta : F(B) \to B$ is a $\mathcal{K}$-morphism $h : A \to B$ with $h \circ \alpha = \beta \circ F(h)$. If α is the initial F-algebra, then h is called a **catamorphism** [79] and (because of its existence and uniqueness) said to be **defined by induction**.

An F**-coalgebra** or F**-codynamics** is a $\mathcal{K}$-morphism $\alpha : A \to F(A)$. $\mathbf{coAlg}_F$ denotes the category of F-coalgebras. A $coAlg(F)$**-morphism** h from $\alpha : A \to F(A)$ to $\beta : B \to F(B)$ is a $\mathcal{K}$-morphism $h : A \to B$ with $F(h) \circ \alpha = \beta \circ h$. If β is the final F-coalgebra, then h is called an **anamorphism** [79] and (because of its existence and uniqueness) said to be **defined by coinduction**.

Do initial/final objects exist if SP satisfies 5.1(1) resp. (2)? This depends on conditions on $\mathcal{K}$ and F. The main conditions and fixpoint constructions are category-theoretic generalizations of notions and results from set theory, in particular of, Kleene's fixpoint theorem for monotone functions. Given a set ("universe") U, the powerset $\mathcal{P}(U)$ extends to a category whose objects are the subsets of U and whose morphisms are set inclusions. The identity morphism is set equality. $\emptyset$ is the initial, U the final object. The union of sets is their colimit, the intersection is their limit. A function $F : \mathcal{P}(U) \to \mathcal{P}(U)$ is monotone iff F is a functor. Hence, if F is monotone, then least fixpoints of F are initial F-algebras, while greatest fixpoints are final F-coalgebras. Ascending chains $A_0 \subseteq A_1 \subseteq A_2 \subseteq \ldots$ and descending chains $B_0 \supseteq B_1 \supseteq B_2 \supseteq \ldots$ in $\mathcal{P}(U)$ become diagrams in an arbitrary category $\mathcal{K}$:

$$A_0 \to A_1 \to A_2 \to \ldots \qquad\qquad B_0 \leftarrow B_1 \leftarrow B_2 \leftarrow \ldots$$

These correspondences between $\mathcal{P}(U)$ and a category $\mathcal{K}$ suggest the following definitions: given an endofunctor F on $\mathcal{K}$, a **fixpoint of** F is a $\mathcal{K}$-object A with $F(A) \cong A$. If $\alpha : F(A) \to A$ is initial in $Alg(F)$, then A is a fixpoint of F, called the **least fixpoint** of F. Analogously, if $\alpha : A \to F(A)$ is final in $coAlg(F)$, then A is a fixpoint of F, called the **greatest fixpoint** of F. F is **continuous** if F preserves colimits of ascending chains in $\mathcal{K}$. F is **cocontinuous** if F preserves limits of descending chains.

Let $\mathcal{K} = \mathcal{P}(U)$. If F is continuous, then Kleene's fixpoint theorem provides us with the least fixpoint $\cup_{i\in\mathbb{N}}F^i(\emptyset)$. If F is cocontinuous, we obtain the greatest fixpoint $\cap_{i\in\mathbb{N}}F^i(U)$. In fact, both fixpoint constructions can be "lifted" to other categories:

Theorem 21.2 (Kleene's fixpoint theorem for *functors*; cf., e.g., [65]) *Suppose that $\mathcal{K}$ has an initial object I and colimits of ascending chains. Let $F : \mathcal{K} \to \mathcal{K}$ be an continuous functor and A be the colimit of the chain $I \to F(I) \to F^2(I)_{-}|_{\dots}$. Then $F(A)$ is the colimit of $F(I) \to F^2(I) \to F^3(I) \to \dots$ and the unique $\mathcal{K}$-morphism $F(A) \to A$ is initial in $Alg(F)$.*

Suppose that $\mathcal{K}$ has a final object T and limits of descending chains. Let $F : \mathcal{K} \to \mathcal{K}$ be a cocontinuous functor and B be the limit of the chain $T \leftarrow F(T) \leftarrow F^2(T) \leftarrow \dots$. Then $F(B)$ is the limit of $F(T) \leftarrow F^2(T) \leftarrow F^3(T) \leftarrow \dots$ and the unique $\mathcal{K}$-morphism $B \leftarrow F(B)$ is final in $coAlg(F)$. ❏

Given a set S sorts, let Set^S be the category of S-sorted sets. Morphisms are S-sorted functions. The initial object of Set^S is the **empty S-sorted set** 0: for all $s \in S$, 0_s is the empty set. The final object is the **one-element S-sorted set** 1: for all $s \in S$, 1_s is a singleton.

Theorem 21.3 (cf., e.g., [4]) *All endofunctors on Set^S that are built up from the identity functor, constant functors, coproducts and finite products are continuous. All endofunctors on Set^S that are built up from the identity functor, constant functors, coproducts and products are cocontinuous.* ❏

Functors built up from the identity functor, constant functors, coproducts and products are called **polynomial**. Theorems 21.2 and 21.3 ensure that polynomial functors have greatest fixpoints.

The powerset functor sending a set A to its powerset $\mathcal{P}(A)$ cannot have a fixpoint: Assume that $f : A \to \mathcal{P}(A)$ has an inverse $f^{-1} : \mathcal{P}(A) \to A$ and define $B = \{a \in A | \ a \notin f(a)\}$. Then we obtain the contradiction $f^{-1}(B) \in B \Leftrightarrow f^{-1}(B) \notin f(f^{-1}(B)) = B$. However, the finite-powerset functor $\mathcal{P}_f$ that maps a set to the set of its finite subsets has final coalgebras because it is *bounded*:

Definition 21.4 An endofunctor F on Set^S is **bounded** by some cardinal κ if for all F-coalgebras $\alpha : A \to F(A)$ and all $a \in A$ there is a subcoalgebra $\beta : B \to F(B)$ of α such that $a \in B$ and $|B| \leq \kappa$. ❏

Theorem 21.5 ([43], Theorem 3.5) *Let F be a functor bounded by κ and $\{\alpha_i\}_{i \in I}$ be the family of all F-coalgebras with cardinality at most κ. The colimes of all homomorphisms with domain $\coprod_{i \in I} \alpha_i$ is final in $coAlg(F)$.* ❏

Example 21.6 Given a set L with cardinality κ, define a functor F on Set by $F(A) = \mathcal{P}_f(A)^L$. F-coalgebras are usually called **image finite labelled transition systems**. F is bounded: Given an F-coalgebra $\alpha : A \to F(A)$ and $a \in A$, let $\langle a \rangle$ be the smallest subcoalgebra of A that contains a. $\langle a \rangle$ consists of all elements $b \in A$ that are α**-reachable** from a, i.e., there are $a = a_1, \dots, a_n = b \in A$ such that $a_1 = a$, $a_n = b$ and for all $1 \leq i < n$ there is $c \in L$ such that $a_{i+1} \in \alpha(a_i)(c)$. Since $\alpha(a_i)(c)$ is finite, b can be represented uniquely by a word w_c over $L \times \mathbb{N}$. Hence $\langle a \rangle$ is bounded by the cardinality of $(L \times \mathbb{N})^*$, which is given by $\sum_{i \in \mathbb{N}} (\kappa * \omega)^i$. ❏

Initiality generalizes to freeness, finality generalizes to cofreeness:

Definition 21.7 Let F be an endofunctor on Set^S and X be an S-sorted set.

An F-algebra $\alpha : F(A) \to A$ together with a map $\eta_A : X \to A$ is **free over** X if for all F-algebras $\beta : F(B) \to B$ and all maps $f : X \to B$ there is a unique $Alg(F)$-morphism $f^* : A \to B$ with $f^* \circ \eta_A = f$.

An F-coalgebra $\alpha : A \to F(A)$ together with a map $\varepsilon_A : A \to X$ is **cofree over** X if for all F-coalgebras $\beta : B \to F(B)$ and all maps $f : B \to X$ there is a unique $coAlg(F)$-morphism $f^* : B \to A$ with $\varepsilon_A \circ f^* = f$. ❏

Theorem 21.8 *Let F be an endofunctor on Set^S. An F-algebra $\alpha : F(A) \to A$ together with a map $\eta_A : X \to A$ is free over X iff the unique coproduct extension $[\eta_A, \alpha] : X + F(A) \to A$ of η_A and α is the initial $X + F(-)$-algebra. In particular, if the initial F-algebra exists, it is the free F-algebra over the empty S-sorted set.*

An F-coalgebra $\alpha : A \to F(A)$ together with a map $\varepsilon_A : A \to X$ is cofree over X iff the unique product extension $\langle \varepsilon_A, \alpha \rangle : A \to X \times F(A)$ of ε_A and α is the final $X \times F(-)$-coalgebra. In particular, if the final

F-coalgebra exists, it is the cofree F-coalgebra over the one-element S-sorted set. ❏

References

[1] P. Aczel, *An Introduction to Inductive Definitions*, in: J. Barwise, ed., Handbook of Mathematical Logic, North-Holland (1977) 739-781

[2] J. Adamek, *Introduction to Coalgebra*, Theory and Applications of Categories 14 (2005) 157-199

[3] M.A. Arbib, E.G. Manes, *Arrows, Structures, Functors: The Categorical Imperative*, Academic Press 1975

[4] M.A. Arbib, E.G. Manes, *Parametrized Data Types Do Not Need Highly Constrained Parameters*, Information and Control 52 (1982) 139-158

[5] E. Astesiano, M. Broy, G. Reggio, *Algebraic Specification of Concurrent Systems*, in [6]

[6] E. Astesiano, H.-J. Kreowski, B. Krieg-Brückner, eds., *Algebraic Foundations of Systems Specification*, IFIP State-of-the-Art Report, Springer 1999

[7] M. Barr, *Terminal Coalgebras in Well-founded Set Theory*, Theoretical Computer Science 114 (1993) 299-315

[8] M. Bidoit, R. Hennicker, *Observer Complete Definitions are Behaviourally Coherent*, Report, University of Munich (1999)

[9] M. Bidoit, R. Hennicker, *Constructor-Based Observational Logic*, to appear in: J. of Logic and Algebraic Programming (2005)

[10] M. Bidoit, R. Hennicker, A. Kurz, *On the Duality between Observability and Reachability*, Proc. FOSSACS 2001, Springer LNCS 2030 (2001) 72-87

[11] M. Bidoit, R. Hennicker, A. Kurz, *Observational Logic, Constructor-Based Logic, and their Duality*, Theoretical Computer Science 298 (2003) 471-510

[12] M. Bidoit, P.D. Mosses, *CASL User Manual*, Springer LNCS 2900 (2004)

[13] J. Brown, *Minds, Machines, and the Multiverse: The Quest for the Quantum Computer*, Simon&Schuster 2000

[14] M. Broy, M. Wirsing, *Partial Abstract Types*, Acta Informatica 18 (1982) 47-64

[15] C. Cîrstea, *A Coalgebraic Equational Approach to Specifying Observational Structures*, Theoretical Computer Science 280 (2002) 35-68

[16] M. Clavel, F. Durán, S. Eker, P. Lincoln, N. Martî-Oliet, J. Meseguer, J.F. Quesada, *A Maude Tutorial*, SRI International 2000, http://maude.csl.sri.com

[17] R. Diaconescu, K. Futatsugi, *CafeOBJ Report*, World Scientific 1998

[18] A. Corradini, *A Completeness Result for Equational Deduction in Coalgebraic Specification*, Proc. WADT 1997, Springer LNCS 1376 (1997) 190-205

[19] A. Corradini, R. Heckel, U. Montanari, *From SOS Specifications to Structured Coalgebras: How to Make a Bisimulation a Congruence*, Proc. CMCS 1999, Elsevier ENTCS 19 (1999) 118-141

[20] G. Costa, G. Reggio, *Specification of Abstract Dynamic Data Types: A Temporal Logic Approach*, Theoretical Computer Science 173 (1997) 513-554

[21] H. Ehrig, H.J. Kreowski, *Refinement and Implementation*, in [6]

[22] H. Ehrig, B. Mahr, *Fundamentals of Algebraic Specification 1*, Springer 1985

[23] C.C. Elgot, *Monadic Computation and Iterative Algebrac Theories*, Logic Colloquium 1973 (North-Holland 1975) 175-230

[24] M. Erwig, *Categorical Programming with Abstract Data Types*, Proc. AMAST 1998, Springer LNCS 1548 (1998) 406-421

[25] M.M. Fokkinga, *Datatype Laws without Signatures*, Math. Structures in Comp. Sci. 6 (1996) 1-32

[26] G. Gentzen, *Untersuchungen ber das logische Schlieen*, Mathematische Zeitschrift 39 (1934) 405-431 (see en.wikipedia.org/wiki/Sequent_calculus)

[27] J. Gibbons, G. Hutton, *Proof Methods for Corecursive Programs*, Fundamenta Informaticae (2004) 1-13

[28] E. Giménez, P. Castéran, *A Tutorial on (Co-)Inductive Types in Coq*, Report, INRIA (2006)

[29] J.A. Goguen, *Stretching First Order Equational Logic: Proofs with Partiality, Subtypes and Retracts*, UCSD Report, San Diego 1997, www-cse.ucsd.edu/users/goguen/ps/ftp97.ps.gz

[30] J.A. Goguen, R. Diaconescu, *An Oxford Survey of Order Sorted Algebra*, Mathematical Structures in Computer Science 4 (1994) 363-392

[31] J.A. Goguen, K. Lin, *Behavioral Verification of Distributed Concurrent Systems with BOBJ*, Proc. Conf. on Quality Software, IEEE Press (2003) 216-235

[32] J.A. Goguen, K. Lin, G. Rosu, *Conditional Circular Coinductive Rewriting with Case Analysis*, Proc. WADT 2002, Springer LNCS 2755 (2004) 216-232

[33] J.A. Goguen, G. Malcolm, *Hidden Coinduction*, Mathematical Structures in Computer Science 9 (1999) 287-319

[34] J.A. Goguen, G. Malcolm, *A Hidden Agenda*, Theoretical Computer Science 245 (2000) 55-101

[35] J.A. Goguen, J. Meseguer, *Correctness of Recursive Parallel Nondeterministic Program Schemes*, J. of Computer and System Sciences 27 (1983) 268-290

[36] J.A. Goguen, J. Meseguer, *Unifying Functional, Object-Oriented and Relational Programming with Logical Semantics*, in: B. Shriver, P. Wegner, eds., Research Directions in Object-Oriented Programming, MIT Press (1987) 417-477

[37] 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

[38] J.A. Goguen, J.W. Thatcher, E.G. Wagner, J.B. Wright, *Initial Algebra Semantics and Continuous Algebras*, J. ACM 24 (1977) 68-95

[39] Andrew D. Gordon, *Bisimilarity as a Theory of Functional Programming*, Theoretical Computer Science 228 (1999) 5-47

[40] J.F. Groote, F. Vaandrager, *Structured Operational Semantics and Bisimulation as a Congruence*, Information and Computation 100 (1992) 202-260

[41] H. P. Gumm, *Elements of the general theory of coalgebras*, Proc. LUATCS 1999, Rand Africaans University, Johannesburg

[42] H. P. Gumm, *Equational and implicational classes of coalgebras*, Theoretical Computer Science 260 (2001) 57-69 Rand Africaans University, Johannesburg

[43] H. P. Gumm, *State based systems are coalgebras*, Cubo - Matematica Educacional 5 (2003) 239-262

[44] H. P. Gumm, *Universelle Coalgebra*, in: Th. Ihringer, *Allgemeine Algebra*, Heldermann Verlag 2003

[45] H. P. Gumm, T. Schröder, *Coalgebras of Bounded Type*, Mathematical Structures in Computer Science 12 (2002) 565-578

[46] T. Hagino, *Codatatypes in ML*, J. Symbolic Computation 8 (1989) 629-650

[47] I. Hasuo, *Modal Logics for Coalgebras - A Survey*, Report, Tokyo Institute of Technology (2003)

[48] D. Hausmann, T. Mossakowski, L. Schröder, *Iterative Circular Coinduction for CoCASL in Isabelle/HOL*, Proc. FASE 2005, Springer LNCS 3442 (2005) 341-356

[49] R. Hennicker, A. Kurz, *(Ω, Ξ)-Logic: On the Algebraic Extension of Coalgebraic Specifications*, Proc. CMCS 1999, Elsevier ENTCS 19 (1999) 164-180

[50] H. Hußmann, M. Cerioli, G. Reggio, F. Tort, *Abstract Data Types and UML Models*, Report DISI-TR-99-15, University of Genova 1999

[51] B. Jacobs, *Mongruences and Cofree Coalgebras*, Proc. AMAST 1995, Springer LNCS 936 (1995) 245-260

[52] B. Jacobs, *Behaviour-Refinement of Coalgebraic Specifications with Coinductive Correctness Proofs*, Proc. TAPSOFT 1997, Springer LNCS 1214 (1997) 787-802

[53] B. Jacobs, *Invariants, Bisimulations and the Correctness of Coalgebraic Refinements*, Proc. AMAST 1997, Springer LNCS 1349 (1997) 276-291

[54] B. Jacobs, *Exercises in Coalgebraic Specification*, in: R. Backhouse, R. Crole, J. Gibbons, eds., *Algebraic and Coalgebraic Methods in the Mathematics of Program Construction*, Springer LNCS 2297 (2002) 237-280

[55] B. Jacobs, *Introduction to Coalgebra. Towards Mathematics of States and Observations*, book draft, Radboud University Nijmegen 2005

[56] B. Jacobs, J. Rutten, *A Tutorial on (Co)Algebras and (Co)Induction*, EATCS Bulletin 62 (1997) 222-259

[57] S. Kamin, *Final Data Type Specifications: A New Data Type Specification Method*, ACM TOPLAS 5 (1983) 97-123

[58] U. Kühler, C.-P. Wirth, *Conditional Equational Specifications of Data Types with Partial Operations for Inductive Theorem Proving*, Proc. RTA 1997, Springer LNCS 1232 (1997) 38-52

[59] A. Kurz, *Specifying Coalgebras with Modal Logic*, Proc. CMCS 1998, Elsevier ENTCS 11 (1998) 56-70

[60] A. Kurz, *Coalgebras and Modal Logic*, Course Notes for ESSLLI 2001, CWI Amsterdam

[61] A. Kurz, R. Hennicker, *On Institutions for Modular Coalgebraic Specifications*, Report, University of Munich 2000

[62] A. Kurz, J. Rosický, *Modal Predicates and Coequations*, Proc. CMCS 2002, Elsevier ENTCS 65 (2002) 1-20

[63] J. Lambek, *A Fixpoint Theorem for Complete Categories*, Mathematische Zeitschrift 103 (1968) 151-161

[64] F.W. Lawvere, *Functorial Semantics of Algebraic Theories*, Proc. Nat. Acad. Sci. U.S.A. 50 (1963) 869-872

[65] D.J. Lehmann, M.B. Smyth, *Algebraic Specification of Data Types: A Synthetic Approach*, Mathematical Systems Theory 14 (1981) 97-139

[66] K. Lin, J.A. Goguen, *A Hidden Proof of the Alternating Bit Protocol*, Dept. of CS and Engineering, UCSD 2003

[67] D. Lucanu, O. Gheorgies, A. Apetrei, *Bisimulation and Hidden Algebra*, Proc. CMCS 1999, Elsevier ENTCS 19 (1999) 181-200

[68] G. Malcolm, *Behavioural Equivalence, Bisimulation, and Minimal Realisation*, Proc. WADT 1995, Springer LNCS 1130 (1996) 359-378

[69] E.G. Manes, *Algebraic Theories*, Springer 1976

[70] E.G. Manes, M.A. Arbib, *Algebraic Approaches to Program Semantics*, Springer 1986

[71] K. Meinke, J.V. Tucker, *Universal Algebra*, in: S. Abramsky et al., eds., Handbook of Logic in Computer Science, Vol. 1, Clarendon Press (1992) 188-411

[72] S. Meng, B.K. Aichernig, *Component-Based Coalgebraic Specification and Verification in RSL*, UNU/IIST Report No. 267, Macau 2002

[73] S. Meng, B.K. Aichernig, *Coalg-KPF: Towards a Coalgebraic Calculus for Component-Based Systems*, UNU/IIST Report No. 271, Macau 2003

[74] S. Meng, B.K. Aichernig, *Towards a Coalgebraic Semantics of UML: Class Diagrams and Use Cases*, UNU/IIST Report No. 272, Macau 2003

[75] S. Meng, L.S. Barbosa, *On Refinements of Generic Software Components*, UNU/IIST Report No. 281, Macau 2003

[76] S. Meng, Z. Naixiao, B.K. Aichernig, *The Formal Foundations in RSL for UML Statechart Diagrams*, UNU/IIST Report No. 299, Macau 2004

[77] J. Meseguer, *Membership Algebra as a Logical Framework for Equational Specification*, Proc. WADT 1997, Springer LNCS 1376 (1998) 18-61

[78] 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

[79] E. Meijer, M. Fokkinga, R. Paterson, *Functional Programming with Bananas, Lenses, Envelopes and Barbed Wire*, Proc. FPCA 1991, Springer LNCS 523 (1991) 124-144

[80] D. Miller, G. Nadathur, F. Pfenning, A. Scedrov, *Uniform Proofs as a Foundation for Logic Programming*, Annals of Pure and Applied Logic 51 (1991) 125-157

[81] R. Milner, *Communication and Concurrency*, Prentice-Hall 1989

[82] R. Milner, *A Communicating and Mobile Systems: the π-Calculus*, Cambridge University Press 1999

[83] Till Mossakowski, Horst Reichel, Markus Roggenbach, Lutz Schröder, *Algebraic-coalgebraic specification in CoCASL*, to appear in: J. of Logic and Algebraic Programming 67 (2006) 121-143

[84] M. Müller-Olm, D. Schmidt, B. Steffen, *Model Checking: A Tutorial Introduction*, Proc. SAS 1999, Springer LNCS 1694 (1999) 330-394

[85] P. Padawitz, *Inductive Expansion: A Calculus for Verifing and Synthesizing Functional and Logic Programs*, J. Automated Reasoning 7 (1991) 27-103

[86] P. Padawitz, *Deduction and Declarative Programming*, Cambridge University Press 1992

[87] P. Padawitz, *Inductive Theorem Proving for Design Specifications*, J. Symbolic Computation 21 (1996) 41-99

[88] P. Padawitz, *Proof in Flat Specifications*, in [6]

[89] P. Padawitz, *Swinging Types = Functions + Relations + Transition Systems*, Theoretical Computer Science 243 (2000) 93-165

[90] P. Padawitz, *Formale Methoden des Systementwurfs*, Course Notes, University of Dortmund, padawitz.de/TdP96.pdf

[91] P. Padawitz, *Swinging Types At Work*, Report, University of Dortmund 2000, ls5-www.cs.uni-dortmund.de/∼peter/BehExa.ps.gz

[92] P. Padawitz, *Swinging UML: How to Make Class Diagrams and State Machines Amenable to Constraint Solving and Proving*, Proc. ⟨⟨UML⟩⟩2000, Springer LNCS 1939 (2000) 162-177

[93] P. Padawitz, *Basic Inference Rules for Algebraic and Coalgebraic Specifications*, Report, University of Dortmund 2001,
ls5-www.cs.uni-dortmund.de/∼peter/WADT01.ps.gz

[94] P. Padawitz, *Structured Swinging Types*, Report, University of Dortmund 2004, ls5-www.cs.uni-dortmund.de/∼peter/SST.ps

[95] P. Padawitz, *Expander2: A Formal Methods Presenter and Animator*, ls5-www.cs.uni-dortmund.de/∼peter/Expander2.html

[96] P. Padawitz, *Expander2: Towards a Workbench for Interactive Formal Reasoning*, ls5-www.cs.uni-dortmund.de/∼peter/Expander2/Chiemsee.ps

[97] B.C. Pierce, Basic Category Theory for Computer Scientists, MIT Press 1991

[98] W. Pohlers, *Subsystems of set Theory and Second Order Number Theory*, in: S.R. Buss, ed., Handbook of Proof Theory, Elsevier (1998) 209-335

[99] E. Poll, J. Zwanenburg, *From Algebras and Coalgebras to Dialgebras*, Proc. CMCS 2001, Elsevier ENTCS 44 (2001) 1-19

[100] H. Reichel, *An Approach to Object Semantics based on Terminal Coalgebras*, Mathematic Structures in Computer Science 5 (1995) 129-152

[101] H. Reichel, *Unifying ADT- and Evolving Algebra Specifications*, EATCS Bulletin 59 (1996) 112-126

[102] G. Roşu, J.A. Goguen, *Hidden Congruent Deduction*, Proc. FTP 1998, Springer LNAI 1761 (2000) 252-266

[103] J. Rothe, H. Tews, B. Jacobs, *The Coalgebraic Class Specification Language CCSL*, J. of Universal Computer Science 7 (2001) 175-193

[104] J.J.M.M. Rutten, *Universal Coalgebra: A Theory of Systems*, Theoretical Computer Science 249 (2000) 3-80

[105] J.J.M.M. Rutten, D. Turi, *Initial Algebra and Final Coalgebra Semantics for Concurrency*, Report CS-R9409, CWI, SMC Amsterdam 1994

[106] K. Schütte, *Proof Theory*, Springer 1977 An algebraic theory of recursive definitions and recursive languages

[107] E.G. Wagner, *An algebraic theory of recursive definitions and recursive languages*, Proc. STOC 1971, http://portal.acm.org/citation.cfm?id=805034

[108] E.G. Wagner, J.W. Thatcher, J.B. Wright, *Programming Languages as Mathematical Objects*, Proc. MFCS 1978, Springer LNCS 64 (1978) 84-101

[109] M. Wand, *Final Algebra Semantics and Data Type Extensions*, J. Computer and System Sciences 19 (1979) 27-44

[110] J.B. Warmer, A.G. Kleppe, *The Object Constraint Language*, Addison-Wesley 1999

[111] W. Wechler, *Universal Algebra for Computer Scientists*, Springer 1992

[112] P. Wegner, *Why Interaction Is More Powerful Than Algorithms*, Communications of the ACM 40 (1997) May, 80-91

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

[114] U. Wolter, *On Corelations, Cokernels, and Coequations*, Proc. CMCS 2000, Elsevier ENTCS 33 (2000) 347-366