Swinging Types = Functions + Relations + Transition Systems

Peter Padawitz
padawitz@cs.uni-dortmund.de
http://ls5.cs.uni-dortmund.de/~peter
University of Dortmund

April 14, 2000

Abstract

Swinging types provide a specification and verification formalism for designing software in terms of many-sorted logic. Current formalisms, be they set- or order-theoretic, algebraic or coalgebraic, rule- or net-based, handle either static system components (in terms of functions or relations) or dynamic ones (in terms of transition systems) and either structural or behavioral aspects, while swinging types combine equational, Horn and modal logic for the purpose of applying computation and proof rules from all three logics.

A swinging specification separates from each other *visible sorts* that denote domains of data identified by their structure; *hidden sorts* that denote domains of data identified by their behavior in response to *observers*; *μ-predicates* (least relations) representing inductive(ly provable) properties of a system; and *ν-predicates* (greatest relations) representing complementary "coinductive" properties, which often describe behavioral aspects "in the infinity".

A model that combines static with dynamic features and structural with behavioral aspects of a system is obtained naturally if all involved entities (objects, states, etc.) are presented as terms built up of constructors for visible or hidden sorts and if functions are specified by conditional equations (= functional programs), least relations by Horn clauses (= logic programs or transition system specifications) and greatest relations by *co-Horn clauses*. Term equivalences are either *structural* or *behavioral*, the former being least, the latter being greatest solutions of particular axioms derived from the type's signature.

This paper mainly presents the theoretical foundations of swinging types, such as standard (term) models, criteria for structural and behavioral consistency, and proof rules. Swinging types admit flexible design guidelines, tailored to particular objectives or application fields. Suitable design methods may be based upon this and the companion paper [62] that explores various application areas and illustrates how swinging types may realize different programming or specification styles. As to structuring concepts for swinging types, parameterization and genericity are involved in this paper, while [64] deals with extensions and refinements.

Contents

1 Introduction

In contrast to modal logic and most approaches for specifying dynamic systems (see, e.g., [1, 25]) we propose a one-tiered framework that admits the specification of "static" data types and "dynamic" transition systems within a uniform logic. In modal logic, state transitions are interpreted on a higher level that does not interfere with the structure of individual states. Swinging types regard states as hidden objects, transition labels as visible data and transition relations as *dynamic predicates*. The behavioral identity of a hidden object depends on functional or relational *observers*, in other approaches also called selectors, accessors, attributes, inquiry operations, methods, mutators, destructors, etc. In functional approaches, behavioral equality usually comes as a sort of contextual equivalence, in modal logic as bisimilarity. In both cases behavioral equality is dual to structural equality insofar as the former is the least and the latter the greatest relations satisfying certain compatibility axioms. Swinging types admit all ways of specifying a behavioral equality, be they functional, relational or "transitional", i.e., determined by dynamic predicates. The latter case motivates the introduction of *weak congruences* that are compatible with static predicates, but only *zigzag compatible* with dynamic ones.

A swinging specification starts out from constructors for building up both visible and hidden data domains. Visible domains precede the hidden ones. A visible domain is characterized by the coincidence of its structural with its behavioral equality. Constructors of visible data are not allowed to have hidden arguments. This ensures that the theory of a hidden type is consistent w.r.t. its visible subtype. A swinging specification need not have hidden sorts, but if there are hidden sorts, there must also be visible ones. Otherwise hidden objects cannot be distinguished from each other. Formally, each hidden domain must be equipped with at least one functional observer that maps to a visible sort or one relational observer, regarded as a function that maps to the visible domain of truth values. Otherwise the hidden domain collapses because its behavioral equality identifies all its elements.

Besides constructors, a swinging specification defines functions and *μ-predicates* in terms of Horn axioms that represent functional-logic programs or transition system specifications. μ-predicates are interpreted as the least solutions of their axioms. μ-predicates are often existential properties such as liveness or reachability. Roughly said, all inductively definable properties are μ-predicates. Hence structural equalities are μ-predicates. μ-predicates dealing with "infinite" objects are often "limits" of conditions on the objects' finite approximations. *ν-predicates* are usually complements of μ-predicates. They represent universal properties, and, if they cannot be turned into equivalent μ-predicates, they often express aspects of behavior "in the infinity", such as safety and invariance conditions on state sequences. Formally, ν-predicates are specified in terms of *co-Horn clauses* and interpreted as the greatest solutions of their axioms. Behavioral equalities are ν-predicates. Above the ν-predicates, a swinging type may have further μ-predicates whose axioms are *generalized Horn clauses*. As co-Horn clauses may involve existential quantifiers in the conclusion, generalized Horn clauses may involve universal quantifiers in the premise. Since these quantifiers may violate the continuity of the consequence operators induced by the axioms, we provide a continuity criterion that generalizes the notion of *image finiteness* from transition systems to arbitrary goals.

The notions "μ-predicate" and "ν-predicate" stem from modal logic's μ-calculus (cf., e.g., [48, 73]) and relational fixpoint semantics (cf., e.g., [36]). The least and greatest fixpoints of state set operators used to define alternation-free μ-formulas can be translated directly into swinging specifications of μ- resp. ν-predicates (see Section 2).

Besides modal logic, swinging types integrate concepts, methods and results from many other formal approaches to system specification and verification. First of all, there is final-semantics approach to data types that was introduced for modelling *permutative types* such as finite sets, multisets and arrays with a finite domain (cf., e.g., [28, 75, 46]). [29, 32] extended it to the hidden-type approach that also covers object-oriented—though purely functional—specifications. From *dynamic data types* we adopt the specification of labelled transition systems (LTS) as ternary predicates (cf. [8, 7, 4]). *Stratified* logic programs with *stable* or *perfect* models provide ideas for constructing swinging types hierarchically (cf. [2]).

Coinductive function definitions in category theory [68, 44] and formats of transition system specifications [37] led us to the criterion of *coinductivity* for the behavioral consistency of a swinging type (see Section 6). Given a suitable functor F, category theorists call a function to be defined by coinduction if the definition is derived from the unique morphism that maps an F-coalgebra to the final F-coalgebra. This dualizes the category-theoretic notion of a definition by induction that is derived from the unique morphism that maps the initial F-algebra to an F-algebra. Initial F-algebras and final F-coalgebras are isomorphisms that are composed of the constructors and destructors of the type described by F. The connections between swinging types and the category-theoretic approach to data types is treated in detail in [62]. Besides the notion of coinductivity the category-theoretic approach yields the insight that not only visible, but also hidden types have constructors. On the other hand, it is purely functional and thus does not contribute to the axiomatic specification of predicates, in particular dynamic ones. Here modal logic and process algebra provide more inspirations.

The rewriting-oriented criteria developed in [57, 59, 60] are fully applicable to swinging types. This provides the basis for ensuring that a swinging type is *functional*, i.e. each of its ground terms is structurally equivalent to a unique normal form (= term consisting of constructors). This seems to exclude the specification of partial functions. However, partiality can always be simulated by introducing sum sorts that comprise "defined" and "undefined" values such as exceptions, error messages, etc. (see, e.g., [34] and the *exception monad* of [54] as used in [62]). Moreover, strong equality turns out to be a behavioral equality induced by a destructor that identifies exceptions, and even arbitrary partial-recursive functions can be specified by axioms of a swinging type (cf. [62], Section 7). Each functional specification can be transformed into an equivalent relational one whose only functions are constructors, while defined functions are transformed into corresponding input-output relations. This fact is crucial for the correctness of applying one of the main proof rules for swinging types, namely fixpoint induction, not only to (μ-)predicates, but also to defined functions. A functional specification can also be extended systematically by axioms for the complements of its structural equalities, in other words, axioms for inequalities. This entails the correctness of practically indispensable proof rules such as *term splitting* and *clash*. The complements of non-equality predicates are accomplished by simply negating axioms (see Section 4).

The axioms for behavioral equalities are determined by those defined functions, static or dynamic predicates that are declared as *destructors*, *separators* and *transition predicates*, respectively, altogether called *observers*. Observational specifications in the sense of [17, 41, 16] and behavioral or hidden ones in the sense of [32, 69] deal exclusively with destructors (called attributes/methods in [32]). λ-calculi, process logics and dynamic data types are based on labelled transition systems (LTS), i.e. transition predicates. On the one hand, only [8] and [21] regard an LTS as a predicate of a many-sorted specification. On the other hand, the dynamic-type approach lacks specification and proof methods that are as powerful as those invented in process algebra [9] and modal logics for proving properties of LTS ("model checking"). But the dynamic-type approach keeps to first-order logic, while the modal-logic and process-algebra reasoning about processes and LTS leaves the structure of individual states out of its discourse.

Similarly to functional-logic programs and transition system specifications, the axioms of a swinging type represent more or less inductive definitions of (defined) functions or (μ-)predicates on constructors.

This is necessary for ensuring that the specification is functional. *Coinductive axioms*, on the other hand, guarantee that the specification is *behaviorally consistent* (see below). For instance, both the visible type of (finite) lists and the hidden type of (infinite) streams have a constructor append-to-the-left, denoted by $\_ :: \_ : entry \times list \to list$ and $\_\&\_ : entry \times stream \to stream$, respectively. In both cases, there are defined functions *head* and *tail*, specified by the same axioms:

$$\begin{array}{ll} head(x :: L) \equiv x & tail(x :: L) \equiv L \\ head(x\&s) \equiv x & tail(x\&s) \equiv s \end{array}$$

Obviously, these equations are part of inductive definitions of *head* and *tail* and thus part of a functional list resp. stream specification. *Co*inductivity, however, is a requirement to axioms for hidden symbols only and thus may or not hold only for the last two equations. Indeed, they are coinductive because we have declared *head* and *tail* as observers. In this simple case it is quite easy to conclude that behavioral stream equality, say $\sim$, is compatible with all involved functions, i.e. *head*, *tail* and &. Declaring *head* and *tail* as observers means to axiomatize $\sim$ as follows:

$$s \sim s' \quad \Rightarrow \quad head(s) \equiv head(s') \wedge tail(s) \sim tail(s'). \tag{1}$$

(1) *is* the compatibility of $\sim$ with *head* and *tail*. But $\sim$ is also compatible with & because $\sim$ denotes the greatest solution of (1) and thus $s \sim s'$ holds true *if and only if* the conclusion of (1) holds true. The argument would fail if *head* and *tail* were not declared as observers. Indeed, the coinductivity requirement to axioms for "non-observing" symbols are more restrictive (see Section 6). For instance, suppose that the list specification is extended by a hidden sort *bag* for finite multisets, the defined function $card : bag \times entry \to nat$ returning the number of occurrences of an element in a bag is declared as an observer and the embedding $mkbag : list \to bag$ of lists into bags is the only bag constructor. Then there is a defined function $chooselist : list \to bag$ specified inductively by the axiom

$$chooselist(bag(L)) \;\equiv\; L. \tag{2}$$

Declaring *chooselist* as an observer would lead to the axiom

$$b \sim b' \quad \Rightarrow \quad chooselist(b) \equiv chooselist(b')$$

for behavioral bag equality, which does not comply with our intuition about this equality. Hence *chooselist* cannot be an observer and thus—as the reader of Section 6 will confirm—(2) is not coinductive. Indeed, if behavioral bag equality were compatible with *chooselist*, it would coincide with list equality!

An "LTS-inspired" stream specification replaces the functional observers *head* and *tail* by a transition predicate $\_ \xrightarrow{\_} \_ : stream \times entry \times stream$, the axioms for *head* and *tail* by $x\&s \xrightarrow{x} s$ and (1) by:

$$\begin{array}{l} s \sim s' \quad \Rightarrow \quad (s \xrightarrow{x} t \quad \Rightarrow \quad \exists t' : (s' \xrightarrow{x} t' \wedge t \sim t')), \\ s \sim s' \quad \Rightarrow \quad (s' \xrightarrow{x} t' \quad \Rightarrow \quad \exists t : (s \xrightarrow{x} t \wedge t \sim t')). \end{array}$$

The syntax of a swinging type leads directly to its *Herbrand model*, $Her(SP)$, which is a pure term model and thus interprets both structural equalities and behavioral ones as term relations, called *structural* and *behavioral SP-equivalence*, respectively, denoted by $\equiv_{SP}$ resp. $\sim_{SP}$. The *initial SP-model*, $Ini(SP)$, is the quotient of $Her(SP)$ by structural SP-equivalence, the *final SP-model*, $Fin(SP)$, is the quotient of $Her(SP)$ by behavioral SP-equivalence. The latter deviates from other final-semantics approaches where the final model comes as a quotient of the initial one. In fact, $\equiv_{SP}$ is included in $\sim_{SP}$ and thus some quotient of $Ini(SP)$ is isomorphic to $Fin(SP)$. However, the theory of the final model is easier to handle if one constructs it as a quotient of the Herbrand model.

The standard axioms for structural equalities render $\equiv_{SP}$ a congruence relation. The algebraist likes congruences because they admit the construction of quotient models. The theorem prover is less keen on new models, but on the correctness of term replacement w.r.t. an equivalence relation, and this is guaranteed *for all first-order formulas* only if the relation is a congruence, in other words, first-order formulas are *congruence invariant*:

$$t \equiv_{SP} t' \quad \Rightarrow \quad \text{for all first-order formulas } \varphi : Her(SP) \models \varphi(t) \Leftrightarrow \varphi(t'). \tag{3}$$

Behavioral SP-equivalence is not a congruence, but a *weak congruence*. Weak congruences are compatible with functions and *static* predicates, but only *zigzag compatible* with *dynamic* predicates. Roughly said, they are bisimulations, generalized to arbitrary dynamic predicates. Modal logic provided us the idea of possible classes of first-order formulas whose elements are *weak-congruence invariant*:

$$t \sim_{SP} t' \quad \Rightarrow \quad \text{for all } \textit{poly-modal} \text{ formulas } \varphi : Her(SP) \models \varphi(t) \Leftrightarrow \varphi(t'). \tag{4}$$

We introduce three classes of modal first-order formulas. Those called **modal** formulas are the results of direct translations of modal-logic sentences into predicate logic. Such translations—whose images are also called *modal* or *guarded fragments* of first-order logic—have been studied by, e.g., Ohlbach [55], Bergstra and van Benthem [13, 14]. The main idea is to "internalize" the "frame" or LTS, which determines the interpretation of modal-logic sentences, as a binary, or if the LTS is labelled, ternary predicate (see Section 2). Modal formulas have a single ("state") variable and can be shown to be *bisimulation invariant* (see below). The greater class of **poly-modal** formulas admits several variables, but restricts (analogously to modal formulas) the "target term" of each dynamic-predicate occurrence to an existentially quantified variable. A **weakly modal** formula may also have *free* variables as target terms, which comprise the *output* of the formula (see Section 2). Weakly modal formulas with empty output are poly-modal. For guaranteeing that the final model of a swinging specification satisfies its axioms, the premises of the Horn axioms must be weakly modal, while the conclusions of the co-Horn axioms must be poly-modal (see Section 3).

There are differences between van Benthem's modal fragment and poly-modal formulas that forbid the direct application of his results to swinging types. [14] translates *propositional* modal logic and thus formulas of the modal fragment are built over a one-sorted signature with only unary (static) and binary (dynamic) predicates, while we start out from a many-sorted signature with predicates of various arities. Nevertheless, even poly-modal formulas enjoy a *Hennessy-Milner theorem*[1], i.e. (4) together with the converse:

$$t \sim_{SP} t' \quad \Leftarrow \quad \text{for all } \textit{poly-modal} \text{ formulas } \varphi : Her(SP) \models \varphi(t) \Leftrightarrow \varphi(t'). \tag{5}$$

This is trivially valid because $\sim_{SP}$ is reflexive and $t \sim x$ is a poly-modal formula.

However, van Benthem ([14], Thm. 4.18) *characterizes* his modal fragment as the class of (bisimulation) invariant first-order formulas. In Section 7, we generalize this *modal invariance theorem* to many-sorted logic and our class of modal (not of poly-modal) formulas. Moreover, it is not the weak congruences that replace the bisimulations in van Benthem's theorem, but something in between: on the one hand, weak congruences, but only w.r.t. unary (static) and binary (dynamic) predicates, on the other hand, pairs $(a, b) \in A \times B$ where A and B are *different* models. Hence our Hennessy-Milner theorem deals with terms replacements within a single model A, while our modal invariance theorem deals with model replacements and thus adopts the two-tiered modal-logic view that different states pertain to different models (see above).

[1] For the original modal-logic version, see [40], Thm. 2.2, or [73], Thms. 5.3.2 and 5.3.3.

In the dynamic-data-type approach of [5], the Hennessy-Milner Theorem has been generalized to a class of *observational formulas* with *patterns of experiments*. They somewhat resemble poly-modal formulas, but are built upon a fixed, rather unstructured interpretation of their visible components.

A structure A interpreting the signature of a swinging specification SP is *behaviorally SP-consistent* if A interprets behavioral equality as a weak congruence and if the quotient of A by that weak congruence satisfies the axioms of SP. SP itself is *behaviorally consistent* if the Herbrand model of SP interprets behavioral equality as a weak congruence. The interpretation is denoted by $\sim_{SP}$, the quotient is called the final SP-model (see above). The modality assumptions on the axioms of SP (see above) imply that each model of SP with a weakly congruent interpretation of behavioral equality is behaviorally SP-consistent (Thm. 3.9(b)). Hence the final model is really a model if $\sim_{SP}$ is a weak congruence. This justifies the notion of behavioral consistency and shows the significance of syntactic criteria for this property.

Swinging types "swing" between many poles: between visible and hidden domains, between several states (= individual hidden objects), between functions declared as constructors and those used as defined functions, between structural and behavioral equalities, between functional-logic programs for functions and static predicates and "transitional programs" for dynamic predicates, between μ- and ν-predicates. Usually, not all these concepts are needed simultaneously. There are swinging types where structural equalities play the dominant rôle, while other types are specified adequately only in terms of observers and behavioral equalities. The integrative approach just makes it easier to state and understand both similarities and conceptual differences between specification formalisms, which so far have been presented separately from each other. Readers who are familiar with other specification approaches are invited to reformulate results of this paper in terms of those approaches. This is also a goal of the integration: to make use of theorems about rule correctness, consistency, etc. in various formal settings.

Section 2 provides the syntax of swinging specifications, recapitulates basic notions of many-sorted logic and introduces modal, co-Horn and generalized Horn formulas. Section 3 deals with the semantics of swinging specifications, in particular, bisimulations, weak congruences and monotone structures. General connections between modality and weak congruences are established by Theorems 3.8 and 3.9. Section 4 focuses on functionality, reviews relational-fixpoint theorems, defines the standard models of a swinging specification and presents basic proof rules that draw on the syntax of swinging types, the Herbrand model's interpretation of predicates as least or greatest solutions of axioms and, as far as defined functions are concerned, on functionality. Section 5 deals with particular properties of the final model and with relationships between several specifications, such as relative completeness, monotonicity, consistency and inductive equivalence. Moreover, image finiteness is established as a criterion for the continuity of the consequence operators that build up the Herbrand model. Section 6 is devoted the behavioral-consistency criterion of coinductivity. The modal invariance theorem is presented and proved in Section 7.

2 The syntax of swinging types

We assume familiarity with the basic notions of many-sorted logic with equality (cf., e.g., [35, 24, 77]). As has been shown by, e.g., [31, 53, 56, 57, 60], this logic admits presenting and verifying not only primitive data types with first-order functions, but also generic types with almost all features of current functional-logic specification or programming languages.

For all expressions **e**, **var(e)** and **free(e)** denote the sets of all resp. free variables of **e**. e is **ground** if $var(e)$ is empty. **e(t)** denotes an expression that includes the (tuple of) subexpression(s) **t**. **e[t/u]** stands for **e** with **t** substituted for **u**.

Given a set S of sorts, $w = s_1 \dots s_n \in S^n$ and an S-sorted set A, A_w stands for the product $A_{s_1} \times \dots \times A_{s_n}$. Given two S-sorted sets A and B, an **S-sorted binary relation** $\approx\ \subseteq A \times B$ is a family of relations $\{\approx_s \subseteq A_s \times B_s\}_{s \in S}$. $\approx$ extends to a family of relations $\{\approx_w\ \subseteq A_w \times B_w\}_{w \in S^+}$ on products and to a relation $\approx\ \subseteq [I \to A] \times [I \to B]$ on functions as follows:

$$(a_1, \dots, a_n) \approx (b_1, \dots, b_n) \iff_{def} \forall 1 \le i \le n : a_i \approx b_i \quad \text{resp.} \quad f \approx g \iff_{def} \forall i \in I : f(i) \approx g(i).$$

Example 2.1 We start with an introductory example of a swinging specification. Precise definitions are given afterwards.

ORDER

`sorts`	$entry$		
`preds`	$\_ \not\equiv \_ : entry \times entry$		(predicates)
	$\_ \le \_ : entry \times entry$		
	$\_ > \_ : entry \times entry$		
`vars`	$x, y : entry$		(variables)
`axioms`	$x \equiv y \vee x \not\equiv y$	$x \not\equiv y \Leftrightarrow \neg(x \equiv y)$	
	$x \le y \vee x > y$	$x > y \Leftrightarrow \neg(x \le y)$	

LISTORD = ORDER `then`

`vissorts`	$bool \quad list = list(entry)$		(visible sorts)
`hidsorts`	$entry \to entry \quad entry \to bool$		(hidden sorts)
`constructs`	$true, false : \to bool$		(constructors)
	$nil : \to list$		
	$\_ :: \_ : entry \times list \to list$		
	$\lambda y.not(eq(\_, y)) : entry \to (entry \to bool)$		
`destructs`	$apply : ((entry \to entry) \times entry) \to entry$		(destructors)
	$apply : ((entry \to bool) \times entry) \to bool$		
`defuncts`	$not : bool \to bool$		(defined functions)
	$eq : entry \times entry \to bool$		
	$[\_] : entry \to list$		
	$\_@\_ : list \times list \to list$		
	$map : (entry \to entry) \times list \to list$		
	$filter : (entry \to bool) \times list \to list$		
	$remove : entry \times list \to list$		
`static` μ`-preds`	$\_ \not\equiv \_ : bool \times bool$		
	$\_ \le \_ : bool \times bool$		
	$\_ > \_ : bool \times bool$		
	$\_ \in \_ : entry \times list$		
	$\_ \notin \_ : entry \times list$		
	$sorted : list$		
	$exists, forall : (entry \to bool) \times list$		
`vars`	$x, y : entry \quad b : bool \quad L, L' : list \quad f : entry \to entry \quad g : entry \to bool$		
`Horn axioms`	$not(true) \equiv false$	$eq(x, y) \equiv true \Leftarrow x \equiv y$	
	$not(false) \equiv true$	$eq(x, y) \equiv false \Leftarrow x \not\equiv y$	

$$
\begin{array}{lll}
 & [x] \equiv x :: nil & \\
 & nil@L \equiv L & \\
 & (x :: L)@L' \equiv x :: (L@L') & \\
 & map(f, nil) \equiv nil & \\
\text{(A)} & map(f, x :: L) \equiv f(x) :: map(f, L) & \\
 & filter(g, nil) \equiv nil & \\
 & filter(g, x :: L) \equiv x :: filter(g, L) \ \Leftarrow\ g(x) \equiv true & \\
 & filter(g, x :: L) \equiv filter(g, L) \ \Leftarrow\ g(x) \equiv false & \\
\text{(B)} & remove(x, L) \equiv filter(\lambda y.not(eq(x, y)), L) & \\
 & true \not\equiv false & \\
 & false \leq true & true > false \\
 & b \leq b & \\
 & x \in x :: L & x \notin nil \\
 & x \in y :: L \ \Leftarrow\ x \in L & x \notin y :: L \ \Leftarrow\ x \not\equiv y \wedge x \notin L \\
 & sorted(nil) & \\
 & sorted(x :: nil) & \\
 & sorted(x :: y :: L) \ \Leftarrow\ x \leq y \wedge sorted(y :: L) & \\
 & exists(g, x :: L) \ \Leftarrow\ g(x) \equiv true & \\
 & exists(g, x :: L) \ \Leftarrow\ exists(g, L) & \\
 & forall(g, nil) & \\
 & forall(g, x :: L) \ \Leftarrow\ g(x) \equiv true \wedge forall(g, L) & \\
\text{(C)} & apply(\lambda y.not(eq(x, y)), y) \ \equiv\ not(eq(x, y)) \ \square &
\end{array}
$$

A **parameterized specification** SP such as LISTORD contains **parameter specifications** (here: ORDER) that consist of **empty sorts**, defined functions, predicates and arbitrary first-order axioms. A sort s of SP is empty if SP does not contain constructors of type $w \to s$. Empty sorts correspond to the type variables of polymorphic type expressions. Consequently, structured sort symbols such as $list(entry)$ denote polymorphic types. The equation $list = list(entry)$ declares $list$ as a short notation for $list(entry)$. We use CASL notations for structuring specifications (cf. [19]): `then` denotes the **extension** operator that combines a specification SP with additional signature symbols and axioms, `and` builds the **union** of specifications and identifies synonymous (and equally-typed) symbols of the argument specifications.

Given terms t, u and $x \in var(t)$, the **λ-abstraction** $\lambda x.t$ is an implicit constructor and the expression $t(u)$ is a short notation for the term $apply(t, u)$ where $apply$ is a (usually implicit) defined function. For instance, Axiom A implicitly involves the defined function $apply : ((entry \to bool) \times entry) \to bool$ and Axiom B uses the constructor $\lambda y.not(eq(\_, y)) : entry \to (entry \to bool)$. Functional sorts, λ-constructors, $apply$-functions and axioms like C are usually not listed explicitly.

Swinging signatures mainly distinguish between visible and hidden sorts, constructors and defined functions, μ- and ν-predicates and static and dynamic predicates. These sets of symbols cover structural as well as behavioral equalities and the observers that determine the latter. The distinctions were motivated intuitively in Section 1. Further more technical reasons can only be given after the signatures are equipped with axioms (see Def. 2.4).

Definition 2.2 (signatures, terms, atoms) A **signature** $\Sigma = (S, F, P)$ consists of a set S of **sorts** and S^+-sorted sets F of **function symbols** and P of **predicates** such that P splits into sets μP of **μ-predicates** and νP of **ν-predicates**. s, s', etc. stand for single sorts, w for sort sequences. A function symbol $f \in F_{\Sigma,ws}$ is written as $f : w \to s$ and a predicate $r \in P_{\Sigma,w}$ as $r : w$.

For all $s \in S$, μP implicitly includes the **(structural) equality (predicate)** $\equiv_s : ss$.[2] Σ is **swinging** if the following conditions hold true:

- S splits into a set $visS$ of **visible sorts** and a set $hidS$ of **hidden sorts**.
- F splits into a set of **constructors** and a set DF of **defined functions**.
- For all $s \in hidS$, DF includes a set of **destructors** $f : sw \to s'$.
- For all $s \in hidS$, μP includes a set of **separators** $r : sw$ and a set of **transition predicates** $\delta : sws'$.
- For all $s \in S$, νP implicitly contains the **behavioral equality (predicate)** $\sim_s : ss$.

A function symbol $f : w \to s$ is **visible** if $ws \in visS^+$. f is **hidden** if f is not visible. For all constructors $c : w \to s$, $s \in visS$ implies $w \in visS^*$. A predicate $r : w$ is **logical** if r is not an equality predicate. r is **visible** if $w \in visS^+$. r is **hidden** if r is not visible. Structural equalities are μ-predicates. Destructors, separators and transition predicates are called **observers**.

- Visible equality predicates, separators and ν-predicates belong to the set $statP$ of **static predicates**.
- Transition and hidden equality predicates belong to the set $dynP$ of **dynamic predicates**, which are always μ-predicates.

Each predicate is static or dynamic. Only visible equality predicates are static *and* dynamic.

Let X be a set of S-sorted variables. $T_\Sigma(X)$ and T_Σ denote the S-sorted sets of Σ-terms and ground Σ-terms, respectively, which are defined as usual. Each Σ-term defines a new function symbol: if $t \in T_\Sigma(X)_s$, $var(t) = \{x_1, \ldots, x_n\}$ and for all $1 \leq i \leq n$, s_i is the sort of x_i, then $t : s_1 \ldots s_n \to s$. We write F^*_Σ for the set of all function symbols derived from $T_\Sigma(X)$.

A **Σ-normal form** is a Σ-term that consists of constructors and variables. $NF_\Sigma(X)$ and NF_Σ denote the S-sorted sets of Σ-normal forms and ground Σ-normal forms, respectively. $t \in T_\Sigma(X)_s$ is **visible** resp. **hidden** if s is visible resp. hidden. t is **unary** if $var(t)$ is a singleton.

Given $r : w \in P$ and $t \in T_\Sigma(X)_w$, $r(t)$ is a **Σ-atom**. If r is a μ-predicate, then $r(t)$ is a **μ-atom**. Otherwise $r(t)$ is a **ν-atom**. $r(t)$ is an **equation** if r is an equality predicate. An equation between term tuples t and t' stands for the conjunction of the equations between corresponding components of t resp. t'. An atom $r(t)$ is **logical**, **visible**, **hidden**, **static** or **dynamic** if r is logical, visible, hidden, static or dynamic, respectively. ❑

Equality and behavioral equalities will not be listed explicitly in signatures examples. Behavioral equalities are specified via observers (see Def. 2.4). Each function symbol $f : s \to s'$ is also regarded as a constructor constant of the functional sort $s \to s'$. Functional sorts are hidden. $s \to s'$ has the (implicit) observer $apply : (s \to s') \times s \to s'$. Non-constant functional-sort constructors such as function composition are specified in terms of $apply$:

$$apply(f \circ g, x) \;\equiv\; apply(f, apply(g, x))$$

or, in more readable notation,

$$(f \circ g)(x) \;\equiv\; f(g(x)).$$

Hence, semantically, the behavioral equality for a functional sort coincides with *extensional equality*.

The purpose of ground normal forms is to represent data. Intuitively, visible normal forms are unique data representations, hidden ones are not because the identity of a hidden object is determined by some

[2] We use "$\equiv$" for distinguishing the *symbol* for structural equality from semantical identity, which is denoted by "$=$".

behavioral equality. A hidden normal form is just a *name* of an object, although the structure of the name often represents the object's "history" or "vita".

Let $\Sigma = (S, F, P)$ and $\Sigma' = (S', F', P')$ be signatures and X be an S-sorted set of variables. A **signature morphism** $\sigma : \Sigma \to \Sigma'$ consists of a function $\sigma_{sorts} : S \to S'$ and S^+-sorted sets of functions $\sigma_{functs} = \{\sigma_w : F_w \to (F')^*_{\sigma(w)}\}$ and $\sigma_{preds} = \{\sigma_w : P_w \to (P')^*_{\sigma(w)}\}$ such that for all $f : w \to s \in F$, $\sigma(f) : \sigma(w) \to \sigma(s)$ and for all $r : w \in P$, $\sigma(r) : \sigma(w)$.

Given a parameterized specification SP with parameter $PAR = (\Sigma, AX)$ and a signature morphism $\sigma : \Sigma \to \Sigma'$, let $domain(\sigma) =_{def} \{s \in \Sigma \mid \sigma(s) \neq s\} = \{s_1, \ldots, s_n\}$. The specification $SP[\sigma]$, usually written as

$$SP[s_1 \mapsto s'_1, \ldots, s_n \mapsto s'_n],$$

is called the **actualization of** SP **along** σ and obtained from SP by replacing all (!) occurrences in SP of $s \in domain(\sigma)$ by $\sigma(s)$ and by deleting the axioms of PAR.

An S-sorted function $\sigma : X \to T_\Sigma(X)$ is called a **substitution**. The **domain of** σ, $dom(\sigma)$, is the set of all variables x with $x\sigma \neq x$. σ_Y denotes the restriction of σ that is defined by $x\sigma_Y = x\sigma$ for all $x \in Y$ and $x\sigma_Y = x$ for all $x \in X \setminus Y$. If σ maps each variable of $dom(\sigma)$ to a term in some given set T of terms, we write $\sigma : X \to T$ in order to indicate that σ satisfies $\sigma(dom(X)) \subseteq T$. The **instance** $t\sigma$ of a term or atom t **by** σ is obtained from t by replacing each variable x by $x\sigma$.

Definition 2.3 (Σ-formulas) A formula φ with a single free variable is **unary**. A **Σ-goal** is a finite conjunction of Σ-atoms. Given a finite subset Y of X and goals G and H, the formula $\exists Y G$ is an **existential goal** and the formula $\forall Y(G \Rightarrow H)$ is a **universal goal**. A **goal set** is a finite disjunction of existential goals. A **dual goal set** is a finite conjunction of universal goals. The empty conjunction is called the **empty goal** and denoted by $\emptyset$ or $TRUE$. The empty disjunction is denoted by $FALSE$.

Let G be a goal, r be a logical predicate and f be a defined function. A formula of the form $r(t) \Leftarrow G$ resp. $f(t) \equiv u \Leftarrow G$ is a **Horn clause for** r resp. f. Given a finite disjunction φ of existential goals, $r(t) \Rightarrow (G \Rightarrow \varphi)$ is a **co-Horn clause for** r. Given a finite conjunction φ of universal goals, $r(t) \Leftarrow \varphi$ is a **generalized Horn clause for** r. The formulas $\varphi \Leftarrow TRUE$ and $TRUE \Rightarrow \varphi$ are identified with φ.

Suppose that Σ is swinging. Let $\mathcal{C}$ be a class of Σ-structures. A formula φ is (first-order) **modal in** $\mathcal{C}$ if φ is equivalent[3] in $\mathcal{C}$ to a formula built up by the following rules:

- A unary static atom is modal.
- If φ and ψ are modal, then $\neg\varphi$ and $\varphi \wedge \psi$ are modal.
- If t is a unary term, φ is modal, $y \in free(\varphi) \setminus var(t)$ and $\delta(t, y)$ is a dynamic atom, then $\exists y(\delta(t, y) \wedge \varphi)$ is modal.

A formula φ is **poly-modal** if φ is equivalent to a formula built up by the following rules:

- A static atom is poly-modal.
- If φ and ψ are poly-modal, then $\neg\varphi$, $\varphi \wedge \psi$ and for all $x \in X$, $\exists x\varphi$ are poly-modal.
- If $\delta(t, x)$ with $x \in X \setminus var(t)$ is a dynamic atom and φ is poly-modal, then $\exists x(\delta(t, x) \wedge \varphi)$ is poly-modal.

A formula φ is **weakly modal with output out**$(\varphi) \subseteq X$ if φ is equivalent to a formula built up by the following rules:

[3] See Def. 3.1.

- A poly-modal formula is weakly modal with output $\emptyset$.
- A dynamic atom $\delta(t,x)$ with $x \in X \setminus var(t)$ is weakly modal with output $\{x\}$.
- If φ and ψ are weakly modal with disjoint outputs Y resp. Z, then $\varphi \wedge \psi$ is weakly modal with output $Y \cup Z$.
- If φ is weakly modal with output Y, then for all $x \in X$, $\exists x\varphi$ is weakly modal with output $Y \setminus \{x\}$. ❑

Modal formulas arise from the translation of modal into predicate logic. Given a transition relation $\rightarrow$ and propositions p representing state sets, assertions of the form "the state x satisfies the modal-logic formula φ" can be compiled into modal formulas in the sense of Def. 2.3 as follows:

$$
\begin{array}{lcl}
compile(x \models p) & = & r_p(x) \quad \text{for all propositions } p \\
compile(x \models \varphi \wedge \psi) & = & compile(x \models \varphi) \wedge compile(x \models \psi) \\
compile(x \models \varphi \vee \psi) & = & compile(x \models \varphi) \vee compile(x \models \psi) \\
compile(x \models \langle . \rangle \varphi) & = & \exists y(x \rightarrow y \wedge compile(y \models \varphi)) \\
compile(x \models [.]\varphi) & = & \forall y(x \rightarrow y \Rightarrow compile(y \models \varphi)) \\
compile(x \models \mu p.(\varphi_1 \vee \cdots \vee \varphi_n)) & = & r_p(x) \;\text{ where } r_p \text{ is specified by the axioms} \\
 & & \quad r_p(x) \Leftarrow compile(x \models \varphi_1),\ \ldots,\ r_p(x) \Leftarrow compile(x \models \varphi_n) \\
compile(x \models \nu p.(\varphi_1 \wedge \cdots \wedge \varphi_n)) & = & r_p(x) \;\text{ where } r_p \text{ is specified by the axioms} \\
 & & \quad r_p(x) \Rightarrow compile(x \models \varphi_1),\ \ldots,\ r_p(x) \Rightarrow compile(x \models \varphi_n)
\end{array}
$$

Other negation-free modal-logic formulas are equivalent to those compiled here (see Ex. 2.7).

Modal formulas are poly-modal. Poly-modal formulas are weakly modal. The output of a weakly modal formula consists of *free* variables.

Conjectures may be arbitrary first-order formulas. Axioms will be restricted to Horn and co-Horn clauses. This complies with usual syntax adopted by functional, relational and even state- or object-oriented programs. Semantically, the restriction to Horn and co-Horn axioms is the main prerequisite for the existence of standard models such as Herbrand, initial and final models and thus of "concrete" implementations. Standard models also enjoy a number of "meta-theorems", which equip program verifiers with indispensable "background knowledge".

A swinging signature Σ is implicitly associated with the set $\mathbf{EQ}_\Sigma$ of **congruence axioms for** Σ, given by the Horn resp. co-Horn clauses:

$$
\begin{array}{l}
x \equiv x \\
y \equiv x \quad \Leftarrow \quad x \equiv y \\
f(x_1,\ldots,x_n) \equiv f(y_1,\ldots,y_n) \quad \Leftarrow \quad x_1 \equiv y_1 \wedge \cdots \wedge x_n \equiv y_n \\
r(x_1,\ldots,x_n) \quad \Leftarrow \quad x_1 \equiv y_1 \wedge \cdots \wedge x_n \equiv y_n \wedge r(y_1,\ldots,y_n) \\
q(x_1,\ldots,x_n) \quad \Rightarrow \quad ((x_1 \equiv y_1 \wedge \cdots \wedge x_n \equiv y_n) \;\Rightarrow\; q(y_1,\ldots,y_n))
\end{array}
$$

for all function symbols f, μ-predicates r and ν-predicates q of Σ.

Definition 2.4 A **specification** $SP = (\Sigma, AX)$ consists of a signature Σ and a set AX of first-order Σ-formulas, called the **axioms** of SP. SP is **swinging** if Σ is swinging and SP has three subspecifications

$$visSP = (vis\Sigma, visAX) \subseteq hidSP = (hid\Sigma, hidAX) \subseteq \nu SP = (\nu\Sigma, \nu AX) \subseteq SP$$

such that $hidAX$ and νAX implicitly include the Horn resp. co-Horn clauses among the congruence axioms for Σ and the following conditions hold true:

(1)	**The visible level** $visSP$
	consists of visible sorts and visible constructors, a set DF of defined functions, a set P of static μ-predicates, Horn axioms $f(t) \equiv u \Leftarrow \varphi$ for DF and $r(t) \Leftarrow \varphi$ for P such that (a) r is logical, t is a tuple of normal forms and $var(u) \subseteq var(t, \varphi)$.
(2)	**The hidden level** $hidSP \setminus visSP$
	consists of hidden sorts and hidden constructors, a set DF of defined functions, a set P of static μ-predicates, a set DP of dynamic predicates and Horn axioms $f(t) \equiv u \Leftarrow \varphi$ for DF, $r(t) \Leftarrow \varphi$ for P and $\delta(t, u) \Leftarrow \varphi$ for DP such that (a) holds true and (b) φ is weakly modal such that $var(t) \cap out(\varphi) = \emptyset$.
(3)	**The ν-level** $\nu SP \setminus hidSP$
	consists of a set P of ν-predicates (including the behavioral equalities) and co-Horn axioms $r(t) \Rightarrow (G \Rightarrow \varphi)$ for P such that (a) holds true, $G \Rightarrow \varphi$ is poly-modal and G is a goal over $hidSP$. The axioms for behavioral equalities are called **behavior axioms** and read as follows: $x \sim_s y \;\Rightarrow\; x \equiv y$ for all visible sorts $s \in \Sigma$, $x \sim_s y \;\Rightarrow\; f(x, z) \sim_{s'} f(y, z)$ for all destructors $f : sw \to s' \in \Sigma$, $x \sim_s y \;\Rightarrow\; (r(x, z) \;\Rightarrow\; r(y, z))$ and $x \sim_s y \;\Rightarrow\; (r(y, z) \;\Rightarrow\; r(x, z))$ for all separators $r : sw \in \Sigma$, $x \sim_s y \;\Rightarrow\; (\delta(x, z, x') \;\Rightarrow\; \exists y'(\delta(y, z, y') \wedge x' \sim_{s'} y'))$ and $x \sim_s y \;\Rightarrow\; (\delta(y, z, y') \;\Rightarrow\; \exists x'(\delta(x, z, x') \wedge x' \sim_{s'} y'))$ for all transition predicates $\delta : sws' \in \Sigma$.
(4)	**The μ-level** $SP \setminus \nu SP$
	consists of a set P of static μ-predicates, a set DP of dynamic predicates and generalized Horn axioms $r(t) \Leftarrow \varphi$ for P and $\delta(t, u) \Leftarrow \varphi$ for DP such that (a) and (b) hold true and for all universal goals $\forall Y(G \Rightarrow H)$ of φ, G is a goal over νSP.

If $SP = visSP$, then SP is **visible**. ❏

Together with the condition of Def. 2.2 that all hidden constructors have hidden ranges the levels of a swinging specification entail a hierarchy of their Herbrand models (cf. Lemma 5.9). Excluding hidden constructors with visible ranges is also motivated intuitively by the viewpoint that objects with hidden components cannot be visible. Hidden constructors with visible ranges represent certain "contexts" and thus are better modelled by observers that make contexts visible. Visible normal forms t of a functional specification are uniquely decomposable: all ground normal forms that are equivalent to a ground instance of t are themselves ground instances of t. Hidden normal forms enjoying the same property with respect to *behavioral* equivalence are *strongly normal* (cf. Def. 6.1). For instance, the stream term $x\&s$ (cf. Ex. 2.8) is strongly normal. A hidden constructor $c : w \to s$ with visible range s can be replaced by a constructor $c' : w \to s'$ such that s' is hidden and $c(t)$ is uniquely decomposable w.r.t. structural s-equivalence iff $c'(t)$ is uniquely decomposable w.r.t. behavioral s'-equivalence.

Condition 2.4(a) reflects the usual syntax of functional-logic programs. It also admits a simple proof that SP is complete (cf. Def. 4.1). If "definedness predicates" $Def : s$ are specified by a Horn axiom

$$Def(c(x_1, \ldots, x_n)) \quad \Leftarrow \quad Def(x_1) \wedge \cdots \wedge Def(x_n)$$

for each constructor c, then SP is complete iff for all defined functions f, $Def(x) \Rightarrow Def(f(x))$ is an inductive theorem of SP (cf. Def. 4.6). Moreover, 2.4(a) is an essential part of most confluence and consistency criteria, such as [60], 10.46 und 10.48. 2.4(a) also ensures that basic deduction rules such as *unfolding* are sound (see Section 4).

The modality assumptions on the axioms of the hidden, μ- and ν-level of SP are essential for the behavioral consistency of SP-models (cf. Def. 3.1). They restrict the occurrences of dynamic predicates in the axioms, but this restriction is much weaker than previous similarly motivated conditions such as the non-existence of hidden equations in Horn axiom premises (cf., e.g., [18], Cor. 4; [77], Thm. 5.4.5; [17], Ex. 3.24). Condition 2.4(b) also reveals the technical reason for the distinction between static and dynamic predicates. While static predicates can often be transformed easily into Boolean functions because all their arguments have a sort of "input mode", a dynamic predicate has at least one argument (usually the last), which takes up output that is produced when an axiom for the predicate is "called". In fact, a static predicate r may also have output arguments, provided that these are not produced by a dynamic predicate δ in the premise of an axiom for r. For instance, an axiom of the form $r(t, u(x)) \Leftarrow \delta(v, x)$ satisfies 2.4(b) only if r is dynamic.

Since the behavior axioms are completely determined by the observers, they are omitted in examples. The separation of the ν-level from the μ-level prevents a μ-predicate and a ν-predicate from being specified in a mutually-recursive way. Such *alternating fixpoints* were difficult to handle and are actually not needed in practice, even for specifying modal operators (cf. Ex. 2.7). The hierarchy assumption in 2.4(3): G is a goal over $hidSP$, and the corresponding one in 2.4(4): G is a goal over νP, are essential for the monotonicity of the consequence operators that build up the Herbrand model (cf. Lemma 4.4).

Example 2.5 The ubiquitious stack data type is presented as a visible swinging specification.

ENTRY

`sorts`	$entry$
`preds`	$\_ \not\equiv \_ : entry \times entry$
`vars`	$x, y : entry$
`axioms`	$x \not\equiv y \Longleftrightarrow \neg(x \equiv y)$

STACK = ENTRY `then`

`vissorts`	$stack \quad entry'$
`constructs`	$def : entry \rightarrow entry'$
	$\bot : \rightarrow entry'$
	$empty : \rightarrow stack$
	$push : entry \times stack \rightarrow stack$
`defuncts`	$pop : stack \rightarrow stack$
	$top : stack \rightarrow entry'$
`vars`	$x : entry \quad s : stack$
`Horn axioms`	$top(empty) \equiv \bot$
	$pop(empty) \equiv empty$
	$top(push(x, s)) \equiv def(x)$
	$pop(push(x, s)) \equiv s$

For specifying a partial function f such as top the original range sort of f (here $entry$) is embedded into a sum sort (here $entry'$) that includes "exceptions" (here $\bot$) and thus totalizes f. In a later design step, $entry'$ may be refined to a hidden sort and structural $entry'$-equality may be implemented as a behavioral equality so that the single exeception $\bot$ can be splitted into several more informative error messages (see [64]). ❑

Example 2.6 The first specification (FLAG1) of a type of flags stems from [29]. two examples illustrate the use of destructors versus separators. While FLAG1 is purely functional and specifies behavioral equivalence in terms of destructors, FLAG2 adopts the relational view and thus uses a separator for determining behavioral equivalence.

FLAG1 (cf. [29])

`hidsorts`	$flag$
`constructs`	$new :\to flag$
	$up, down, rev : flag \to flag$
`destructs`	$up? : flag \to bool$
`vars`	$b : bool \quad x : flag$
`Horn axioms`	$up?(new) \equiv true$
	$up?(up(x)) \equiv true$
	$up?(down(x)) \equiv false$
	$up?(rev(x)) \equiv not(up?(x))$

FLAG2

`hidsorts`	$flag$
`constructs`	$new :\to flag$
	$up, down, rev : flag \to flag$
`separators`	$up?, down? : flag$
`vars`	$x : flag$
`Horn axioms`	$up?(new)$
	$up?(up(x))$
	$down?(down(x))$
	$up?(rev(x)) \Leftarrow down?(x)$
	$down?(rev(x)) \Leftarrow up?(x)$ ❏

Example 2.7 It is well-known that all modal operators associated with classical modal logics such as CTL (cf. [27]) or the μ-calculus (cf. [73]) are least or greatest fixpoints of state set functions. Co-Horn axioms are sufficient for specifying greatest fixpoints, (generalized) Horn axioms are a suitable syntax for least fixpoints. Hence (instances of) modal operators yield typical predicates of the μ- or ν-level of a swinging specification involving transition systems:

STATE

`vissorts`	$action_1, \ldots, action_n$	
`hidsorts`	$state$	
`static` μ`-preds`	$q, r : state$	
`transpreds`	$\_ \stackrel{\_}{\longrightarrow}_i \_ : state \times action_i \times state$	$1 \le i \le n$
	$\ldots$	
`Horn axioms`	$\ldots$	

MODSPEC = STATE `then`

`dynamic preds`	$\_ \longrightarrow_i \_ : state \times state$	$1 \le i \le n$
	$\_ \longrightarrow \_ : state \times state$	
`static` μ`-preds`	$enabled : state$	the actual state has a direct successor in the graph of $\longrightarrow$
	$\langle . \rangle r : state$	r holds true in some direct successor
	$EF(r) : state$	r "exists finally" (also written $\Diamond r$)
	$E(q\ U\ r) : state$	on some path (starting out from the actual state), q holds true until r is valid and r becomes valid eventually
ν`-preds`	$disabled : state$	the actual state has no direct successor
	$[.]r : state$	r holds in all direct successor states
	$AG(r) : state$	r "always generally" (also written $\Box r$)
	$E_\infty G(r) : state$	r "exists generally" on infinite paths
	$EG(r) : state$	r "exists generally"
	$E(q \leadsto r) : state$	on some path, q leads to r
	$E(q\ wU\ r) : state$	on some path, q holds until r becomes valid
	$A(q\ wU\ r) : state$	on all paths, q holds until r becomes valid

	$\mathit{some\_infinite} : \mathit{state}$	some path starting out from the actual state is infinite
`static` μ`-preds`	$A_\infty F(r) : \mathit{state}$	r "always finally" on infinite paths
	$AF(r) : \mathit{state}$	r "always finally"
	$A(q \rightsquigarrow r) : \mathit{state}$	on all paths, q leads to r
	$A(q\ U\ r) : \mathit{state}$	on all paths, q holds true until r is valid and r becomes valid eventually
	$\mathit{all\_finite} : \mathit{state}$	all paths starting out from the actual state are finite
`vars`	$a : \mathit{action}_i \quad s, s' : \mathit{state}$	$1 \leq i \leq n$
`Horn axioms`	$s \longrightarrow_i s' \quad \Leftarrow \quad s \stackrel{a}{\longrightarrow}_i s'$	
	$s \longrightarrow s' \quad \Leftarrow \quad s \longrightarrow_i s'$	
	$\mathit{enabled}(s) \quad \Leftarrow \quad s \longrightarrow s'$	
	$\langle . \rangle r(s) \quad \Leftarrow \quad s \longrightarrow s' \ \wedge \ r(s')$	
	$EF(r)(s) \quad \Leftarrow \quad r(s)$	
	$EF(r)(s) \quad \Leftarrow \quad s \longrightarrow s' \ \wedge \ EF(r)(s')$	
	$EG(r)(s) \quad \Leftarrow \quad \mathit{disabled}(s) \ \wedge \ r(s)$	
	$EG(r)(s) \quad \Leftarrow \quad s \longrightarrow s' \ \wedge \ r(s) \ \wedge \ EG(r)(s')$	
	$E(q \rightsquigarrow r)(s) \quad \Leftarrow \quad q(s) \ \wedge \ EF(r)(s)$	
	$E(q \rightsquigarrow r)(s) \quad \Leftarrow \quad s \longrightarrow s' \ \wedge \ E(q \rightsquigarrow r)(s')$	
	$E(q\ U\ r)(s) \quad \Leftarrow \quad r(s)$	
	$E(q\ U\ r)(s) \quad \Leftarrow \quad q(s) \ \wedge \ s \longrightarrow s' \ \wedge \ E(q\ U\ r)(s')$	
	$\mathit{some\_infinite}(s) \quad \Leftarrow \quad EG(\mathit{enabled})(s)$	
`co-Horn axioms`	$\mathit{disabled}(s) \quad \Rightarrow \quad (s \longrightarrow s' \ \Rightarrow \ \mathit{FALSE})$	
	$[.]r(s) \quad \Rightarrow \quad (s \longrightarrow s' \ \Rightarrow \ r(s'))$	
	$AG(r)(s) \quad \Rightarrow \quad r(s)$	
	$AG(r)(s) \quad \Rightarrow \quad (s \longrightarrow s' \ \Rightarrow \ AG(r)(s'))$	
	$E_\infty G(r)(s) \quad \Rightarrow \quad r(s)$	
	$E_\infty G(r)(s) \quad \Rightarrow \quad \exists s'(s \longrightarrow s' \wedge E_\infty G(r)(s'))$	
	* $EG(r)(s) \quad \Rightarrow \quad r(s)$	
	* $EG(r)(s) \quad \Rightarrow \quad (s \longrightarrow s' \ \Rightarrow \ \exists s'(s \longrightarrow s' \wedge EG(r)(s'))$	
	$AF(r)(s) \quad \Rightarrow \quad (\mathit{disabled}(s) \ \Rightarrow \ r(s))$	
	$AF(r)(s) \quad \Rightarrow \quad (s \longrightarrow s' \ \Rightarrow \ (r(s) \ \vee \ AF(r)(s')))$	
	$A(q \rightsquigarrow r)(s) \quad \Rightarrow \quad (q(s) \ \Rightarrow \ AF(r)(s))$	
	$A(q \rightsquigarrow r)(s) \quad \Rightarrow \quad (s \longrightarrow s' \ \Rightarrow \ A(q \rightsquigarrow r)(s'))$	
	$A(q\ wU\ r)(s) \quad \Rightarrow \quad (q(s) \ \vee \ r(s))$	
	$A(q\ wU\ r)(s) \quad \Rightarrow \quad (s \longrightarrow s' \ \Rightarrow \ (r(s) \ \vee \ (q(s) \ \wedge \ A(q\ wU\ r)(s'))))$	
	$E(q\ wU\ r)(s) \quad \Rightarrow \quad (q(s) \ \vee \ r(s))$	
	$E(q\ wU\ r)(s) \quad \Rightarrow \quad (\mathit{enabled}(s) \ \Rightarrow \ (r(s) \ \vee \ \exists s'(s \longrightarrow s' \ \wedge \ E(q\ wU\ r)(s'))))$	
	* $\mathit{some\_infinite}(s) \quad \Rightarrow \quad \exists s'(s \longrightarrow s' \wedge \mathit{some\_infinite}(s'))$	
`generalized Horn axioms`		
	$A_\infty F(r)(s) \quad \Leftarrow \quad r(s)$	
	$A_\infty F(r)(s) \quad \Leftarrow \quad \forall s'(s \longrightarrow s' \Rightarrow A_\infty F(r)(s'))$	
	* $AF(r)(s) \quad \Leftarrow \quad r(s)$	
	* $AF(r)(s) \quad \Leftarrow \quad s \longrightarrow s' \ \wedge \ \forall s'(s \longrightarrow s' \Rightarrow AF(r)(s'))$	
	$A(q\ U\ r)(s) \quad \Leftarrow \quad r(s)$	
	$A(q\ U\ r)(s) \quad \Leftarrow \quad q(s) \ \wedge \ \mathit{enabled}(s) \ \wedge \ \forall s'(s \longrightarrow s' \Rightarrow A(q\ U\ r)(s'))$	
	* $A(q\ U\ r)(s) \quad \Leftarrow \quad A(q\ wU\ r)(s) \ \wedge \ AF(r)(s)$	

$$
\begin{array}{lll}
* \; E(q \; wU \; r)(s) & \Leftarrow & E(q \; U \; r)(s) \\
* \; E(q \; wU \; r)(s) & \Leftarrow & EG(q)(s) \\
\mathit{all\_finite}(s) & \Leftarrow & AF(\mathit{disabled})(s) \\
* \; \mathit{all\_finite}(s) & \Leftarrow & \forall s'(s \longrightarrow s' \Rightarrow \mathit{all\_finite}(s'))
\end{array}
$$

Most of these formulas are derived form *temporal propositions* insofar as they quantify over finite or infinite *runs* (= paths in the graph of $\longrightarrow$). E-formulas quantify existentially. A-formulas quantify universally. Formulas preceded by an asterisk (*) provide alternative axioms for the specified predicates. ❑

Example 2.8 Let NAT be a specification of natural number arithmetic. For LISTORD see Ex. 2.1. A swinging specification of infinite sequences of *entry*-elements reads as follows:

INFSEQ = LISTORD `and` NAT `then`

`hidsorts`	$stream = stream(entry)$	
`constructs`	$\_\&\_ : entry \times stream \to stream$	
	$blink : \to stream(nat)$	
	$nats : nat \to stream$	
	$odds : stream \to stream$	
	$zip : stream \times stream \to stream$	
	$map : (entry \to entry) \times stream \to stream$	
`destructs`	$head : stream \to entry$	
	$tail : stream \to stream$	
`defuncts`	$\_\#\_ : list \times stream \to stream$	
	$evens : stream \to stream$	
	$firstn : nat \times stream \to list$	
	$nthtail : nat \times stream \to stream$	
`static` μ`-preds`	$exists : (entry \to bool) \times stream$	
ν`-preds`	$forall : (entry \to bool) \times stream$	
	$fair : (entry \to bool) \times stream$	
`vars`	$n : nat \quad x, y : entry \quad L : list \quad s, s' : stream$	
	$f : entry \to entry \quad g : entry \to bool$	
`Horn axioms`	$head(x\&s) \equiv x$	$tail(x\&s) \equiv s$
	$head(blink) \equiv 0$	$tail(blink) \equiv 1\&blink$
	$head(nats(n)) \equiv n$	$tail(nats(n)) \equiv nats(n+1)$
	$head(zip(s, s')) \equiv head(s)$	$tail(zip(s, s')) \equiv zip(s', tail(s))$
	$head(odds(s)) \equiv head(s)$	$tail(odds(s)) \equiv odds(tail(tail(s)))$
	$head(map(f, s)) \equiv f(s)$	$tail(map(f, s)) \equiv map(f, tail(s))$
	$nil\#s \equiv s$	
	$(x :: L)\#s \equiv x\&(L\#s)$	
	$evens(s) \equiv odds(tail(s))$	
	$firstn(0, s) \equiv nil$	
	$firstn(n+1, s) \equiv head(s) :: firstn(n, tail(s))$	
	$nthtail(0, s) \equiv s$	
	$nthtail(n+1, s) \equiv nthtail(n, tail(s))$	
	$exists(g, s) \;\Leftarrow\; g(head(s)) \equiv true$	
	$exists(g, s) \;\Leftarrow\; exists(g, tail(s))$	
`co-Horn axioms`	$forall(g, s) \;\Rightarrow\; g(head(s)) \equiv true \wedge forall(g, tail(s))$	

$$fair(g, s) \Rightarrow exists(g, s) \wedge fair(g, tail(s))$$

The following should hold in a standard model of INFSEQ. & 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. $odds(s)$ returns the stream of all elements of s that have odd-numbered positions in s. *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. *head*, *tail*, *firstn*, *nthtail*, *map*, *exists* and *forall* have the same meaning as stream functions as they have as list functions. $fair(g, s)$ holds true iff g holds true for infinitely many elements of s. ❏

3 Structures and congruences

Definition 3.1 (semantical notions) Let $\Sigma = (S, F, P)$ be a signature. A **Σ-structure** A consists of an S-sorted set, the **carrier** of A, also denoted by A, for all $f : w \to s \in F$, a function $f^A : A_w \to A_s$, and for all $r : w \in P$, a relation $r^A \subseteq A_w$. $\overline{r} : w \in P$ is called the **complement of** r **w.r.t.** A if $\overline{r}^A = A_w \setminus r^A$. If P is empty, A is called a **Σ-algebra**. A is a **Herbrand structure** if for all $s \in S$, $A_s = T_{\Sigma,s}$, and for all $f : w \to s \in F$ and $t \in T_\Sigma(X)_w$, $f^A(t) = f(t)$.

Given $\approx_s : ss \in P$ for all $s \in S$, A is a **structure with $\approx$-equality** if for all $s \in S$, $\approx_s^A = \{(a, a) \mid a \in A_s\}$. A Σ-structure B is **monotone w.r.t.** A if

- for all ground static μ-atoms p, $A \models p$ implies $B \models p$,
- for all ground dynamic atoms $\delta(t, u)$, $A \models \delta(t, u)$ implies $B \models \delta(t, v)$ for some $v \in T_{\Sigma,s}$ with $v^A = u^A$,
- for all ground ν-atoms q, $B \models q$ implies $A \models q$.

A $\mu\nu\Sigma$- resp. **$\nu\mu\Sigma$-homomorphism** $h : A \to B$ is an S-sorted function such that for all $f : w \to s \in F$, $h_s \circ f^A = f^B \circ h_w$, for all $r \in \mu\Sigma$ resp. $r \in \nu\Sigma$, $h(r^A) \subseteq r^B$, and for all $r \in \nu\Sigma$ resp. $r \in \mu\Sigma$, $r^B \subseteq h(r^A)$. h is a **Σ-isomorphism** if there is a $\mu\nu\Sigma$- resp. $\nu\mu\Sigma$-homomorphism $g : B \to A$ such that $g \circ h = id_A$ and $h \circ g = id_B$. A and B are **Σ-isomorphic** iff there is a Σ-isomorphism $h : A \to B$.

The interpretation of Σ-terms in a Σ-structure A depends on a (first-order) **valuation** of variables **in** A, i.e. an S-sorted function $b : X \to A$. Given a further valuation $c : X \to A$ and $Y \subseteq X$, we write $b =_Y c$ if $b(x) = c(x)$ for all $x \in X \setminus Y$. Given $x \in X$ and $a \in A$, **b[a/x]**: $X \to A$ is defined by $b[a/x](x) = a$ and $b[a/x] =_x b$. a/x denotes $b[a/x]$ for any b. b extends to a function $b^* : T_\Sigma(X) \to A$ defined by $b^*(x) = b(x)$ for all $x \in X$ and $b^*(t) = f^A(b^*(t_1), \ldots, b^*(t_n))$ for all $t = f(t_1, \ldots, t_n) \in T_\Sigma(X)$. Given a term t with $var(t) = \{x_1, \ldots, x_n\}$, we sometimes use the function $t^A : A^n \to A$ defined by $t^A(b(x_1), \ldots, b(x_n)) = b^*(t)$. A is **reachable** if for all $a \in A$ there is $t \in T_\Sigma$ with $t^A = a$.[4]

A valuation $b : X \to A$ **solves** an atom $r(t)$ in A if $b^*(t) \in r^A$. This notion extends to first-order formulas as usual. If b solves φ in A, we write $A \models_b \varphi$. A **satisfies** or is a **model of** φ or φ is **valid** in A, written $A \models \varphi$, if all valuations in A solve φ in A. A class $\mathcal{C}$ of Σ-structures satisfies φ iff all $A \in \mathcal{C}$ satisfy φ. Two Σ-formulas φ and ψ are **equivalent in** a class $\mathcal{C}$ of Σ-structures if $\mathcal{C}$ satisfies $\varphi \Leftrightarrow \psi$. Two Σ-formulas are **equivalent** if they are equivalent in all Σ-structures.

Let Σ be swinging. An S-sorted binary relation $\approx \subseteq A \times B$ is a **Σ-bisimulation** if

[4] Each Σ-structure has a least reachable Σ-substructure (with respect to the inclusion of carriers).

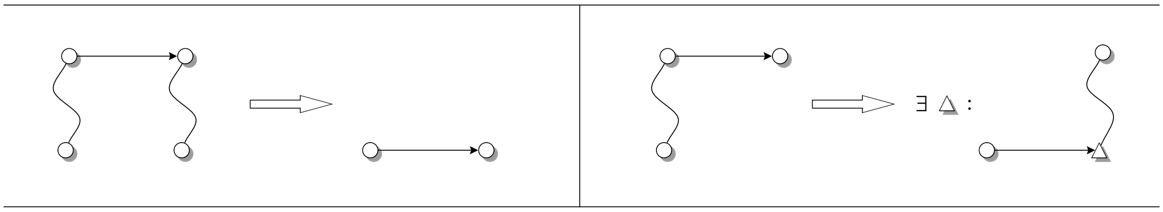

Figure 1. *Compatibility versus zigzag compatibility of $\sim$ with $\longrightarrow$*

- for all $f : w \to s \in F$, $1 \leq i \leq n$, $a \in A_{s_i}$, $b \in B_{s_i}$ and $t_j \in T_{\Sigma,s_j}$, $1 \leq j \neq i \leq n$, $a \approx b$ implies

$$f^A(t_1^A, \ldots, a, \ldots, t_n^A) \approx f^B(t_1^B, \ldots, b, \ldots, t_n^B),$$

- for all $r : s_1 \ldots s_n \in statP$, $1 \leq i \leq n$, $a \in A_{s_i}$, $b \in B_{s_i}$ and $t_j \in T_{\Sigma,s_j}$, $1 \leq j \neq i \leq n$, $a \approx b$ implies

$$(t_1^A, \ldots, a, \ldots, t_n^A) \in r^A \quad \text{iff} \quad (t_1^B, \ldots, b, \ldots, t_n^B) \in r^B,$$

- for all $\delta : s_1 \ldots s_n s \in dynP$, $1 \leq i \leq n$, $a \in A_{s_i}$, $a' \in A_s$, $b \in B_{s_i}$, $b' \in B_s$ and $t_j \in T_{\Sigma,s_j}$, $1 \leq j \neq i \leq n$,

$$\begin{array}{lll}(t_1^A, \ldots, a, \ldots, t_n^A, a') \in \delta^A \ \wedge \ a \approx b & \text{implies} & \exists\, b' \in B : (t_1^B, \ldots, b, \ldots, t_n^B, b') \in \delta^B \ \wedge \ a' \approx b', \\ (t_1^B, \ldots, b, \ldots, t_n^B, b') \in \delta^B \ \wedge \ a \approx b & \text{implies} & \exists\, a' \in A : (t_1^A, \ldots, a, \ldots, t_n^A, a') \in \delta^A \ \wedge \ a' \approx b'.\end{array}$$

$\approx$ is **compatible** with $f : w \to s \in F$ if for all $a \in A_w$ and $b \in B_w$, $a \approx b$ implies $f^A(a) \approx f^B(b)$. $\approx$ is **compatible** with $r : w \in P$ if for all $a \in A_w$ and $b \in B_w$, $a \approx b$ implies $a \in r^A$ iff $b \in r^B$. $\approx$ is **zigzag compatible** with $\delta : ws \in P$ if for all $(a, a') \in \delta^A$, $a \approx b$ implies $(b, b') \in \delta^B$ for some $b' \in B_s$ with $a' \approx b'$ and for all $(b, b') \in \delta^B$, $a \approx b$ implies $(a, a') \in \delta^A$ for some $a' \in A_s$ with $a' \approx b'$.

A first-order formula φ is **bisimulation invariant** in a class $\mathcal{C}$ of Σ-structures if for all $A, B \in \mathcal{C}$, bisimulations $\approx \subseteq A \times B$, $b : X \to A$ and $c : X \to B$, $b \approx c$ implies $A \models_b \varphi$ iff $B \models_c \varphi$.

An S-sorted equivalence relation $\approx \subseteq A \times A$ is a **Σ-congruence on** A if $\approx$ is compatible with $F \cup P$. $\approx$ is a **weak Σ-congruence on** A if $\approx$ is compatible with $F \cup statP$ and zigzag compatible with $dynP$.

Let $\approx$ be a (behavioral) Σ-congruence on A. Then the **quotient** $B = A/\!\approx$ **of** A **by** $\approx$ is the Σ-structure that interprets $s \in S$ as the quotient set $A_s/\!\approx$ and $f : w \to s \in F$ as the function $f^B : B_w \to B$ defined by $f^B([a]) = [f^A(a)]$ where $[a]$ denotes the **equivalence class** of a consisting of all $b \in A$ with $a \approx b$.[5] **nat**: $A \to B$ denotes the **natural mapping** that sends an element a to its equivalence class $[a]$.

If $\approx$ is a congruence, then B interprets $r : w \in P$ as the set of $[a] \in B_w$ with $a \in r^A$. If $\approx$ is a weak congruence, this definition is restricted to static predicates, while for all dynamic predicates $\delta : ws \in P$,

$$([a], [b]) \in \delta^B \quad \Longleftrightarrow_{def} \quad \exists\, b' \approx b : (a, b') \in \delta^A.$$

Let $SP = (\Sigma, AX)$ be a (swinging) specification. A Σ-structure A is an **SP-model** if A satisfies AX and EQ_Σ. A is **behaviorally SP-consistent** if $\sim^A$ is a weak Σ-congruence and $A/\!\sim^A$ is an SP-model.[6] **Mod(SP)** is the class of all SP-models. $\mathbf{Mod}_\equiv\mathbf{(SP)}$ is the class of SP-models with $\equiv$-equality. $\mathbf{Mod}_{be}\mathbf{(SP)}$ is the class of SP-models A such that $\sim^A$ is an equivalence relation that includes $\equiv^A$. $\mathbf{Mod}_{bc}\mathbf{(SP)}$ is the class of SP-models A such that $\sim^A$ is a weak Σ-congruence. $\mathbf{Mod}_{bcr}\mathbf{(SP)}$ is the class of reachable elements

[5] If $a = (a_1, \ldots, a_n)$, then $[a]$ stands for $([a_1], \ldots, [a_n])$.
[6] In [22], a function that is compatible with $\sim^A$ is called *behaviorally coherent*.

of $Mod_{bc}(SP)$. $\mathbf{Mod_{\mu\nu}(SP)}$ is the class of SP-models A that interpret all μ-predicates as the least relations and all ν-predicates as the greatest relations satisfying AX.

Given a signature morphism $\sigma : \Sigma \to \Sigma'$ and a Σ'-structure A, the **σ-reduct $A|_\sigma$ of A** is the Σ-structure defined by $(A|_\sigma)_s = A_{\sigma(s)}$ for all $s \in S$ and $f^{A|_\sigma} = \sigma(f)^A$ for all $f \in F \cup P$. The least reachable Σ-substructure of $A|_\sigma$ is denoted by A_σ. If σ is an inclusion, i.e. $\Sigma \subseteq \Sigma'$, we write $A|_\Sigma$ instead of $A|_\sigma$ and A_Σ instead of A_σ and call $A|_\Sigma$ the **Σ-reduct of A**. ❑

Σ-congruences are weak Σ-congruences because the latter are reflexive. The difference between congruence and weak congruence becomes clear if one transforms a static predicate $r : w$ and a dynamic predicate $\delta : ws$ into function symbols $\chi_r : w \to bool$ and $f_\delta : w \to set(s)$, respectively, and interprets χ_r as the *characteristic function* $\chi_r^A : A_w \to \{0,1\}$, defined by $\chi_r^A(a) = 1 \Leftrightarrow a \in r^A$, and $f\delta$ as the set-valued function $f_\delta^A : A_w \to \wp(A_s)$, defined by $f_\delta^A(a) = \{b \in A_s \mid (a,b) \in \delta^A\}$. In fact, an equivalence relation on A is compatible with r iff it is compatible with χ_r, while compatibility with δ is equivalent to compatibility with f_δ.

Proposition 3.2 *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 $\sigma(\varphi)$. A_σ satisfies φ iff for all $\tau : X \to T_\Sigma$, A satisfies $\sigma(\varphi\tau)$.* ❑

Proposition 3.3 *Let $SP = (\Sigma, AX)$ be a swinging specification, $A \in Mod_{\mu\nu}(SP)$, $\approx$ be a weak Σ-congruence on A and $B =_{def} A/\!\approx\, \in Mod(SP)$. Then $B \in Mod_{\mu\nu}(SP)$ (cf. Def. 3.1).*

Proof. Let F be the set of function symbols of Σ, C be an SP-model whose F-reduct agrees with $B|_F$ and D be the Σ-structure whose F-reduct agrees with $A|_F$ and which interprets each predicate $r : w \in \Sigma$ as the set $\{a \in A_w \mid [a] \in r^C\}$. Since C is an SP-model, D is an SP-model. Since $A \in Mod_{\mu\nu}(SP)$ and $\approx$ is compatible with νP and zigzag compatible with μP, we obtain for all $r \in \mu P$,

$$[a] \in r^B \;\Rightarrow\; \exists a' \approx a : a' \in r^A \Rightarrow\; \exists a' \approx a : a' \in r^D \;\Rightarrow\; \exists a' : [a] = [a'] \in r^C,$$

and for all $r \in \nu P$,

$$[a] \in r^C \;\Rightarrow\; a \in r^D \;\Rightarrow\; a \in r^A \;\Rightarrow\; [a] \in r^B.$$

Hence $B \in Mod_{\mu\nu}(SP)$. ❑

Lemma 3.4

(1) Let A be a reachable Σ-structure. An S-sorted equivalence relation $\approx\, \subseteq A \times A$ is a bisimulation iff $\approx$ is a weak congruence.

(2) Suppose that $\sim^A$ is the greatest relation on A satisfying the set $AX_\sim$ of behavior axioms for Σ (cf. Def. 2.4(3)). Then each weak congruence on A satisfies $AX_\sim$ iff it is a subrelation of $\sim^A$.

Proof. (1) The "if"-part follows immediately. Suppose that $\approx$ is a bisimulation. Let $f : s_1 \ldots s_n \to s$ be a function symbol, $a = (a_1, \ldots, a_n) \in A_{s_1 \ldots s_n}$ and $b = (b_1, \ldots, b_n) \in B_{s_1 \ldots s_n}$ such that $a \approx b$. Then there is $(t_1, \ldots, t_n) \in T_{\Sigma, s_1 \ldots s_n}$ such that $a = (t_1^A, \ldots, t_n^A)$. Hence

$$f^A(a) \approx f^A(b_1, t_2^A, \ldots, t_n^A) \approx f^A(b_1, b_2, t_3^A, \ldots, t_n^A) \approx \cdots \approx f^A(b).$$

The compatibility of $\approx$ with $statP$ and the zigzag compatibility of $\approx$ with $dynP$ can be shown analogously.

(2) Let $\approx$ be a weak congruence on A. $\approx$ satisfies the behavior axioms for hidden sorts because they are part of the definition of a weak congruence. If $\approx$ is a subrelation of $\sim^A$, then $a \approx b$ implies $a \sim^A b$. If $a \in A_s$ for some visible sort s, then $a \equiv^A b$ because A satisfies the behavior axiom $x \sim y \Rightarrow x \equiv y$. Hence

$\approx$ satisfies $AX_\sim$. Conversely, if $\approx$ satisfies $AX_\sim$, then $\approx$ is a subrelation of $\sim^A$ because $\sim^A$ is the greatest relation on A satisfying $AX_\sim$. ❑

Lemma 3.5 (monotonicity and homomorphism) *For all sorts $s \in \Sigma$ let $\approx$: ss be a predicate of Σ. Let A and B be Σ-structures.*

(1) Suppose that $\approx$ is a μ-predicate, A is reachable, $\approx^A$ is reflexive and B is a structure with $\approx$-equality. B is monotone w.r.t. A iff there is a (unique) $\mu\nu\Sigma$-homomorphism $h : A \to B$.

(2) Suppose that $\approx$ is a ν-predicate, B is reachable, $\approx^B$ is reflexive and A is a structure with $\approx$-equality. B is monotone w.r.t. A iff there is a (unique) $\nu\mu\Sigma$-homomorphism $h : B \to A$.

(3) Suppose that $\approx$ is a μ-predicate and A is a Herbrand structure. B is monotone w.r.t. A iff there is a (unique) $\mu\nu\Sigma$-homomorphism $h : A \to B$.

Proof. (1) "$\Rightarrow$": Let B be monotone w.r.t. A. Since A is reachable, for all $a \in A$ there is $t \in T_\Sigma$ with $t^A = a$. We define h by $h(t^A) = t^B$. h is well-defined: Let $t^A = u^A$. Since $\approx^A$ is reflexive, we obtain $t^A \approx^A u^A$, i.e. $A \models t \approx u$. Hence $B \models t \approx u$ because B is monotone w.r.t. A and $\approx$ is a μ-predicate. Therefore, $t^B \approx^B u^B$ and thus $h(t^A) = t^B = u^B = h(u^A)$ because B is a structure with $\approx$-equality. Let $f : w \to s \in F$ and $t^A \in A_w$. Then

$$h(f^A(t^A)) = h(f(t)^A) = f(t)^B = f^B(t^B) = f^B(h(t^A)).$$

Let r be a static μ-predicate and $t^A \in r^A$. Then $A \models r(t)$ and thus $B \models r(t)$ because B is monotone w.r.t. A. Hence $h(t^A) = t^B \in r^B$. We have shown $h(r^A) \subseteq r^B$. Let $\delta : ws$ be a dynamic predicate and $(t^A, u^A) \in \delta^A$. Then $A \models \delta(t, u)$ and thus $B \models \delta(t, v)$ for some $v \in T_{\Sigma,s}$ with $v^A = u^A$ because B is monotone w.r.t. A. Hence $h(t^A, u^A) = h(t^A, v^A) = (t^B, v^B) \in \delta^B$ and thus $h(\delta^A) \subseteq \delta^B$. Let r be a ν-predicate and $t^B \in r^B$. Hence $B \models r(t)$ and thus $A \models r(t)$ because B is monotone w.r.t. A. We conclude $t^A \in r^A$ and thus $t^B = h(t^A) \in h(r^A)$. Therefore, $r^B \subseteq h(r^A)$.

"$\Leftarrow$": Let $h : A \to B$ be a $\mu\nu\Sigma$-homomorphism and $t \in T_\Sigma$. By induction on the size of t one shows $h(t^A) = t^B$. Let $r(t)$ be a ground μ-atom such that $A \models r(t)$. Then $t^A \in r^A$ and thus $t^B = h(t^A) \in h(r^A) \subseteq r^B$. Therefore, $B \models r(t)$. Let $r(t)$ be a ground ν-atom such that $B \models r(t)$. Then $t^B \in r^B \subseteq h(r^A)$. Hence $t^B = h(t^A)$ for some $t^A \in r^A$. We conclude $A \models r(t)$.

(2) "$\Rightarrow$": Let B be monotone w.r.t. A. Since B is reachable, for all $b \in B$ there is $t \in T_\Sigma$ with $t^B = b$. We define h by $h(t^B) = t^A$. h is well-defined: Let $t^B = u^B$. Since $\approx^B$ is reflexive, we obtain $t^B \approx^B u^B$, i.e. $B \models t \approx u$. Hence $A \models t \approx u$ because B is monotone w.r.t. A and $\approx$ is a ν-predicate. Therefore, $t^A \approx^A u^A$ and thus $h(t^B) = t^A = u^A = h(u^B)$ because A is a structure with $\approx$-equality. Let $f : w \to s \in F$ and $t^B \in B_w$. Then

$$h(f^B(t^B)) = h(f(t)^B) = f(t)^A = f^A(t^A) = f^A(h(t^B)).$$

Let r be a ν-predicate and $t^B \in r^B$. Then $B \models r(t)$ and thus $A \models r(t)$ because B is monotone w.r.t. A. Hence $h(t^B) = t^A \in r^A$. We have shown $h(r^B) \subseteq r^A$. Let r be a static μ-predicate and $t^A \in r^A$. Hence $A \models r(t)$ and thus $B \models r(t)$ because B is monotone w.r.t. A. We conclude $t^B \in r^B$ and thus $t^A = h(t^B) \in h(r^B)$. Therefore, $r^A \subseteq h(r^B)$. Let $\delta : ws$ be a dynamic predicate and $(t^A, u^A) \in \delta^A$. Hence $A \models \delta(t, u)$ and thus $B \models \delta(t, v)$ for some $v \in T_{\Sigma,s}$ with $v^A = u^A$ because because B is monotone w.r.t. A. We conclude $(t^B, v^B) \in \delta^B$ and thus $(t^A, u^A) = (t^A, v^A) = h(t^B, v^B) \in h(\delta^B)$. Therefore, $\delta^A \subseteq h(\delta^B)$.

"$\Leftarrow$": Let $h : B \to A$ be a $\nu\mu\Sigma$-homomorphism and $t \in T_\Sigma$. By induction on the size of t one shows $h(t^B) = t^A$. Let $r(t)$ be a ground μ-atom such that $A \models r(t)$. Then $t^A \in r^A \subseteq h(r^B)$. Hence $t^A = h(t^B)$ for some $t^B \in r^B$. We conclude $B \models r(t)$. Let $r(t)$ be a ground ν-atom such that $B \models r(t)$. Then $t^B \in r^B$ and thus $t^A = h(t^B) \in h(r^B) \subseteq r^A$. Therefore, $A \models r(t)$.

(3) "$\Rightarrow$": Let B be monotone w.r.t. A. We define h by $h(t^A) = t^B$. Let $f : w \to s \in F$ and $t^A \in A_w$. Then

$$h(f^A(t)) = h(f(t)) = f(t)^B = f^B(t^B) = f^B(h(t)).$$

$h(r^A) \subseteq r^B$ for all $r \in \mu P$ and $r^B \subseteq h(r^A)$ for all $r \in \nu P$ follow as in the proof of (1).

"$\Leftarrow$": As in the proof of (1). ❑

The interpretation of $\sim$ in an SP-model A need not be a weak congruence. It is easy to see that the quotient $A/{\sim^A}$ is well-defined if and only if $\sim^A$ is a weak congruence. Hence A is behaviorally SP-consistent *only if* $\sim^A$ is a weak congruence. Due to the modality assumptions on the axioms of SP, the converse holds true as well: if $\sim^A$ is a weak congruence, then A is behaviorally consistent (Thm. 3.9(b)).

Lemma 3.6 *Let $SP = (\Sigma, AX)$ be a swinging specification and $A \in Mod_{be}(SP)$. Then $\sim^A$ is compatible with all visible function symbols and all visible or behavioral-equality predicates of Σ. Moreover, $\sim^A$ is zigzag compatible with all equality predicates of Σ.*

Proof. Let $AX_\sim$ be the set of behavior axioms for Σ (cf. 2.4(4)). Since $\sim^A$ satisfies $AX_\sim$ and $\equiv^A$ is a subset of $\sim^A$, $\sim^A$ and $\equiv^A$ coincide on visible carriers.

Since $\sim^A$ and $\equiv^A$ coincide on visible carriers and $\equiv^A$ is transitive, $\sim^A$ is compatible with all visible function symbols and predicates of Σ.

Since $\sim^A$ is transitive, $\sim^A$ is compatible with all behavioral equalities of Σ. Since $\equiv^A$ is a subset of $\sim^A$, $\sim^A$ is transitive and $\equiv^A$ is reflexive, $\sim^A$ is zigzag compatible with all equality predicates of Σ. ❑

Proposition 3.7 *Let $\Sigma' \subseteq \Sigma$, A be a reachable Σ'-structure and B be a reachable Σ-structure such that for all ground Σ'-atoms p, $A \models p$ iff $B \models p$. Then $A \cong B_{\Sigma'}$.* ❑

Let A be an SP-model. Then $\equiv^A$ is a Σ-congruence and thus $A/{\equiv^A}$ is an SP-model with $\equiv$-equality. If $\sim^A$ is a weak congruence, then $A/{\sim^A}$ is an SP-model (Thm. 3.9(b)). For obtaining this result we have restricted the axioms of SP to clauses with modal premises resp. conclusions (cf. 2.4). The difference between a congruence and a weak congruence only concerns dynamic predicates (see Figure 1).

So far ADT approaches[7] mostly stick to *functions* for specifying behavioral properties. Transition *relations* only occur in the dynamic-data-type approach [5, 21]. Other restrictions concern the axioms. For instance, dynamic atoms are not admitted as axiom premises because otherwise factorizing w.r.t. behavioral equivalence may violate the axioms' validity. Are such constraints really necessary?

In Def. 3.1, we have given the interpretation of $dynP$ in quotients by weak congruences. The question is which sets of formulas are closed under the modified quotient construction. Modal logic's Hennessy-Milner Theorem (see Section 1) provides the key idea: two states s and t are bisimilar iff for all modal-logic formulas $\varphi(x)$, $\varphi(s) \Leftrightarrow \varphi(t)$. The following theorem provides corresponding results for our notions of modality (cf. Def. 2.3).

Theorem 3.8 (invariance properties of modal formulas) *Let Σ be a swinging signature and A be a Σ-structure.*

(1) Modal formulas are bisimulation invariant in all classes of Σ-structures.

[7] ADT = abstract data types

(2) *Let $\approx$ be a weak congruence on A, φ be a weakly modal formula with output Y and $b, c : X \to A$. Then $b \approx c$ and $A \models_b \varphi$ imply $A \models_{c'} \varphi$ for some c' with $b \approx c' =_Y c$.*

(3) **Hennessy-Milner Theorem.** *Suppose that $\sim^A$ is a weak congruence. Then for all $b, c : X \to A$, $b \sim^A c$ iff for all poly-modal formulas φ, $A \models_b \varphi$ iff $A \models_c \varphi$.*

Proof. (1) Let A and B be Σ-structures, $\approx \subseteq A \times B$ be a bisimulation, $\varphi = \varphi(x)$ be a modal formula, $a \in A$ and $b \in B$ such that $a \approx b$ and w.l.o.g. $A \models_{a/x} \varphi$.

Case 1. φ is a static atom, say $\varphi = r(t_1, \ldots, x, \ldots, t_n)$. Then $A \models_{a/x} \varphi$ implies $(t_1^A, \ldots, a, \ldots, t_n^A) \in r^A$. Since r is static and $\approx$ is a bisimulation, $a \approx b$ implies $(t_1^B, \ldots, b, \ldots, t_n^B) \in r^B$, i.e. $B \models_{b/x} \varphi$.

Case 2. $\varphi = \neg\psi$ for a modal formula ψ. Then $A \not\models_{a/x} \psi$. By induction hypothesis, $B \not\models_{b/x} \psi$. Hence $B \models_{b/x} \varphi$.

Case 3. $\varphi = (\psi(x) \wedge \vartheta(x))$ for modal formulas $\psi = \psi(x)$ and $\vartheta = \vartheta(x)$. Then $A \models_{a/x} \psi$ and $A \models_{a/x} \vartheta$. By induction hypothesis, $B \models_{b/x} \psi$, $B \models_{b/x} \vartheta$. Hence $B \models_{b/x} \varphi$.

Case 4. $\varphi = \exists y(\delta(t(x), y) \wedge \psi)$ for a dynamic atom $\delta(t(x), y)$ and a modal formula $\psi = \psi(y)$ such that $x \neq y$. Let $t(x) = (t_1, \ldots, t_i(x), \ldots, t_n)$. Since $\approx$ is a bisimulation, $a \approx b$ implies $t_i^A(a) \approx t_i^B(b)$. Moreover, $A \models_{a/x} \varphi$ implies $(t_1^A, \ldots, t_i^A(a), \ldots, t_n^A), a') \in \delta^A$ and $A \models_{a'/y} \psi$ for some $a' \in A$. Since $\approx$ is a bisimulation, $t_i^A(a) \approx t_i^B(b)$ implies $(t_1^B, \ldots, t_i^B(b), \ldots, t_n^B), b') \in \delta^B$ and thus $B \models_{(b/x)[b'/y]} \delta(t(x), y)$ for some $b' \in B$ with $a' \approx b'$. Since $\psi(y)$ is modal, the induction hypothesis implies $B \models_{b'/y} \psi$. Hence $B \models_{b/x} \varphi$.

(2) Let $b, c : X \to A$ such that $b \approx c$ and $A \models_b \varphi$.

Case 1. φ is poly-modal.

Case 1.1. φ is a static atom. Then $B \models_c \varphi$ follows from the compatibility of $\approx$ with function symbols and static predicates.

Case 1.2. $\varphi = \neg\psi$ for a poly-modal formula ψ. Then $A \not\models_b \psi$. By induction hypothesis implies $A \not\models_c \psi$. Hence $A \models_c \varphi$.

Case 1.3. $\varphi = (\psi \wedge \vartheta)$ for poly-modal formulas ψ and ϑ. Then $A \models_b \psi$ and $A \models_b \vartheta$. By induction hypothesis, $A \models_c \psi$, $A \models_c \vartheta$. Hence $A \models_c \varphi$.

Case 1.4. $\varphi = \exists x\psi$ for a poly-modal formula ψ. Then $A \models_b [a/x]\psi$ for some $a \in A$. Since $\approx$ is reflexive, the induction hypothesis implies $A \models_c [a/x]\psi$. Hence $A \models_c \varphi$.

Case 1.5. $\varphi = \exists x(\delta(t, x) \wedge \psi)$ for a dynamic atom $\delta(t, x)$ and a poly-modal formula ψ such that $x \notin var(t)$. Since $A \models_b \varphi$, there is $a \in A$ such that $(b^*(t), a) \in \delta^A$ and $A \models_b [a/x]\psi$. Since $\approx$ is zigzag compatible with δ, $b \approx c$ implies $(c^*(t), a') \in \delta^A$ and thus $A \models_c [a'/x]\delta(t, x)$ for some $a' \approx a$. By induction hypothesis, $A \models_c [a'/x]\psi$. Hence $A \models_c \varphi$.

Case 2. $\varphi = \delta(t, x)$ is a dynamic atom with $x \in X \setminus var(t)$. Then $(b^*(t), b(x)) \in \delta^A$. Since $\approx$ is zigzag compatible with δ, $b \approx c$ implies $(c^*(t), a) \in \delta^A$ for some $a \approx b(x)$. Define c' by $c'(x) = a$ and $c' =_x c$. Since $x \notin var(t)$, $A \models_{c'} \varphi$.

Case 3. $\varphi = (\psi \wedge \vartheta)$ for weakly modal formulas ψ and ϑ with disjoint outputs Y resp. Z. By induction hypothesis, $A \models_d \psi$, $A \models_{d'} \vartheta$ for some d, d' with $b \approx d =_Y c$ and $b \approx d' =_Z c$. Since Y and Z are disjoint, we may define c' by $c' =_{Y \cup Z} c$, $c'(x) = d(x)$ for all $x \in Y$ and $c'(x) = d'(x)$ for all $x \in Z$. Since $d =_Y c$, $A \models_d \psi$ implies $A \models_{c'} \psi$. Since $d' =_Z c$, $A \models_{d'} \vartheta$ implies $A \models_{c'} \vartheta$. Hence $A \models_{c'} \varphi$. Moreover, $b \approx c'$.

Case 4. $\varphi = \exists x\psi$ for a weakly modal formula ψ with output Y. Then $A \models_b [a/x]\psi$ for some $a \in A$. By induction hypothesis, $A \models_d \psi$ for some d with $b[a/x] \approx d =_Y c$. We define c' by $c'(x) = c(x)$ and $c' =_x d$. Hence $A \models_{c'[d(x)/x]} \psi$ and thus $A \models_{c'} \varphi$. Moreover, $b \approx c' =_{Y\setminus\{x\}} c$.

(3) Let φ be poly-modal and $b, c : X \to A$ such that $b \sim^A c$ and $A \models_b \varphi$. Then (2) implies $A \models_c \varphi$. Suppose that, conversely, for all poly-modal formulas φ, $A \models_b \varphi$ iff $A \models_c \varphi$. Then, in particular, $b \sim^A b$ implies $b \sim^A c$ because $A \models_b x \sim x$ and $x \sim x$ is poly-modal. ❑

The converse of Thm. 3.8(1): bisimulation invariant formulas are modal, will be proved in Section 7 (Thm. 7.9). So far it provides the only reason for our consideration of bisimulations between *different* structures. The proof of Thm. 7.9 involves steps from a given structure to new ones. Hence the result can only be obtained with respect to a *class* of structures that is closed under all model constructions used in the proof.

The following result deals only with structures that interpret $\sim$ as a weak Σ-congruence and is proved similarly to Thm. 3.8(2):

Theorem 3.9 (modal formulas and behaviorally consistent models) *Let SP be a swinging specification and A be a Σ-structure such that $\sim^A$ is a weak Σ-congruence. Let φ be a weakly modal formula with output Y and $B = A/\sim^A$.*

- *(a) For all $c : X \to A$, $B \models_{nat\circ c} \varphi$ iff $A \models_{c'} \varphi$ for some c' with $c \sim^A c' =_Y c$.*
- *(b) A is behaviorally SP-consistent.*

Proof. **(a)** follows from:

(1) $B \models_{nat\circ c} \varphi \;\Rightarrow\; A \models_c \varphi$ if φ is poly-modal and $A \models_{c'} \varphi$ for some c' with $c \sim^A c' =_Y c$ otherwise,

(2) $A \models_c \varphi \;\Rightarrow\; B \models_{nat\circ c} \varphi$.

We show (1) and (2) by induction on the structure of φ.

(1) Let $B \models_{nat\circ c} \varphi$. *Case 1.* φ is a poly-modal formula.

Case 1.1. φ is a static atom. Then $A \models_c \varphi$ follows from the interpretation of functions symbols and static predicates in B.

Cases 1.2 and 1.3. $\varphi = \neg\psi$ or $\varphi = (\psi \wedge \vartheta)$ for poly-modal formulas ψ and ϑ. $A \models_c \varphi$ can be shown analogously to Case 1.2 resp. 1.3 of the proof of Thm. 3.8(2).

Case 1.4. $\varphi = \exists z\psi$ for a poly-modal formula ψ. Then $B \models_{nat\circ c[a/x]} \psi$ for some $a \in A$. By induction hypothesis, $A \models_{c[a/x]} \psi$. Hence $A \models_c \varphi$.

Case 1.5. $\varphi = \exists x(\delta(t,x) \wedge \psi)$ for a dynamic atom $\delta(t,x)$ and a poly-modal formula ψ such that $x \notin var(t)$. Then $B \models_{nat\circ c[a/x]} (\delta(t,x) \wedge \psi)$ for some $a \in A$. By the interpretation of dynamic predicates in B, $(c^*(t), a') \in \delta^A$ and thus $A \models_{c[a'/x]} \delta(t,x)$ for some $a' \sim^A a$. Since $B \models_{nat\circ c[a/x]} \psi$ and ψ is poly-modal, the induction hypothesis implies $A \models_{c[a/x]} \psi$. Since $\sim^A$ is a weak congruence and $a \sim^A a'$, Thm. 3.8(3) implies $A \models_{c[a'/x]} \psi$. Hence $A \models_c \varphi$.

Case 2. $\varphi = \delta(t,x)$ is a dynamic atom with $x \in X \setminus var(t)$. By the interpretation of dynamic predicates in B, $B \models_{nat\circ c} \varphi$ implies $(c^*(t), a) \in \delta^A$ for some $a \sim^A c(x)$. We obtain $A \models_{c'} \varphi$ for c' defined by $c'(x) = a$ and $c' =_x c$. Hence $c' \sim^A c$.

Case 3. $\varphi = (\psi \wedge \vartheta)$ for weakly modal formulas ψ and ϑ with disjoint outputs Y resp. Z. $A \models_{c'} \varphi$ for some c' with $c \sim^A c' =_{Y\cup X} c$ can be shown analogously to Case 3 of the proof of Thm. 3.8(2).

Case 4. $\varphi = \exists x\psi$ for a weakly modal formula ψ with output Y. $A \models_{c'} \varphi$ for some c' with $c \sim^A c' =_{Y\setminus\{x\}} c$

can be shown analogously to Case 4 of the proof of Thm. 3.8(2).

(2) Let $A \models_c \varphi$. *Case 1.* φ is a poly-modal formula.

Case 1.1. φ is a static atom. Then $B \models_{natoc} \varphi$ follows from the interpretation of functions symbols and static predicates in B.

Cases 1.2 and 1.3. $\varphi = \neg\psi$ or $\varphi = (\psi \wedge \vartheta)$ for poly-modal formulas ψ and ϑ. $B \models_{natoc} \varphi$ can be shown analogously to Case 1.2 resp. 1.3 of the proof of Thm. 3.8(2).

Case 1.4. $\varphi = \exists x\psi$ for a poly-modal formula ψ. Then $A \models_{c[a/x]} \psi$ for some $a \in A$. By induction hypothesis, $B \models_{natoc[a/x]} \psi$. Hence $B \models_{natoc} \varphi$.

Case 1.5. $\varphi = \exists x(\delta(t,x) \wedge \psi)$ for a dynamic atom $\delta(t,x)$ and a poly-modal formula ψ such that $x \notin var(t)$. Then $A \models_{c[a/x]} (\delta(t,x) \wedge \psi)$ for some $a \in A$. Since ψ is modal, the induction hypothesis implies $B \models_{natoc[a/x]} \psi$, while $B \models_{natoc[a/x]} \delta(t,x)$ follows from the interpretation of dynamic predicates in B. Hence $B \models_{natoc} \varphi$.

Case 2. $\varphi = \delta(t,x)$ for a dynamic atom $\delta(t,x)$ with $x \in X \setminus var(t)$. By the interpretation of dynamic predicates in B, $A \models_c \varphi$ implies $B \models_{natoc} \varphi$.

Case 3. $\varphi = (\psi \wedge \vartheta)$ for weakly modal formulas ψ and ϑ with disjoint outputs Y resp. Z. $B \models_{natoc} \varphi$ can be shown analogously to Case 3 of the proof of Thm. 3.8(2).

Case 4. $\varphi = \exists x\psi$ for a weakly modal formula ψ with output Y. By induction hypothesis, $B \models_{natoc[a/x]} \psi$. Hence $B \models_{natoc} \varphi$.

(b) Let φ be an axiom of SP and $A \models \varphi$. We show $B \models \varphi$.

Case 1. φ is a (generalized) Horn axiom, say $\varphi = (p \Leftarrow \psi)$. Let $c : X \to A$ such that $B \models_{natoc} H$. Then φ belongs to $hidSP$ or the μ-level of SP and thus, by Def. 2.4(b), ψ is weakly modal. Hence (1) implies $A \models_{c'} H$ for some $c' \sim^A c$. Since A satisfies φ, we obtain $A \models_{c'} p$ and thus $B \models_{natoc'} p$ by the interpretation of predicates in B. Hence $c' \sim^A c$ implies $B \models_{natoc} p$. Therefore, $B \models_{natoc} \varphi$, and we conclude $B \models \varphi$.

Case 2. φ is a co-Horn axiom, say $\varphi = (p \Rightarrow \psi)$. Let $c : X \to A$ such that $B \models_{natoc} p$. Since p is a static atom, $A \models_c p$ and thus $A \models_c \psi$ because A satisfies φ. Since ψ is poly-modal, (2) implies $B \models_{natoc} \psi$. Therefore, $B \models_{natoc} \varphi$. Again we conclude $B \models \varphi$. ❏

Lemma 3.10 *Let SP be a visible specification and $A \in Mod_{be}(SP)$. Then A is behaviorally SP-consistent.*

Proof. Since $\sim^A$ includes $\equiv^A$ and A satisfies the first behavior axiom for Σ, $\sim^A$ coincides with $\equiv^A$. Hence $\sim^A$ is a congruence and thus a weak congruence. By Thm. 3.9(b), A is behaviorally SP-consistent. ❏

4 Functionality, fixpoints, standard models

Definition 4.1 (structural SP-equivalence, functionality) Let SP be a swinging specification and $hidSP = (\Sigma, AX)$ (cf. Def. 2.4). The **cut calculus for** SP consists of the following inference rules for deriving Horn clauses.[8]

axiom rule $\quad \dfrac{TRUE}{\varphi} \Downarrow \quad$ where $\varphi \in AX \cup EQ_\Sigma$

[8] Arrows attached to a rule indicate the direction of consequence, here with respect to all Σ-structures.

instantiation $\dfrac{\varphi}{\varphi\sigma} \Downarrow$ where $\sigma : X \to T_\Sigma(X)$

modus ponens $\dfrac{p \Leftarrow \varphi, \quad \varphi}{p} \Downarrow$

$\wedge$-introduction $\dfrac{\varphi, \quad \psi}{\varphi \wedge \psi} \Downarrow$

Given a formula φ, we write $SP \vdash_{cut} \varphi$ if φ is derivable with the cut calculus for SP. **(Structural) SP-equivalence** is the binary relation on T_Σ that is defined as follows:

$$t \equiv_{SP} t' \quad \Longleftrightarrow_{def} \quad SP \vdash_{cut} t \equiv t'.$$

Given $t \in T_\Sigma$ and $u \in NF_\Sigma$, u is a **normal form of** t if t and u are SP-equivalent. SP is **complete** if each ground Σ-term has a normal form. SP is **(structurally) consistent** if each two SP-equivalent ground normal forms are equal. SP is **relational** if it does not contain defined functions. SP is **functional** if it is complete and consistent. In this case **nf(t)** denotes the unique normal form of a ground term (tuple) t. ❑

Relational specifications are functional.[9]

Both completeness and consistency are essential for ensuring the soundness of proof rules dealing with constructors and defined functions. Consistency calls for syntactic criteria some of which are already involved in the definition of a swinging specification. For sufficient ones, see [60, 63]. Completeness is a much simpler proof obligation: SP is complete iff for all defined functions $f : w \to s$ and $t \in NF_{\Sigma,w}$, $f(t) \equiv_{SP} u$ for some $u \in NF_\Sigma$. This is either shown "by hand" and induction on t or by constructing a (semi-)automatic proof of the formula $Def(x) \Rightarrow Def(f(x))$ (see Section 2).

A term model of *hidSP* (cf. 2.4) could be defined directly in terms of the cut calculus. Since this does not work for the ν- and μ-levels of SP, we prefer an equivalent definition that uses *consequence operators* on substructures. Roughly said, a consequence operator Φ stepwise adds valid atoms to models of a subspecification. If Φ is *monotone*, a model of the entire specification is obtained from a fixpoint of Φ. The existence of a suitable fixpoint is ensured by the fixpoint theorem of Knaster and Tarski. Morever, if Φ is *continuous*, then Kleene's fixpoint theorem provides a stepwise construction of the fixpoint. Let us recapitulate set-theoretical versions of these fixpoint theorems.

Definition and Theorem 4.2 (continuity, fixpoints) (cf., e.g., [49]) Let U be a sorted set and $\Phi : \wp(U) \to \wp(U)$ be a monotone function with respect to sorted set inclusion. $B \subseteq U$ is a **fixpoint** of Φ if $\Phi(B) = B$. $\Phi^\infty =_{def} \cup_{i\in\mathbb{N}}\Phi^i(\emptyset)$ and $\Phi_\infty =_{def} \cap_{i\in\mathbb{N}}\Phi^i(U)$ are the **Kleene closures of Φ**. Φ is **upward continuous** if for all increasing chains $B_0 \subseteq B_1 \subseteq B_2 \subseteq \ldots$ of subsets of U, $\Phi(\cup_{i\in\mathbb{N}}B_i)$ is a subset of $\cup_{i\in\mathbb{N}}\Phi(B_i)$. Φ is **downward continuous** if for all decreasing chains $B_0 \supseteq B_1 \supseteq B_2 \supseteq \ldots$ of subsets of U, $\cap_{i\in\mathbb{N}}\Phi(B_i)$ is a subset of $\Phi(\cap_{i\in\mathbb{N}}B_i)$.

Knaster-Tarski Theorem. *$lfp(\Phi) =_{def} \cap\{B \subseteq U \mid \Phi(B) \subseteq B\}$ is the least fixpoint of Φ and a superset of Φ^∞. $gfp(\Phi) =_{def} \cup\{B \subseteq U \mid B \subseteq \Phi(B)\}$ is the greatest fixpoint of Φ and a subset of Φ_∞.*

Kleene's Theorem. *If Φ is upward continuous, then $\Phi(\Phi^\infty) \subseteq \Phi^\infty$ and thus $\Phi^\infty = lfp(\Phi)$. If Φ is downward continuous, then $\Phi_\infty \subseteq \Phi(\Phi_\infty)$ and thus $\Phi_\infty = gfp(\Phi)$.* ❑

Definition 4.3 (consequence operator) Let $SP = (\Sigma, AX)$ be a specification, $SP' = (\Sigma', AX')$ be a subspecification of SP, A be a Σ'-structure and $\mathcal{C}$ be the class of Σ-structures whose Σ'-reduct agrees with A. The **$(AX \setminus AX')$-consequence operator on** A, $\Phi : \mathcal{C} \to \mathcal{C}$, is defined as follows. For all μ-predicates

[9] No joke!

$r \in \Sigma \setminus \Sigma'$ and $B \in \mathcal{C}$,

$$a \in r^{\Phi(B)} \iff_{def} \exists\ (r(t) \Leftarrow \varphi) \in AX \setminus AX'\ \exists\ b : X \to A : a = b^*(t) \wedge B \models_b \varphi.$$

For all ν-predicates $r \in \Sigma \setminus \Sigma'$ and $B \in \mathcal{C}$,

$$a \in r^{\Phi(B)} \iff_{def} \forall\ (r(t) \Rightarrow \varphi) \in AX \setminus AX'\ \forall\ b : X \to A : a = b^*(t) \Rightarrow B \models_b \varphi. \ ❑$$

In terms of Thm. 4.2, $B \in \mathcal{C}$ is regarded as a $(\Sigma \setminus \Sigma')$-sorted subset of the structure $U \in \mathcal{C}$ that interprets $r : w \in \Sigma \setminus \Sigma'$ as an all-relation, i.e. $r^U =_{def} A_w$. On the other hand, $\emptyset \in \mathcal{C}$ interprets $r : w \in \Sigma \setminus \Sigma'$ as an empty relation, i.e. $r^\emptyset =_{def} \emptyset$. $\Phi : \mathcal{C} \to \mathcal{C}$ is monotone iff for all $B, C \in \mathcal{C}$ and $r \in \Sigma \setminus \Sigma'$, $r^B \subseteq r^C$ implies $r^{\Phi(B)} \subseteq r^{\Phi(C)}$.

Lemma 4.4 *Let $SP = (\Sigma, AX)$ be a swinging specification and $\Sigma = (S, F, P)$.*

(1) $hidAX$-consequence operators on $(S, F, \emptyset)$-algebras are monotone and upward continuous.

(2) $(\nu AX \setminus hidAX)$-consequence operators on $hid\Sigma$-structures are monotone.

(3) $(AX \setminus \nu AX)$-consequence operators on $\nu\Sigma$-structures are monotone.

Proof. (1) holds true because $hidAX$ consists of Horn clauses and thus for all $p \Leftarrow \varphi \in hidAX$, φ does not contain neither negation symbols nor implication symbols nor universal quantifiers.

(2) $\nu AX \setminus hidAX$ consists of co-Horn clauses such that, by Def. 2.4(3) and (4), for all $p \Rightarrow (G \Rightarrow \varphi) \in \nu AX \setminus hidAX$, G is a $hid\Sigma$-goal. Hence $G \Rightarrow \varphi$ is equivalent to a first-order formula ψ such that all negation symbols of ψ directly precede $hid\Sigma$-atoms and ψ does not contain $FALSE$. Let A be a $hid\Sigma$-structure and $\mathcal{C}$ be the class of $\nu\Sigma$-structures whose $hid\Sigma$-reduct agrees with A. Since for all $B \in \mathcal{C}$ and predicates $r \in hid\Sigma$, $r^B = r^A$, we may assume that $hid\Sigma$ also includes the complement $\overline{r}$ of r (cf. Def. 3.1). Hence for all $B \in \mathcal{C}$ and $b : X \to B$,

$$B \models_b G \Rightarrow \varphi \iff B \models_b \psi \iff B \models_b \psi[\overline{r}(t)/\neg r(t) \mid r \in hid\Sigma],$$

i.e. $G \Rightarrow \varphi$ is equivalent in $\mathcal{C}$ to a negation- and implication-free formula. Therefore, the $(\nu AX \setminus hidAX)$-consequence operator on A is monotone.

(3) $AX \setminus \nu AX$ consists of generalized Horn clauses such that, by Def. 2.4(5), for all $p \Leftarrow \varphi \in AX \setminus \nu AX$ and all universal goals $\forall Y(G \Rightarrow h)$ of φ, G is a $\nu\Sigma$-goal. Hence φ is equivalent to a first-order formula ψ where all negation symbols directly precede ν-atoms. Let A be a $\nu\Sigma$-structure and $\mathcal{C}$ be the class of Σ-structures whose $\nu\Sigma$-reduct agrees with A. Since for all $B \in \mathcal{C}$ and predicates $r \in \nu\Sigma$, $r^B = r^A$, we may assume that $\nu\Sigma$ also includes the complement $\overline{r}$ of r. Hence for all $B \in \mathcal{C}$ and $b : X \to B$,

$$B \models_b \varphi \iff B \models_b \psi \iff B \models_b \psi[\overline{r}(t)/\neg r(t) \mid r \in \nu\Sigma],$$

i.e. φ is equivalent in $\mathcal{C}$ to a negation- and implication-free formula. Therefore, the $(AX \setminus \nu AX)$-consequence operator on A is monotone. ❑

Definition 4.5 (initial and final structures) Let Σ be a swinging signature and $\mathcal{C}$ be a class of Σ-structures, $I \in \mathcal{C}$ is **initial in** $\mathcal{C}$ if for all $A \in \mathcal{C}$ there is a unique $\mu\nu\Sigma$-homomorphism $h : I \to A$. $T \in \mathcal{C}$ is **final in** $\mathcal{C}$ if for all $A \in \mathcal{C}$ there is a unique $\nu\mu\Sigma$-homomorphism $h : A \to T$ (cf. Def. 3.1). ❑

Each two initial (resp. final) Σ-structures are Σ-isomorphic.

Definition 4.6 (standard models, behavioral equivalence) Let $SP = (\Sigma, AX)$ be a swinging specification with empty parameter. The **Herbrand SP-model**, **Her(SP)**, is the Herbrand Σ-structure that is defined as follows (cf. Def. 4.3):

- For all predicates $r \in hid\Sigma$, $r^{Her(SP)} = r^{lfp(\Phi)}$ where Φ is the $hidAX$-consequence operator on T_Σ.
- For all $r \in \nu\Sigma \setminus hid\Sigma$, $r^{Her(SP)} = r^{gfp(\Psi)}$ where Ψ is the $(\nu AX \setminus hidAX)$-consequence operator on $Her(SP)|_{hid\Sigma}$.
- For all $r \in \Sigma \setminus \nu\Sigma$, $r^{Her(SP)} = r^{lfp(\Theta)}$ where Θ is the $(AX \setminus \nu AX)$-consequence operator on $Her(SP)|_{\nu\Sigma}$.

The interpretation of $\sim$ in $Her(SP)$ is called **behavioral SP-equivalence** and denoted by $\sim_{SP}$. SP is **behaviorally consistent** if $\sim_{SP}$ is a weak Σ-congruence. SP is **continuous** if the above consequence operators Ψ and Θ are downward resp. upward continuous. A first-order formula satisfied by $Her(SP)$ is called an **inductive theorem of** SP.

The **initial SP-model**, **Ini(SP)**, is the quotient of $Her(SP)$ by $\equiv_{SP}$. Provided that SP is behaviorally consistent, the **final SP-model**, **Fin(SP)**, is the quotient of $Her(SP)$ by $\sim_{SP}$.

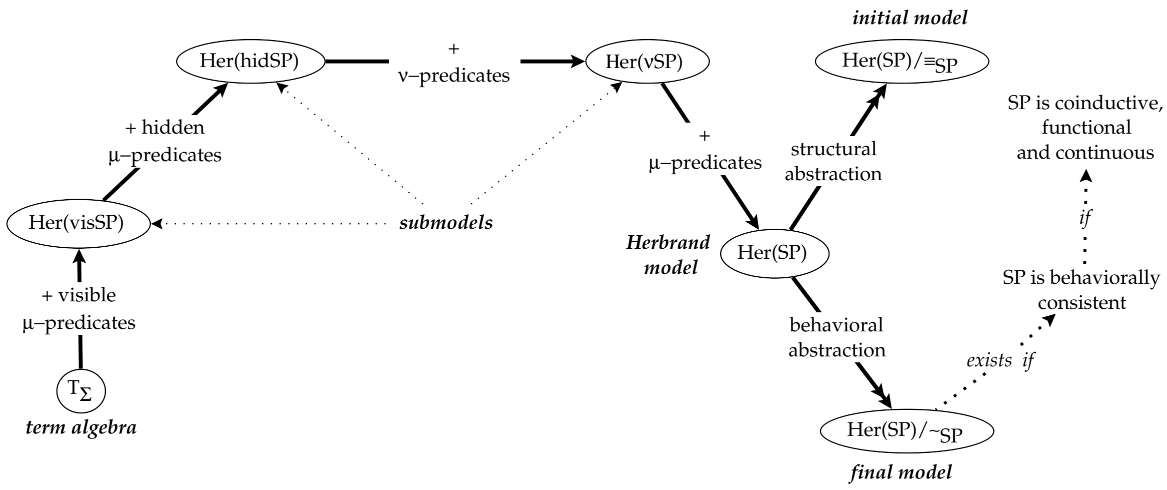

Figure 2. *Standard models of a swinging specification*

Given a parameterized specification SP with parameter $PAR = (\Sigma, AX)$ (cf. Section 2), an **actualization** $SP[\sigma]$ **of** SP **along** σ is **correct w.r.t.** a specification $SP' \supseteq SP[\sigma]$ if $Her(SP')$ satisfies $\sigma(AX)$. The **Herbrand SP-model** is the class of Herbrand models of actualizations of SP. An **inductive theorem of** SP is an inductive theorem of all actualizations of SP. SP is **functional**, **behaviorally consistent** or **continuous** if all actualizations of SP are functional, behaviorally consistent or continuous, respectively. ❑

Figure 2 illustrates the stepwise construction of the standard models $Her(SP)$, $Ini(SP)$ and $Fin(SP)$. Image finiteness and coinductivity, besides functionality the main criteria for the existence of $Fin(SP)$, are defined and discussed in Sections 5 and 6, respectively.

The notion "inductive theorem" stems from the fact that the valuations of variables in a Herbrand model are ground term substitutions and thus a first-order Σ-formula φ can be proved by structural induction on the instances of φ by ground terms, i.e.

$$Her(SP) \models \varphi \iff \forall \sigma : X \to T_\Sigma : Her(SP) \models \varphi\sigma.$$

The definition of an inductive theorem φ of a swinging specification SP with parameter PAR entails that the axioms of PAR are the only assumptions about the parameter a proof of φ may refer to.

If SP is functional, then the set NF_Σ of ground normal forms extends to a Σ-structure: for all $f : w \to s \in \Sigma$ and $t \in NF_{\Sigma,w}$, $f^{NF_\Sigma}(t) =_{def} nf(f(t))$ (cf. 4.1), and for all $r : w \in \Sigma$, $r^{NF_\Sigma} =_{def} \{t \in NF_\Sigma \mid t \in r^{Her(SP)}\}$. The normal form function nf induces a Σ-isomorphism from the initial SP-model to NF_Σ that maps the $\equiv_{SP}$-equivalence class of $t \in T_\Sigma$ to the normal form of t. Hence NF_Σ is an SP-model. *This* is the model construction one has in mind when talking about initial semantics. Consequently, if SP is not functional and thus NF_Σ is not an SP-model, SP has no "proper" initial semantics.

First of all, functionality and, in particular, consistency (cf. 4.1) depend on the *constructors* of SP, whereas behavioral consistency is a property of behavioral equivalence, which is specified by the behavior axioms that, in turn, are determined by the *observers* of SP (cf. 2.4). This reveals a duality between constructors and observers. On the one hand, SP may lack observers. Then all hidden terms are behaviorally equivalent, and visible terms are behaviorally equivalent iff they are structurally equivalent. On the other hand, in contrast to coalgebraic specifications of hidden types (cf. [62]), SP should not lack constructors. Constructors are the "building blocks" of both visible and hidden data. If SP has no constructors, SP can be functional only if the Herbrand, initial and final SP-models are empty.

If all predicates of SP are equalities and thus all observers are functions (destructors), then behavioral SP-equivalence is *contextual equivalence*, i.e. for all $t, t' \in T_\Sigma$,

$$t \sim_{SP} t' \iff \forall \text{ visible terms } c(x) : Her(SP) \models c(t) \equiv c(t'). \tag{6}$$

The construction of $Her(SP)$ reflects the hierarchical syntax of SP, such as stable models mirror the hierarchical syntax of stratified logic programs (cf. [2]):

Lemma 4.7 (stepwise constructions of the Herbrand model) *Let $SP = (\Sigma, AX)$ be a swinging specification and Φ, Ψ and Θ be the consequence operators of Def. 4.6. Then $Her(SP)|_{hid\Sigma} = \cup_{i\in\mathbb{N}}\Phi^i(\emptyset)$ and for all ground $hid\Sigma$-atoms p,*

$$Her(SP) \models p \iff \exists i \in \mathbb{N} : \Phi^i(\emptyset) \models p, \tag{1}$$

$$Her(SP) \models p \iff SP \vdash_{cut} p, \tag{2}$$

in particular, for all equality predicates $\equiv$ of SP, $\equiv^{Her(SP)} = \equiv_{SP}$ (cf. 4.1). If Ψ is downward continuous, then $Her(SP)|_{\nu\Sigma} = \cap_{i\in\mathbb{N}}\Psi^i(T_\Sigma)$ and for all ground ν-atoms p,

$$Her(SP) \models p \iff \forall i \in \mathbb{N} : \Psi^i(T_\Sigma) \models p. \tag{3}$$

If Θ is upward continuous, then $Her(SP) = \cup_{i\in\mathbb{N}}\Theta^i(\emptyset)$ and for all ground Σ-atoms p,

$$Her(SP) \models p \iff \exists i \in \mathbb{N} : \Theta^i(\emptyset) \models p. \tag{4}$$

Proof. (1), (3) and (4) follow from Lemma 4.4 and Kleene's fixpoint theorem (cf. Thm. 4.2). Let $\mathcal{C}$ be the class of Herband $hid\Sigma$-structures. The cut calculus is correct w.r.t. $Mod(hidSP)$. Hence for all $A \in \mathcal{C} \cap Mod(hidSP)$ and ground $hid\Sigma$-atoms p,

$$SP \vdash_{cut} p \ \text{ implies } \ A \models p. \tag{5}$$

Let B be Herband $hid\Sigma$-structure with $r^B = \{t \in T_{\Sigma,w} \mid SP \vdash_{cut} r(t)\}$ for all μ-predicates $r : w \in \Sigma$. B satisfies $AX_1 \cup EQ_{hid\Sigma}$. Hence by (5), B is the least element of $\mathcal{C} \cap Mod(hidSP)$ and thus the least fixpoint of Φ. Therefore, $B = Her(SP)|_{hid\Sigma}$, and we conclude (2). ❑

For instance, $\Psi^i(T_\Sigma)$ interprets the predicate $\lambda g.fair(g,s) : stream$ (cf. Ex. 2.8) as the set of all ground INFSEQ-terms representing streams with at least i elements satisfying g.

Herbrand and initial models always exist. Final models, however, presuppose behavioral consistency (cf. Thm. 5.1).

Theorem 4.8 *Let $SP = (\Sigma, AX)$ be a swinging specification (cf. Defs. 3.1 and 4.6).*

(1) $Her(SP) \in Mod_{be}(SP) \cap Mod_{\mu\nu}(SP)$.

(2) For all first-order formulas φ and $\sigma : X \to T_\Sigma$,

$$Her(SP) \models_\sigma \varphi \iff Ini(SP) \models_{nat \circ \sigma} \varphi.$$

(3) $Ini(SP) \in Mod_\equiv(SP) \cap Mod_{\mu\nu}(SP)$.

(4) $Her(SP)$ is initial in $Mod(SP)$.

(5) $Ini(SP)$ is initial in $Mod_\equiv(SP)$.

Proof. (1) Let $AX_\sim$ be the set of behavior axioms for Σ (cf. 2.4(3)). Since both $\equiv_{SP}$ and the equivalence closure of any relation satisfying $AX_\sim$ satisfies $AX_\sim$ and since $\sim_{SP}$ is the greatest solution of $AX_\sim$, both $\equiv_{SP}$ and the equivalence closure of $\sim_{SP}$ are subsets of $\sim_{SP}$. Hence $\sim_{SP}$ is an equivalence relation including $\equiv_{SP}$ and thus $Her(SP) \in Mod_{be}(SP)$. $Her(SP) \in Mod_{\mu\nu}(SP)$ follows directly from the interpretation of predicates in $Her(SP)$.

(2) holds true because $\equiv_{SP}$ is a Σ-congruence. (3) follows from (2) and Prop. 3.3.

Let $A \in Mod_\equiv(SP)$ and Her' be the Herbrand Σ-structure with $r^{Her'} = \{t \in T_\Sigma^+ \mid t^A \in r^A\}$ for all predicates $r \in \Sigma$. Then for all ground Σ-atoms, $Her' \models p$ iff $A \models p$. Hence Her' satisfies AX because A satisfies AX. Since $Her(SP) \in Mod_{\mu\nu}(SP)$, we obtain for all ground μ-atoms $r(t)$,

$$Ini(SP) \models r(t) \overset{(2)}{\Rightarrow} Her(SP) \models r(t) \Rightarrow t \in r^{Her} \Rightarrow t \in r^{Her'} \Leftrightarrow t^A \in r^A \Rightarrow A \models r(t), \tag{6}$$

and for all ground ν-atoms $q(u)$,

$$A \models q(u) \Rightarrow u^A \in q^A \Leftrightarrow u \in q^{Her'} \Rightarrow u \in q^{Her} \Rightarrow Her(SP) \models q(u) \overset{(2)}{\Rightarrow} Ini(SP) \models q(u). \tag{7}$$

(4) follows from (6), (7) and Lemma 3.5(3) because $\equiv$ is a μ-predicate and $Her(SP)$ is a Herbrand structure. (5) follows from (6), (7) and Lemma 3.5(1) because $Ini(SP)$ is reachable, $\equiv$ is a μ-predicate, $\equiv^{Ini}$ is reflexive and A is a structure with $\equiv$-equality. ❏

The choice of $Ini(SP)$ as the standard model of SP has been motivated thoroughly in the literature (cf., e.g., [51, 24, 56]). Initial models reduce the reasoning about data types to inductive theorem proving. Initial semantics neatly complies with functional sorts, polymorphism and parameter specifications (cf. Section 1). If a functional sort $s \to s'$ and the associated application operator $apply : (s \to s') \times s \to s'$ are declared as hidden and a destructor, respectively, the $(s \to s')$-component of behavioral SP-equivalence agrees with the *extensional equivalence* of terms denoting functions:

$$f \sim_{SP,s \to s'} g \iff \forall t \in T_{\Sigma,s} : apply(f,t) \sim_{SP,s} apply(g,t).$$

The fact that $Her(SP)$ interprets μ- and ν-predicates as least resp. greatest relations on T_Σ that satisfy AX is crucial for the soundness of the following proof rules. Let AX_r be the set of axioms for a logical predicate r.

fixpoint induction on r $$\frac{r(x) \Rightarrow \psi}{\exists q : q(x) \Rightarrow \psi \ \wedge \ \bigwedge_{\varphi \in AX_r} \varphi[q/r]} \Updownarrow \quad \text{if } r \in \mu P$$

coinduction on r $$\frac{\psi \Rightarrow r(x)}{\exists q : \psi \Rightarrow q(x) \ \wedge \ \bigwedge_{\varphi \in AX_r} \varphi[q/r]} \Updownarrow \quad \text{if } r \in \nu P$$

The case $r = \sim$ provides a rule for proving behavioral equivalences:

coinduction on $\sim$ $$\frac{\psi \Rightarrow t \sim u}{\exists q : \psi \Rightarrow q(t,u) \ \wedge \ \bigwedge_{\varphi \in AX_\sim} \varphi[q/\sim]} \Updownarrow$$

Note that $AX_\sim$ is the set of behavior axioms of SP (cf. Def. 2.4(3)).

Fixpoint induction is due to Park (cf. [65]). Both rules can be generalized easily from a single predicate r to several predicates $r_1, \ldots, r_n$ such that it admits proving n conjectures $\psi_1 \Leftarrow r_1(x), \ldots, \psi_n \Leftarrow r_n(x)$ resp. $\psi_1 \Rightarrow r_1(x), \ldots, \psi_n \Rightarrow r_n(x)$ simultaneously.

Fixpoint induction deals with conjectures $\psi \Leftarrow r(x)$ stating that ψ holds true for data related to each other by r. r is often the graph of a defined function (see below). Coinduction deals with inverse conjectures $\psi \Rightarrow r(x)$ stating that r holds true for a set of data specified by ψ. In both cases, the conjecture must be given as an implication $\psi \Leftarrow \varphi$. Fixpoint induction is applicable if the premise φ can be specified as a μ-predicate. Coinduction is applicable if the conclusion ψ can be specified as a ν-predicate. Applying the rule eliminates this predicate from the conjecture so that the rules in some way reduce a proof obligation.

Fixpoint induction and coinduction are equivalence transformations. The downward implication $\Downarrow$ holds true because $Her(SP)$ satisfies AX and thus we may define q as r. The upward implication $\Uparrow$ is valid because all solutions of AX_r in r are supersets resp. subsets of the least resp. greatest solution of AX_r that provides the interpretation of r in the Herbrand model. In other words, if $Her(SP)$ satisfies $\varphi[q/r]$ for all $\varphi \in AX_r$, then $q(x) \Leftarrow r(x)$ resp. $q(x) \Rightarrow r(x)$ are also satisfied. Hence the antecedent of fixpoint induction resp. coinduction follows from the succedent $\psi \Leftarrow q(x)$ resp. $\psi \Rightarrow q(x)$.

q is an existentially quantified predicate variable whose value ranges between ψ and r. Choosing q as a proper subset (in the case of induction) resp. superset (in the case of coinduction) of ψ means to **generalize** resp. **co-generalize** ψ. This complies with the intuition that a smaller relation expresses a stronger condition.

If all predicates of SP are static, then the restriction of conduction on $\sim$ to unconditional behavioral equivalences essentially agrees with the proof technique of **hidden coinduction** introduced in [22, 33]. In this case the behavior axioms of SP are congruence axioms except for the first one that expresses the coincidence of $\sim$ and $\equiv$ an visible terms. Moreover, the succedent of conduction on $\sim$ reduces to:

$$\exists q : q(t,u) \ \wedge \bigwedge_{\varphi \in AX_\sim} \varphi[q/\sim].$$

This is also the proof obligation of hidden coinduction: choose a binary relation q that contains the pair (t,u) and satisfies the behavior axioms of SP with $\sim$ replaced by q. If all predicates of SP are static, each of these axioms describes either a congruence property of q or the condition that for all visible terms t, t', $q(t,t')$ holds true only if t and t' are structurally equivalent (cf. Def. 2.4(3)).

While the above rules are correct w.r.t. $Her(SP)$ because the Herbrand model interprets predicates as *least/greatest* relations satisfying their axioms, the following rules are sound because $Her(SP)$ is a fixpoint of the consequence operators constructed from SP. Unfolding an atom $r(t)$ means to apply all axioms for r to $r(t)$ and thus to split the conjecture surrounding $r(t)$ into as many subgoals as there are axioms for r.

Since $Her(SP)$ is a fixpoint of the consequence operators Φ, Ψ and Θ (cf. Def. 4.6), complete unfolding rules are *equivalence* transformations:

μ-atom unfolding $\qquad \dfrac{r(t)}{\bigvee_{i=1}^{n} \exists Z_i : (t \equiv t_i \wedge \varphi_i)} \Updownarrow \quad$ if $r \in \mu P$,

$\{r(t_1) \Leftarrow \varphi_1, \ldots, r(t_n) \Leftarrow \varphi_n\} = AX_r$ and $Z_i = var(r(t_i) \Leftarrow \varphi_i)$

ν-atom unfolding $\qquad \dfrac{r(t)}{\bigwedge_{i=1}^{n} \forall Z_i : (t \equiv t_i \Rightarrow \varphi_i)} \Updownarrow \quad$ if $r \in \nu P$,

$\{r(t_1) \Rightarrow \varphi_1, \ldots, r(t_n) \Rightarrow \varphi_n\} = AX_r$ and $Z_i = var(r(t_i) \Rightarrow \varphi_i)$

Theorem provers usually combine unfolding with term splitting and clash (see below), applied to the equations $t \equiv t_i$. For all $1 \leq i \leq n$ such that term splitting and clash lead to $FALSE$, the summand $t \equiv t_i \wedge \varphi_i)$ resp. factor $t \equiv t_i \Rightarrow \varphi_i)$ can be omitted when the unfolding rule's succedent is constructed without violating the rule's correctness.

In contrast to fixpoint induction and coinduction the application of an unfolding rule to an atom $r(t)$ need not remove r from the rule's antecedent. Whenever some axioms for r are recursive, i.e. contain r in the premise resp. conclusion, these occurrences will appear in the rule's succedent.

Given a predicate r of SP, the fixpoint properties of $Her(SP)$ often lead to (co-)Horn axioms for the complement $\overline{r}$ of r w.r.t. $Her(SP)$ if one simply negates the premises resp. conclusions of the axioms for r. For instance, if $AX_r = \{r(t_1) \Leftarrow \varphi_1, \ldots, r(t_n) \Leftarrow \varphi_n\}$, then the fixpoint property implies that $Her(SP)$ satisfies

$$r(x) \;\Leftrightarrow\; \bigvee_{i=1}^{n} \exists Z_i : (x \equiv t_i \wedge \varphi_i) \quad \text{and thus} \quad \neg r(x) \;\Leftrightarrow\; \bigwedge_{i=1}^{n} \forall Z_i : (x \not\equiv t_i \vee \neg\varphi_i).$$

If this conjunction is equivalent to a goal set $\bigvee_{i=1}^{n} \exists X_i \psi_i$ consisting of atoms and negative literals $\neg q(t)$ such that $q = r$ or SP contains the complement $\overline{q}$ of q, then the Horn clauses

$$\overline{r}(x) \Leftarrow \psi_1[\overline{q}(t)/\neg q(t)], \;\ldots,\; \overline{r}(x) \Leftarrow \psi_n[\overline{q}(t)/\neg q(t)]$$

axiomatize $\overline{r}$ as the complement of r—provided that the extended specification *terminates* ([60], Thm. 10.39; [63], Satz 6.1.9).

If r is a predicate of the visible or hidden level of SP, then $\overline{r}$ can be specified as a ν-predicate, either in terms of r:

$$\overline{r}(x) \;\Rightarrow\; (r(x) \;\Rightarrow\; FALSE),$$

or by grouping the axioms for r as follows:

$$r(t_1), \;\ldots,\; r(t_k), \;\; r(u_1) \Leftarrow (\varphi_1 \wedge q(v_1)), \;\ldots,\; r(u_n) \Leftarrow (\varphi_1 \wedge q(v_n)),$$

and "dualizing" them:

$$\overline{r}(t_1) \Rightarrow FALSE, \;\ldots,\; \overline{r}(t_k) \Rightarrow FALSE, \;\; \overline{r}(u_1) \Rightarrow (\varphi_1 \Rightarrow \overline{q}(v_1)), \;\ldots,\; \overline{r}(u_n) \Rightarrow (\varphi_n \Rightarrow \overline{q}(v_n))$$

([63], Satz 8.3.4). The negation of a ν-predicate r specified by arbitrary co-Horn axioms such as:

$$r(t) \;\Rightarrow\; (G \;\Rightarrow\; (\exists X_1(G_1 \wedge r_1(t_1)) \vee \cdots \vee \exists X_n(G_n \wedge r_n(t_n)))),$$

leads to generalized Horn axioms for $\overline{r}$:

$$\overline{r}(t) \;\Leftarrow\; (G \;\wedge\; \forall X_1(G_1 \Rightarrow \overline{r_1}(t_1)) \wedge \;\cdots \wedge \forall X_n(G_n \Rightarrow \overline{r_n}(t_n))).$$

Axioms for complements is all one needs for refuting conjectures in the Herbrand model. The main inference rule used in a refutation proof is the unfolding of complement atoms $\overline{r}(t)$.

Given a functional specification SP, ground goals over $hidSP$ can be proved in a rewriting-oriented way, by applying Horn axioms as logic programs and reducing goals to $TRUE$. Moreover, instead of applying congruence axioms goal reductions rewrite terms analogously to the way they *resolve* logical atoms.

A **fresh variable** of a Horn clause $\varphi = (r(t) \Leftarrow H)$ (resp. $\varphi = (t \equiv u \Leftarrow H)$) is a variable that occurs in u or H, but not in t. **fresh(φ)** denotes the set of fresh variables of φ. Note that Condition 2.4(a) implies $fresh(\varphi) \subseteq H$ if φ is a Horn axiom of a swinging specification.

Definition 4.9 Let SP be a swinging specification and $hidSP = (\Sigma, AX)$. The **reduction calculus for** SP consists of the following rules for reducing goals. Let G be a Σ-goal and $\sigma : X \rightarrow T_\Sigma(X)$.

rewriting	$\dfrac{G(t\sigma)}{G(u\sigma) \wedge H\sigma} \Uparrow$	if $\varphi = (t \equiv u \Leftarrow H) \in AX$ and $fresh(\varphi)\sigma \subseteq NF_\Sigma(X)$
resolution	$\dfrac{r(t\sigma) \wedge G}{H\sigma \wedge G} \Uparrow$	if $r \not\equiv \equiv$, $\varphi = (r(t) \Leftarrow H) \in AX$ and $fresh(\varphi)\sigma \subseteq NF_\Sigma(X)$
reflection	$\dfrac{t \equiv t \wedge G}{G} \Uparrow$	

A sequence $G_1, \ldots, G_n$ of goals such that for all $1 \leq i < n$, G_{i+1} is obtained from G_i by applying one of the above rules, is called an SP**-reduction of** G_1 **into** G_n and we write $G_1 \vdash_{SP} G_n$. ❑

Definition and Theorem 4.10 (Church-Rosser Theorem) [60] *Let SP be a swinging specification and $hidSP = (\Sigma, AX)$. For all ground Σ-goals G, $G \vdash_{SP} \emptyset$ implies $SP \vdash_{cut} G$. A complete specification SP is functional iff SP is* **confluent**, *i.e. for all ground goals G,*

$$SP \vdash_{cut} G \quad implies \quad G \vdash_{SP} \emptyset. \;❑ \tag{1}$$

Theorem 4.10 also implies that a functional specification can be transformed into an equivalent relational one by turning each defined function into its graph or input-output relation:

Definition 4.11 (flat formula) Let Σ be a swinging signature and Σ' be Σ without defined functions. A first-order Σ-formula φ is **flat** if all logical atoms of φ are Σ'-atoms and for all equations $t \equiv u$ of φ, u is a normal form and either t is a normal form or there are a defined function f and a normal form t' such that $t = f(t')$. The following function *mkflat* transforms a first-order formula φ into an equivalent flat formula $flat(\varphi) = mkflat(\varphi, \emptyset)$:

- $mkflat(p, V) \;=_{def}\; p$ for all flat atoms p,
- $mkflat(r(c(f(t))), V) \;=_{def}\; \exists x \; mkflat(r(c(x)) \wedge x \equiv f(t),\; V \cup \{x\})$ for all non-flat atoms $r(c(f(t)))$ where r is a predicate, f is a defined function, $c(x)$ is a normal form und $x \in X \setminus V$,
- $mkflat(\neg\varphi, V) \;=_{def}\; \neg mkflat(\varphi, V)$,
- $mkflat(\varphi \oplus \psi, V) \;=_{def}\; mkflat(\varphi, V) \oplus mkflat(\psi, V)$ for all $\oplus \in \{\wedge, \vee, \Rightarrow\}$,
- $mkflat(\forall Y \varphi, V) \;=_{def}\; \forall Y \; mkflat(\varphi, V \cup Y)$,

- $mkflat(\exists Y\varphi, V) =_{def} \exists Y\ mkflat(\varphi, V\cup Y)$.

For a set F of formulas, $flat(F) =_{def} \{flat(\varphi) \mid \varphi\in F\}$. Moreover, $rel(\Sigma)$ is obtained from Σ by replacing each defined function $f : w\to s\in\Sigma$ by the **graph** $r_f : ws$ of f. $rel(F)$ is obtained from F by replacing each equation $f(t)\equiv u$ of F with defined function f by the atom $r_f(t,u)$. ❑

For all modal, poly-modal and weakly modal formulas φ, $flat(\varphi)$ is modal, poly-modal or weakly modal, respectively (cf. 2.3).

Definition 4.12 Let $SP = (\Sigma, AX)$ be a swinging specification. The swinging specifications $flat(SP) = (\Sigma, flat(AX))$ and $rel(SP) = (rel(\Sigma), rel(flat(AX)))$ are called the **flat** and **relational versions of** SP, respectively. ❑

The only function symbols of $rel(\Sigma)$ are the constructors of Σ. A visible predicate of $rel(\Sigma)$ is a visible predicate of Σ or the graph of a visible defined function of Σ. A transition predicate of $rel(\Sigma)$ is a transition predicate of Σ or the graph of a destructor of Σ. A dynamic predicate of $rel(\Sigma)$ is a transition predicate of Σ or the graph of a defined function of Σ.

Theorem 4.13 (equivalence of a functional specification and its relational version) *Let SP be a functional and continuous specification. Then $rel(SP)$ is functional and continuous and for all ground Σ-atoms p,*

$$Her(SP)\models p \iff Her(rel(SP))\models rel(flat(p)). \tag{1}$$

Proof. Let $SP = (\Sigma, AX)$ and $rel(SP) = (\Sigma', AX')$. Since relational specifications are functional, $rel(SP)$ is functional. Since the consequence operators Ψ and Θ of Def. 4.6 are downward resp. upward continuous, the corresponding consequence operators Ψ' and Θ' on corresponding reducts of $Her(rel(SP))$ are also downward resp. upward continuous. Hence by Thm. 4.10 and Lemma 4.7(3) and (4), (1) holds true if for all defined functions f, predicates r that are specified on the visible or hidden level, ν-predicates q, predicates p that are specified on the μ-level, $i\in\mathbb{N}$ and $t,u\in NF_\Sigma$,

$$\begin{aligned}
f(t)\equiv u\vdash_{SP}\emptyset &\iff r_f(t,u)\vdash_{rel(SP)}\emptyset, & (2)\\
t\equiv u\vdash_{SP}\emptyset &\iff t\equiv u\vdash_{rel(SP)}\emptyset, & (3)\\
r(t)\vdash_{SP}\emptyset &\iff r(t)\vdash_{rel(SP)}\emptyset, & (4)\\
t\in q^{\Psi^i(T_\Sigma)} &\iff t\in q^{(\Phi')^i(T_\Sigma)}, & (5)\\
t\in p^{\Theta^i(\emptyset)} &\iff t\in p^{(\Theta')^i(\emptyset)}. & (6)
\end{aligned}$$

One may first show (2)-(4) by induction on the length of SP- resp. $rel(SP)$-reductions. Then (5) and (6) follow by induction on i. ❑

Corollary 4.14 *Let SP be a functional and continuous specification. For all first-order formulas formulas φ,*

$$Her(SP)\models\varphi \iff Her(rel(SP))\models rel(flat(\varphi)). \quad ❑$$

Corollary 4.15 *Let SP be a functional and continuous specification. Given a defined function $f\in\Sigma$, $\sim_{SP}$ is compatible with f iff $\sim_{rel(SP)}$ is zigzag compatible with the graph r_f of f.*

Proof. Let $f : w\to s$ and $t,t'\in T_{\Sigma,w}$ such that $t\sim_{SP} t'$ and $\sim_{rel(SP)}$ is zigzag compatible with r_f. By Thm. 4.13, $Ini(SP)\models f(nf(t))\equiv nf(f(t))$ implies $Ini(rel(SP))\models r_f(nf(t), nf(f(t)))$. Since $\equiv_{SP}$ is a subset of $\sim SP$ and $\sim SP$ is transitive, $t \sim_{SP} t'$ implies $nf(t) \sim_{SP} nf(t')$ and thus $nf(t) \sim_{rel(SP)} nf(t')$. Since $\sim_{rel(SP)}$ is zigzag compatible with r_f, there is $u \in NF_\Sigma$ such that $nf(f(t)) \sim_{SP} u$ and $Ini(rel(SP)) \models r_f(nf(t'), u)$. Hence by Thm. 4.13, $Ini(SP) \models f(nf(t')) \equiv u$ and thus $f(t') \equiv_{SP} u$. Since $\equiv_{SP}$ is a subset of $\sim SP$ and $\sim SP$ is transitive, we conclude $f(t) \sim_{SP} f(t')$. Hence $\sim_{SP}$ is compatible with f. The converse can be shown in a similar way. ❑

One of the most useful consequences of Thm. 4.13 is the soundness of fixpoint induction and unfolding for proving inductive theorems about defined functions: if SP is functional, then by Thm. 4.13(1), the following functional counterparts of fixpoint induction on μ-predicates and μ-atom unfolding, respectively, are correct. Let f be a defined function and AX_f be the set of axioms for f (cf. Def. 4.11).

fixpoint induction on f
$$\frac{f(x) \equiv y \Rightarrow \psi}{\exists q : q(x,y) \Rightarrow \psi \ \wedge\ \bigwedge_{\varphi \in flat(AX_f)} \varphi[q(t,u)/(f(t) \equiv u)]} \Updownarrow$$

term unfolding
$$\frac{\varphi(f(t))}{\bigvee_{i=1}^{n} \exists Z_i : (t \equiv t_i \wedge \varphi(u_i) \wedge \psi_i)} \Updownarrow$$

where $\{f(t_1) \equiv u_1 \Leftarrow \psi_1, \ldots, f(t_n) \equiv u_n \Leftarrow \psi_n\} = AX_f$ and $Z_i = var(f(t_i) \equiv u_i \Leftarrow \psi_i)$

A further consequence of functionality is the soundness of rules for removing constructors:

term splitting
$$\frac{c(t_1,\ldots,t_n) \equiv c(u_1,\ldots,u_n)}{t_1 \equiv u_1 \wedge \cdots \wedge t_n \equiv u_n} \Updownarrow \qquad \text{where } c \text{ is a constructor}$$

clash
$$\frac{c(t) \equiv d(u)}{FALSE} \Updownarrow \qquad \text{where } c \text{ and } d \text{ are different constructors}$$

If SP is functional, these equivalences imply that the **standard inequality axioms for** SP specify the complement of $\equiv$ (cf. Def. 3.1):

$$c(x_1,\ldots,x_n) \not\equiv c(y_1,\ldots,y_n) \ \Leftarrow\ x_i \not\equiv y_i \qquad \text{for all constructors } c : s_1 \ldots s_n \to s \text{ and } 1 \le i \le n,$$
$$c(x) \not\equiv d(y) \qquad \text{for all different constructors } c \text{ and } d.$$

Repeated applications of term splitting and clash remove an equation $t \equiv t'$ iff t and t' are ground normal forms. For eliminating equations with variables one also needs *term replacement*:

term replacement
$$\frac{t \equiv t' \wedge \varphi(t)}{\varphi(t')} \Downarrow$$
$$\frac{(t \equiv t' \wedge \varphi(t)) \ \Rightarrow\ \psi(t)}{\varphi(t') \ \Rightarrow\ \psi(t')} \Uparrow$$

5 The final model and hierarchy conditions

Final semantics was introduced for modelling **permutative types** such as finite sets, finite bags (multisets) and arrays with a finite domain (cf., e.g., [28, 75, 46]). These types are constructor-based, but need equations between normal forms for axiomatizing structural equality. Hence specifications of permutative types are complete, but not consistent (cf. 4.1). From a model-theoretic point of view, initial semantics is sufficient for handling permutative types. Normal form equations are Horn axioms, hence there is an initial model. From a proof-theoretic viewpoint, however, this model is inadequate. Resolution- or rewriting-oriented proof methods treat normal form equations separately from other axioms (cf., e.g., [66, 72, 45]). Here it is not the normal forms, but their equivalence classes modulo the equivalence relation $\equiv_E$ induced by the set E of normal form equations that represent data. Resolution and rewriting modulo $\equiv_E$ work well if E is restricted to particular equations such as those expressing the associativity, commutativity or idempotence of a binary function. Otherwise suitable proof rules are complicated and difficult to handle.

In the swinging specification of a permutative type, normal form equations $t \equiv t'$ come as valid behavioral equivalences $t \sim t'$. Results on coalgebras, coinduction and greatest fixpoints obtained in category theory and modal logic revealed that permutative types are particular hidden types and thus can be handled with the same approaches that tackle state-oriented *object types* and *infinite types* such as streams and processes (cf., e.g., [8, 29, 68, 70, 32, 44]). Vice versa, these types extend the range of applications for final-semantics approaches. As we have seen in Ex. 2.8, even streams can be presented as a *functional* specification (cf. 4.1). At first sight, this seems to be inadequate because functionality includes completeness, while uncountably many streams cannot be represented by countably many normal forms. But it need not bother us since uncountable sets can never be implemented entirely. The fact that the final model is *embedded* in the intended domain is completely sufficient for any formal reasoning about the type. The existence of an embedding is usually guaranteed if the specification, say SP, is behaviorally consistent (cf. [62], Section 6). Hence the final model of a behaviorally consistent extension of SP by more hidden constructors will also be embedded in the intended domain. For instance, if SP = INFSEQ (cf. Ex. 2.8), then $Fin(SP)_{stream}$ is embedded in $[\mathbb{N} \to Ini(SP)_{entry}]$, and, if SP = STREAM (cf. Ex. 6.6), then $Fin(SP)_{stream}$ is embedded in $[\mathbb{N} \to Ini(SP)_{entry}] \cup Ini(SP)^*_{entry}$.

Theorem 5.1 *Let $SP = (\Sigma, AX)$ be a behaviorally consistent specification (cf. Defs. 3.1 and 4.6).*

(1) $Her(SP) \in Mod_{bcr}(SP)$ and thus by Thm. 3.9(b), $Fin(SP) \in Mod(SP)$.

(2) For all poly-modal formulas φ and $\sigma : X \to T_\Sigma$,

$$Her(SP) \models_\sigma \varphi \iff Fin(SP) \models_{nat \circ \sigma} \varphi.$$

(3) $Fin(SP) \in Mod_{bcr}(SP) \cap Mod_{\mu\nu}(SP)$ (cf. Def. 4.6).

(4) If SP is visible, then $Fin(SP)$ coincides with $Ini(SP)$.

(5) $Fin(SP)$ is final in $Mod_{bcr}(SP)$.

Proof. (1) holds true by assumption and since $Her(SP)$ is reachable. (2) follows from Thm. 3.9(a).

(3) Since $Her(SP)$ is behaviorally consistent, $Fin(SP)$ is an SP-model. $\sim^{Fin}$ is equality and hence a weak congruence. Since $Fin(SP)$ is reachable, we conclude $Fin(SP) \in Mod_{bcr}(SP)$ from the interpretation of dynamic predicates in $Fin(SP)$. Since $Fin(SP) \in Mod(SP)$, Prop. 3.3 implies $Fin(SP) \in Mod_{\mu\nu}(SP)$.

(4) holds true because $\sim_{SP}$ agrees with $\equiv_{SP}$ on visible terms.

(5) Let $A \in Mod_{bcr}(SP)$ and Her' be the Herbrand Σ-structure defined by $r^{Her'} = \{t \in T^+_\Sigma \mid t^A \in r^A\}$ for all predicates $r \in \Sigma$. Then for all ground Σ-atoms, $Her' \models p$ iff $A \models p$. Hence Her' satisfies AX because A satisfies AX. By Thm. 4.8(1), $Her(SP) \in Mod_{\mu\nu}(SP)$. Hence for all ground ν-atoms $q(u)$,

$$A \models q(u) \;\Rightarrow\; u^A \in q^A \;\Leftrightarrow\; u \in q^{Her'} \;\Rightarrow\; u \in q^{Her} \;\Rightarrow\; Her(SP) \models q(u) \overset{(2)}{\Rightarrow} Fin(SP) \models q(u), \tag{6}$$

and for all ground static μ-atoms $r(t)$ and ground dynamic atoms $\delta(t,u)$,

$$Fin(SP) \models r(t) \overset{(2)}{\Rightarrow} Her(SP) \models r(t) \;\Rightarrow\; t \in r^{Her} \;\Rightarrow\; t \in r^{Her'} \;\Leftrightarrow\; t^A \in r^A \;\Rightarrow\; A \models r(t), \tag{7}$$

$$\begin{aligned} &Fin(SP) \models \delta(t,u) \;\Rightarrow\; \exists v : Her(SP) \models \delta(t,v) \wedge v \sim_{SP} u \;\Rightarrow\; (t,v) \in \delta^{Her} \wedge v \sim_{SP} u \\ &\Rightarrow\; (t,v) \in \delta^{Her'} \wedge v \sim_{SP} u \;\Leftrightarrow\; (t^A, v^A) \in \delta^A \wedge v \sim_{SP} u \;\Rightarrow\; A \models \delta(t,v) \wedge v^{Fin} = u^{Fin}. \end{aligned} \tag{8}$$

Since $\sim^A$ is reflexive, (5) follows from (6), (7) and Lemma 3.5(2) because $\sim$ is a ν-predicate, A is reachable and $Fin(SP)$ is a structure with $\sim$-equality. ❏

Behavioral consistency ensures the existence of the final model. This is a model-theoretic side-effect of behavioral consistency, but not its most significant consequence. More important in practice is the fact that behavioral consistency ensures that—due to our Hennessy-Milner Theorem 3.8(3)—*behavioral term replacement* is sound for poly-modal formulas as term replacement is sound for arbitrary first-order formulas (see Section 4):

behavioral term replacement $\dfrac{t \sim t' \wedge \varphi(t)}{\varphi(t')} \Downarrow$ if φ is poly-modal

$\dfrac{(t \sim t' \wedge \varphi(t)) \;\Rightarrow\; \psi(t)}{\varphi(t') \;\Rightarrow\; \psi(t')} \Uparrow$ if φ and ψ are poly-modal

Example 5.2 Suppose that for some hidden sort s there are neither separators $r : sw$ nor transition predicates $\delta : sws'$ and all destructors $f : sw \to s'$ are *methods*, i.e. s' is a hidden sort. Then $\sim_{SP,s}$ cover all pairs of ground s-terms and thus $Fin(SP)_s$ is a singleton! For instance, consider the following swinging specification of integer numbers:

INT

`hidsorts`	int		
`constructs`	$0, 1 :\to int$		
	$\_+\_ : int \times int \to int$		
	$\_-\_ : int \times int \to int$		
`destructs`	$succ, pred : int \to int$		
`separators`	$is0 : int$		
`vars`	$x, y : int$		
`Horn axioms`	$succ(0) \equiv 1$	$pred(0) \equiv 0-1$	$is0(0)$
	$succ(1) \equiv 1+1$	$pred(1) \equiv 0$	
	$succ(x+y) \equiv succ(x)+y$	$pred(x+y) \equiv pred(x)+y$	
	$succ(x-y) \equiv succ(x)-y$	$pred(x-y) \equiv pred(x)-y$	

The final INT-model $Fin(\text{INT})$ is isomorphic to $\mathbb{Z}$. The "normal form equations" $(x+y)-y \sim x$ and $(x-y)+y \sim x$ are inductive theorems of INT. If the separator $is0$ were omitted, behavioral INT-equivalence would identify all ground INT-terms, i.e. $Fin(\text{INT})$ were a singleton. ❏

Definition 5.3 (relative completeness and consistency) Let SP and SP' be swinging specifications and $\sigma : \Sigma \to \Sigma'$ be a signature morphism. SP' is **complete w.r.t.** (SP, σ) if for all sorts $s \in \Sigma$ and $t' \in T_{\Sigma',\sigma(s)}$ there is $t \in T_\Sigma$ such that $t' \equiv_{SP} \sigma(t)$. SP' is **monotone w.r.t.** (SP, σ) if for all ground μ-atoms p,

$$Her(SP) \models p \quad\Rightarrow\quad Her(SP')_\sigma \models p, \tag{1}$$

and for all ground ν-atoms p,

$$Her(SP')_\sigma \models p \quad\Rightarrow\quad Her(SP) \models p. \tag{2}$$

SP' is **(relatively) consistent w.r.t.** (SP, σ) if, conversely, (1) holds true for all ground ν-atoms p and (2) holds true for all ground μ-atoms p. If σ is an inclusion, i.e. $\Sigma \subseteq \Sigma'$, we write SP instead of (SP, σ). ❏

Proposition 5.4 *If SP' is monotone w.r.t. (SP,σ), then for all $t,t' \in T_\Sigma$, $t \equiv_{SP} t'$ implies $\sigma(t) \equiv_{SP'} \sigma(t')$ and $t \not\sim_{SP} t'$ implies $\sigma(t) \not\sim_{SP'} \sigma(t')$. If SP' is complete, monotone and consistent w.r.t. (SP,σ), then for all first-order Σ-formulas φ, $Her(SP')_\sigma \models \varphi$ iff $Her(SP) \models \varphi$.* ❑

Definition 5.5 (inductive equivalence) Let SP and SP' be swinging specifications with the same signature Σ. SP and SP' are **inductively equivalent** if SP' is monotone and consistent w.r.t. SP, or equivalently: for all ground Σ-atoms p, $Her(SP) \models p$ iff $Her(SP') \models p$. ❑

Proposition 5.6 *Let SP and SP' be inductively equivalent specifications with signature Σ.*

(1) For all first-order Σ-formulas φ, $Her(SP) \models \varphi$ iff $Her(SP') \models \varphi$.

(2) SP is functional iff SP' is functional.

(3) SP is behaviorally consistent iff SP' is behaviorally consistent.

Proof. (1) follows from Prop. 5.4. By assumption, (behavioral) SP-equivalence coincides with (behavioral) SP'-equivalence. This implies (2) and (3). (3) also relies upon the inductive equivalence of SP and SP' with respect to other predicates of Σ. ❑

Lemma 5.7 *Let $SP = (\Sigma, AX)$ and $SP' = (\Sigma', AX')$ be swinging specifications, $\Sigma = (S,F,P)$, $\Sigma' = (S',F',P')$, $\sigma : \Sigma \to \Sigma'$ be a signature morphism and C be a set of generalized Horn clauses or co-Horn clauses over Σ'.*

(1) If $Her(SP')_\sigma$ is an SP-model, then SP' is monotone w.r.t. (SP,σ). In particular, $SP' \cup C = (\Sigma', AX \cup C)$ is monotone w.r.t. SP'.

(2) $Her(SP') \models C$ iff $SP' \cup C$ is consistent w.r.t. SP'.

(3) Let $\Sigma = \Sigma'$. SP and SP' are inductively equivalent iff $Her(SP) \models AX'$ and $Her(SP') \models AX$.

Proof. (1) By assumption, $Her(SP')_\sigma$ satisfies $EQ_\Sigma \cup AX$. Since $Her(SP)$ is the least solution of the Horn axioms among $EQ_\Sigma \cup AX$, for all $r \in \mu P$, $r^{Her(SP)}$ is a subset of $r^{Her(SP')_\sigma}$. Since $Her(SP)$ is the greatest solution of the co-Horn axioms among $EQ_\Sigma \cup AX$, for all $r \in \nu P$, $r^{Her(SP')_\sigma}$ is a subset of $r^{Her(SP)}$.

(2) "$\Rightarrow$": Let μC and νC be the set of Horn resp. co-Horn clauses of $SP' \cup C$. Since $Her(SP' \cup C)$ is the least solution of $EQ_{\Sigma'} \cup \mu C$, $Her(SP') \models EQ_{\Sigma'} \cup \mu C$ implies that for all $r \in \mu P'$, $r^{Her(SP' \cup C)}$ is a subset of $r^{Her(SP')}$. Since $Her(SP' \cup C)$ is the greatest solution of νC, $Her(SP') \models \nu C$ implies that for all $r \in \nu P'$, $r^{Her(SP')}$ is a subset of $r^{Her(SP' \cup C)}$. Hence $SP' \cup C$ is consistent w.r.t. SP'.

"$\Leftarrow$": Let $p \Leftarrow H$ be a Horn clause of C and $\sigma : X \to T_{\Sigma'}$ such that $Her(SP') \models H\sigma$. By (1) and assumption, $Her(SP' \cup C) \models H\sigma$ and thus $Her(SP' \cup C) \models p\sigma$. Again by (1) and assumption, $Her(SP') \models p\sigma$.

Let $p \Rightarrow (H \Rightarrow \varphi)$ be a co-Horn clause of C and $\sigma : X \to T_{\Sigma'}$ such that $Her(SP') \models p\sigma \wedge H\sigma$. By (1) and assumption, $Her(SP' \cup C) \models p\sigma \wedge H\sigma$ and thus $Her(SP' \cup C) \models \varphi\sigma$. Again by (1) and assumption, $Her(SP') \models \varphi\sigma$.

(3) "$\Leftarrow$": Let p be a μ-atom such that $Her(SP') \models p$. (1) implies $Her(SP \cup AX') = Her(SP' \cup AX) \models p$. Since $Her(SP)$ satisfies AX', (2) implies $Her(SP) \models p$. Conversely, let $Her(SP) \models p$. (1) implies $Her(SP' \cup AX) = Her(SP \cup AX') \models p$. Since $Her(SP')$ satisfies AX, (2) implies $Her(SP') \models p$.

Let p be a ν-atom such that $Her(SP') \models p$. Since $Her(SP')$ satisfies AX, (2) implies $Her(SP \cup AX') = Her(SP' \cup AX) \models p$. By (1), $Her(SP) \models p$. Conversely, let $Her(SP) \models p$. Since $Her(SP)$ satisfies AX', (2) implies $Her(SP' \cup AX) = Her(SP \cup AX') \models p$. By (1), $Her(SP') \models p$.

"⇒": Follows from Prop. 5.6(1) if one sets first $\varphi = AX'$ and then $\varphi = AX$. ❑

Corollary 5.8 (negation and consistency) *In addition to the assumptions of Lemma 5.7 suppose that for each predicate $r \in \Sigma$, the complement $\overline{r}$ of r w.r.t. $Her(SP)$ is in Σ, $\sigma(\overline{r})$ is the complement of $\sigma(r)$ w.r.t. $Her(SP')$ and $r \in \mu P$ implies $\overline{r} \in \mu P$ or for all $t \in T_\Sigma$,*

$$Her(SP')_\sigma \models r(t) \quad \Rightarrow \quad Her(SP) \models r(t). \tag{1}$$

SP' is consistent w.r.t. (SP, σ) if $Her(SP')_\sigma$ is an SP-model.

Proof. Let $r(t)$ be a Σ-atom such that r is a μ-predicate and $Her(SP) \not\models r(t)$. Then $Her(SP) \models \overline{r}(t)$. Let $\overline{r}$ be a μ-predicate. Since $Her(SP')_\sigma$ satisfies $EQ_\Sigma \cup AX$ and $Her(SP)$ is the least solution of the Horn axioms among $EQ_\Sigma \cup AX$, we obtain $Her(SP')_\sigma \models \overline{r}(t)$ and thus

$$Her(SP') \;\models\; \sigma(\overline{r}(t)) \;=\; \sigma(\overline{r})(\sigma(t)) \;=\; \overline{\sigma(r)}(\sigma(t)).$$

We conclude $Her(SP') \not\models \sigma(r)(\sigma(t)) = \sigma(r(t))$. Hence $Her(SP')_\sigma \not\models r(t)$. Conversely, we have shown (1). If $\overline{r}$ is a ν-predicate, then (1) holds true by assumption.

Let $r(t)$ be a Σ-atom such that r is a ν-predicate and $Her(SP')_\sigma \not\models r(t)$, i.e. $Her(SP') \not\models \sigma(r(t)) = \sigma(r)(\sigma(t))$. Then

$$Her(SP') \;\models\; \overline{\sigma(r)}(\sigma(t)) \;=\; \sigma(\overline{r})(\sigma(t)) \;=\; \sigma(\overline{r}(t))$$

and thus $Her(SP')_\sigma \models \overline{r}(t)$. If $\overline{r}$ is a μ-predicate, then, by the first part of the proof, (1) holds true. Hence $Her(SP) \models \overline{r}(t)$ and thus $Her(SP) \not\models r(t)$. Let $\overline{r}$ be a ν-predicate. Since $Her(SP')_\sigma$ satisfies $EQ_\Sigma \cup AX$ and $Her(SP)$ is the greatest solution of the co-Horn axioms among $EQ_\Sigma \cup AX$, $Her(SP')_\sigma \models \overline{r}(t)$ implies $Her(SP) \models \overline{r}(t)$ and thus $Her(SP) \not\models r(t)$. ❑

Lemma 5.9 *A functional specification SP is complete, monotone and consistent w.r.t. $visSP$, $hidSP$ and νSP.*

Proof. Let $SP = (\Sigma, AX)$. By Thms. 4.8 and 5.1, $Her(SP)$, $Ini(SP)$ and $Fin(SP)$ are reachable SP-models. Since SP is complete and for all sorts $s \in visSP$, s-sorted normal forms are $vis\Sigma$-terms, SP is complete w.r.t. $visSP$, $hidSP$ and νSP.

Since all predicates of $vis\Sigma$ are μ-predicates, Lemma 5.7(1) implies that $hidSP$ is monotone w.r.t. $visSP$. Since SP is functional and all hidden constructors have hidden range sorts, [60], Thm. 10.48(3) implies that $hidSP$ is consistent w.r.t. $visSP$. νSP is monotone and consistent w.r.t. $hidSP$ because $\nu AX \setminus hidAX$ consists of axioms for $\nu\Sigma \setminus hid\Sigma$. SP is monotone and consistent w.r.t. νSP because $AX \setminus \nu AX$ consists of axioms for $\Sigma \setminus \nu\Sigma$. Hence SP is monotone and consistent w.r.t. $visSP$, $hidSP$ and νSP. ❑

Consistency criteria based on confluence (cf. Thm. 4.10) are provided by, e.g., [60], Thm. 10.48 (see also [63]). Lemma 5.9 suggests a stepwise construction of $Her(hidSP)$ via a consequence operator on $Her(visSP)$ (cf. Lemma 4.7):

Lemma 5.10 (stepwise construction of $Her(hidSP)$ on $Her(visSP)$) *Let $SP = (\Sigma, AX)$ be a functional specification, $\Sigma = (S, F, P)$ and A be the Herbrand $(vis\Sigma \cup F)$-structure that is defined as follows:*

- *For all predicates $r : w \in vis\Sigma$ and $t \in T_{\Sigma,w}$,* $\quad t \in r^A \iff_{def} Her(visSP) \models r(nf(t))$.

Let Γ be the $hidAX$-consequence operator on A. Then $Her(SP)|_{hid\Sigma} = \cup_{i \in \mathbb{N}} \Gamma^i(\emptyset)$ and thus for all $hid\Sigma$-atoms p,

$$Her(SP) \models p \iff \exists i \in \mathbb{N} : \Gamma^i(\emptyset) \models p.$$

Proof. By Lemma 4.4(1), $lfp(\Gamma) = \cup_{i\in\mathbb{N}}\Gamma^i(\emptyset)$. As a fixpoint of Γ, $lfp(\Gamma)$ satisfies $hidAX \setminus visAX$. Since $Her(SP)|_{hid\Sigma}$ is the least fixpoint satisfying $hidAX$, for all $hid\Sigma$-atoms p,

$$Her(SP) \models p \;\Rightarrow\; lfp(\Gamma) \models p \;\iff\; \exists i \in \mathbb{N} : \Gamma^i(\emptyset) \models p.$$

By Lemma 5.9, $Her(SP)$ is consistent w.r.t. $Her(visSP)$. Hence for all visible $hid\Sigma$-atoms $r(t)$ and $i \in \mathbb{N}$,

$$Her(SP) \models r(t) \iff Her(SP) \models r(nf(t)) \iff Her(visSP) \models r(nf(t)) \iff t \in r^A = r^{\Gamma^i(\emptyset)}. \quad (1)$$

Therefore, it remains to show that for all $i \in \mathbb{N}$ and hidden $hid\Sigma$-atoms $r(t)$,

$$t \in r^{\Gamma^i(\emptyset)} \;\Rightarrow\; Her(SP) \models r(t). \quad (2)$$

Since $r^\emptyset = \emptyset$, (2) holds true for $i = 0$. Let $i > 0$ and $t \in r^{\Gamma^i(\emptyset)}$. By the definition of Γ, there are $(r(u) \Leftarrow H) \in hidAX \setminus visAX$ and $\sigma : X \to T_\Sigma$ such that $t = u\sigma$ and $\Gamma^{i-1}(\emptyset) \models H\sigma$. By induction hypothesis, (2) holds true for $i-1$. Hence by (1), $\Gamma^{i-1}(\emptyset) \models H\sigma$ implies $Her(SP) \models H\sigma$ and thus $Her(SP) \models r(u\sigma) = r(t)$ because $Her(SP)$ satisfies $r(u) \Leftarrow H$. ❑

Let $SP = (\Sigma, AX)$ be a continuous specification and Ψ, Θ as in Def. 4.6. By Lemma 4.7(3) and (4),

$$\text{for all ground } \nu\text{-atoms } p,\quad Her(SP) \models p \iff \forall i \in \mathbb{N} : \Psi^i(T_\Sigma) \models p,$$
$$\text{for all ground } \Sigma\text{-atoms } p,\quad Her(SP) \models p \iff \exists i \in \mathbb{N} : \Theta^i(\emptyset) \models p.$$

How can the downward resp. upward continuity of Ψ resp. Θ be violated? Remember that Ψ is induced by co-Horn axioms, while Θ is induced by generalized Horn axioms. Suppose that $r(x) \Rightarrow \exists y q(x,y)$ is the only axiom for some predicate $r \in \nu\Sigma \setminus hid\Sigma$. Ψ is downward continuous if for all decreasing chains $\{B_i\}_{i\in\mathbb{N}}$ of $\nu\Sigma$-structures whose $hid\Sigma$-reduct agrees with $Her(SP)|_{hid\Sigma}$, $r^{\cap_i\Psi(B_i)}$ is a subset of $r^{\Psi(\cap_i B_i)}$. But this means that $\forall i \exists y : q^{B_i}(x,y)$ implies $\exists y \forall i : q^{B_i}(x,y)$, which, obviously, need not hold true. Dually, suppose that $r(x) \Leftarrow \forall y q(x,y)$ is the only axiom for some predicate $r \in \Sigma \setminus \nu\Sigma$. Θ is upward continuous if for all increasing chains $\{B_i\}_{i\in\mathbb{N}}$ of Σ-structures whose $\nu\Sigma$-reduct agrees with $Her(SP)|_{\nu\Sigma}$, $r^{\Theta(\cup_i B_i)}$ is a subset of $r^{\cup_i\Theta(B_i)}$. But this means that $\forall y \exists i : q^{B_i}(x,y)$ implies $\exists i \forall y : q^{B_i}(x,y)$, which need not hold true either.

Hence existential quantifiers in the conclusions of co-Horn axioms and universal quantifiers in the premises of generalized Horn axioms may violate the continuity of a swinging specification. Modal logic suggests a sufficient condition on quantified subformulas to ensure continuity. If such a formula is modal in the sense of Def. 2.3, it only occurs in one of the following forms:

$$\exists y(\delta(t(x),y) \wedge \varphi(y)) \quad \text{or} \quad \forall y(\delta(t(x),y) \Rightarrow \varphi(y))$$

where δ is a dynamic predicate. Modal logic would call δ *finitely branching* or *image finite* if for all ground terms u there are only finitely many ground terms v such that $\delta(u,v)$ holds true. The generalization of image finiteness to arbitrary existential or universal goals in the sense of Def. 2.3 leads to the following definition:

Definition 5.11 (image finiteness) Let $SP = (\Sigma, AX)$ be a swinging specification. Given a Σ-goal G,

$$\mathcal{S}(G) \;=_{def}\; \{\tau : var(G) \to NF_\Sigma \mid Her(SP) \models G\tau\}$$

is the set of **normal form solutions of** G. Given $Y \subseteq X$, G is Y**-image finite** if for all $\sigma : X \to NF_\Sigma$, $\mathcal{S}(G\sigma_{X\setminus Y})$ is finite.

An existential $\nu\Sigma$-goal $\exists Y\varphi$ is **image finite** if φ is a $hid\Sigma$-goal or φ contains a nonempty Y-image finite $hid\Sigma$-goal. A universal Σ-goal $\forall Y(G \Rightarrow H)$ is **image finite** if G and H are $\nu\Sigma$-goals or G is a Y-image finite $\nu\Sigma$-goal.

A (dual) goal set is **image finite** if it consists of image finite existential (resp. universal) goals (cf. Def. 2.3). SP is **image finite** if for all co-Horn axioms $p \Rightarrow (G \Rightarrow \varphi)$ of SP, φ is an image finite goal set and for all generalized Horn axioms $p \Leftarrow \varphi$ of SP, φ is an image finite dual goal set. ❑

MODSPEC (cf. Ex. 2.7) is image finite if the dynamic predicate $\rightarrow$: $state \times state$ is finitely branching: for all $t \in T_\Sigma$ there are at most finitely many $t' \in T_\Sigma$ such that $Her(\text{MODSPEC})$ satisfies $t \rightarrow t'$.

Example 5.12 INFSEQ (cf. Ex. 2.8) is image finite. However, the conclusion of the following axiom for $fair$ is not image finite:

$$fair(g,s) \;\Rightarrow\; \exists n,s' : (nthtail(n,s) \equiv s' \wedge g(head(s')) \equiv true \wedge fair(tail(s'))).$$

In terms of Def. 5.11, $G = (nthtail(n,s) \equiv s' \wedge g(head(s')) \equiv true)$ and $H = fair(tail(s'))$. The existential goal $\exists n,s' : (G \wedge H)$ is not image finite because there are streams t such that $G[t/s]$ has infinitely many normal form solutions. However, let $G' = (G \wedge forall(not \circ g, firstn(n,s)))$. Then $\exists n,s' : (G' \wedge H)$ is image finite because for all streams t, $G'[t/s]$ has at most one normal form solution. ❑

Before presenting the general proof that image finiteness implies continuity let us illustrate the essential points at the ν-predicate p = *some_infinite* and the μ-predicate q = *all_finite* of Ex. 2.7. We recall the axioms for p and q:

$$\begin{array}{rcl} p(s) & \Rightarrow & \exists s'(s \rightarrow s' \wedge p(s')) \\ q(s) & \Leftarrow & \forall s'(s \rightarrow s' \Rightarrow q(s')) \end{array}$$

The corresponding consequence operators, say Ψ and Θ, are defined as follows: For all subsets S of $T_{\Sigma,state}$,

$$\begin{array}{rcl} \Psi(S) & =_{def} & \{s \in T_{\Sigma,state} \mid \exists s'(s \rightarrow s' \wedge s' \in S)\}, \\ \Theta(S) & =_{def} & \{s \in T_{\Sigma,state} \mid \forall s'(s \rightarrow s' \Rightarrow s' \in S)\}. \end{array}$$

Let $\rightarrow$ be image finite. We show that Ψ is downward continuous. This holds true iff for all decreasing chains $\{S_i\}_{i \in \mathbb{N}} \subseteq T_{\Sigma,state}$,

$$\forall i \exists s' : (s \rightarrow s' \wedge s' \in S_i) \;\Longrightarrow\; \exists s' \forall i : (s \rightarrow s' \wedge s' \in S_i). \tag{1}$$

Indeed, (1) is valid:

$$\begin{array}{lll} \forall i \exists s'_i : (s \rightarrow s'_i \wedge s'_i \in S_i) & \overset{\rightarrow \text{ is image finite}}{\Longrightarrow} & \exists s' : |\{i \mid s' = s'_i\}| = \omega \;\Longrightarrow\; \forall i \exists j_i \geq i : s'_{j_i} = s' \\ & \overset{S_{j_i} \subseteq S_i}{\Longrightarrow} & \exists s' \forall i : (s \rightarrow s' \wedge s' \in S_i). \end{array}$$

We show that Θ is upward continuous. This holds true iff for all increasing chains $\{S_i\}_{i \in \mathbb{N}} \subseteq T_{\Sigma,state}$,

$$\forall s' \exists i : (s \rightarrow s' \Rightarrow s' \in S_i) \;\Longrightarrow\; \exists i \forall s' : (s \rightarrow s' \Rightarrow s' \in S_i). \tag{2}$$

Since $\rightarrow$ is image finite and $\{S_i\}$ is increasing, there is $m \in \mathbb{N}$ such that for all s', if $s \rightarrow s' \in S_i$ for some i, then $s' \in S_m$. Hence (2) is obtained as follows:

$$\forall s' \exists i : (s \rightarrow s' \Rightarrow s' \in S_i) \;\overset{S_i \subseteq S_m}{\Longrightarrow}\; \forall s' : (s \rightarrow s' \Rightarrow s' \in S_m) \;\Longrightarrow\; \exists i \forall s' : (s \rightarrow s' \Rightarrow s' \in S_i).$$

Lemma 5.13 *Given a complete specification SP and the notations of Def. 5.11, let $\mathcal{C}$ be the class of $\nu\Sigma$-structures whose $hid\Sigma$-reduct agrees with $A = Her(SP)|_{hid\Sigma}$, $B_0 \supseteq B_1 \supseteq B_2 \supseteq \cdots \in \mathcal{C}$ and φ be (1) an image finite existential goal or (2) an image finite goal set over $\nu\Sigma$. Then for all $b : X \rightarrow A$,*

$$\forall i \in \mathbb{N} : B_i \models_b \varphi \quad \textit{implies} \quad \cap_{i \in \mathbb{N}} B_i \models_b \varphi.$$

Proof. (1) Let $\varphi = \exists Y(G \wedge H)$ be an existential goal and $b : X \to A$ such that for all $i \in \mathbb{N}$, $B_i \models_b \varphi$. If G and H are $hid\Sigma$-goals, then $A \models_b \varphi$ and thus $\cap_{i\in\mathbb{N}} B_i \models_b \varphi$ follows immediately. Let G be a Y-image finite $hid\Sigma$-goal. Then for all $\sigma : X \to NF_\Sigma$ with $dom(\sigma) = X \setminus Y$, $\mathcal{S}(G\sigma)$ is finite. Since SP is complete, for all $i \in \mathbb{N}$ there is $\tau^i : X \to NF_\Sigma$ such that $B_i \models_b (G \wedge H)\tau^i$, $dom(\tau^i) = Y$ and $b(x) \equiv_{SP} \tau^i(x)$ for all $x \in Y$. Since G is a $hid\Sigma$-goal, we obtain $A \models G\sigma\tau^i$ for some $\sigma : X \to NF_\Sigma$ with $dom(\sigma) = X \setminus Y$ and $b(x) \equiv_{SP} \sigma(x)$ for all $x \in X \setminus Y$. Since $\mathcal{S}(G\sigma)$ is finite, there is $\tau : X \to NF_\Sigma$ such that $dom(\tau) = Y$ and $\tau = \tau^i$ for infinitely many i. Hence for all $i \in \mathbb{N}$ there is $j_i \geq i$ such that $\tau^{j_i} = \tau$.[10] Since for all $i \in \mathbb{N}$, $B_{j_i} \models_b (G \wedge H)\tau^{j_i}$, we conclude that for all $i \in \mathbb{N}$, $B_{j_i} \models_b (G \wedge H)\tau$ and thus $B_i \models_b (G \wedge H)\tau$ because $B_{j_i} \subseteq B_i$. Hence $\cap_{i\in\mathbb{N}} B_i \models_b \varphi$.

(2) Let $\varphi = (\varphi_1 \vee \cdots \vee \varphi_n)$ be an image finite goal set and $b : X \to A$ such that for all $i \in \mathbb{N}$, $B_i \models_b \varphi$. We show $\cap_{i\in\mathbb{N}} B_i \models_b \varphi$ by induction on n. If $n = 1$, then the conjecture follows from (1). Otherwise let $\psi = (\varphi_2 \vee \cdots \vee \varphi_n)$. If for all $i \in \mathbb{N}$, $B_i \models_b \psi$, then by induction hypothesis, $\cap_{i\in\mathbb{N}} B_i \models_b \psi$ and thus $\cap_{i\in\mathbb{N}} B_i \models_b \varphi$. Otherwise $B_i \not\models_b \psi$ for some $i \in \mathbb{N}$. Let $k = min\{i \mid B_i \not\models_b \psi\}$. Since $\{B_i\}$ is decreasing, we have $B_i \not\models_b \psi$ and thus $B_i \models_b \varphi_1$ for all $i \geq k$. Hence by (1), $\cap_{i\geq k} B_i \models_b \varphi_1$. Since for all $0 \leq i < k$, $B_i \models_b \psi$ and thus $B_i \models_b \varphi$, we conclude $\cap_{i\in\mathbb{N}} B_i = B_0 \cap \cdots \cap B_{k-1} \cap (\cap_{i\geq k} B_i) \models_b \varphi$. ❑

Lemma 5.14 *Given a complete specification SP and the notations of Def. 5.11, let $\mathcal{C}$ be the class of Σ-structures whose $\nu\Sigma$-reduct agrees with $A = Her(SP)|_{\nu\Sigma}$, $B_0 \subseteq B_1 \subseteq B_2 \subseteq \cdots \in \mathcal{C}$ and φ be (1) an image finite universal goal or (2) an image finite dual goal set over Σ. For all $b : X \to A$,*

$$\cup_{i\in\mathbb{N}} B_i \models_b \varphi \quad \textit{implies} \quad \exists i \in \mathbb{N} : B_i \models_b \varphi.$$

Proof. (1) Let $\varphi = \forall Y(G \Rightarrow H)$ be an image finite universal goal and $b : X \to A$ such that $\cup_{i\in\mathbb{N}} B_i \models_b \varphi$. If G and H are $\nu\Sigma$-goals, then $A \models_b \varphi$ and thus $\exists i \in \mathbb{N} : B_i \models_b \varphi$ follows immediately. Let G be a Y-image finite $\nu\Sigma$-goal. Since SP is complete, there is $\sigma : X \to NF_\Sigma$ with $b \equiv_{SP} \sigma$. Hence $\mathcal{S}(G\sigma_{X\setminus Y})$ is finite. Since G is a $\nu\Sigma$-goal, for all $i \in \mathbb{N}$ and $b : X \to A$, $B_i \models_b G$ iff $A \models_b G$. Hence $B_i \models_b \varphi$ is equivalent to:

$$\forall\ c =_Y b\ :\ A \models_c G\ \Rightarrow\ B_i \models_c H, \tag{3}$$

while the assumption $\cup_{i\in\mathbb{N}} B_i \models_b \varphi$ is equivalent to:

$$\forall\ c =_Y b\ :\ A \models_c G\ \Rightarrow\ \exists\ i \in \mathbb{N} : B_i \models_c H. \tag{4}$$

Since SP is complete, (3) and (4) are equivalent to:

$$\forall\ \tau =_Y \sigma\ :\ A \models G\tau\ \Rightarrow\ B_i \models H\tau, \tag{5}$$

and to:

$$\forall\ \tau =_Y \sigma\ :\ A \models G\tau\ \Rightarrow\ \exists\ i \in \mathbb{N} : B_i \models H\tau, \tag{6}$$

respectively. It remains to conclude from (6) that (5) holds true for some i. We reformulate (6) as follows:

$$\forall\ \tau \in \mathcal{S}(G\sigma_{X\setminus Y})\ \exists\ i \in \mathbb{N} : B_i \models H\sigma_{X\setminus Y}\tau. \tag{7}$$

Since $\mathcal{S}(G\sigma_{X\setminus Y})$ is finite and $\{B_i\}$ is increasing, (7) implies that there is i with $B_i \models H\sigma_{X\setminus Y}\tau$ for all $\tau \in \mathcal{S}(G\sigma_{X\setminus Y})$. But this is equivalent to (5).

[10] This—crucial—proof step follows the proof of [40], Thm. 2.1, which states a corresponding result in modal logic.

(2) Let $\varphi = (\varphi_1 \wedge \cdots \wedge \varphi_n)$ be an image finite dual goal set and $b : X \to A$ such that $\cup_{i\in\mathbb{N}} B_i \models_b \varphi$. Then for all $1 \leq j \leq n$, $\cup_{i\in\mathbb{N}} B_i \models_b \varphi_j$. Hence by (1), for all $1 \leq j \leq n$ there is $m_j \in \mathbb{N}$ such that $B_{m_j} \models_b \varphi_j$. Since $\{B_i\}$ is increasing, we conclude $B_m \models_b \varphi$ for $m = max\{m_j \mid 1 \leq j \leq n\}$. ❑

Theorem 5.15 (image finiteness implies continuity) *A complete and image finite specification SP is continuous.*

Proof. Let $\mathcal{C}$ be the class of $\nu\Sigma$-structures whose $hid\Sigma$-reduct agrees with $A = Her(SP)|_{hid\Sigma}$ and $B_0 \supseteq B_1 \supseteq B_2 \supseteq \cdots \in \mathcal{C}$. The $(\nu AX \setminus hidAX)$-consequence operator Ψ on A is downward continuous iff for all predicates $r \in \nu\Sigma \setminus hid\Sigma$,

$$\cap_{i\in\mathbb{N}} r^{\Psi(B_i)} \subseteq r^{\Psi(\cap_{i\in\mathbb{N}} B_i)},$$

which is equivalent to:

$$\forall i \in \mathbb{N} : a \in r^{\Psi(B_i)} \quad \text{implies} \quad a \in r^{\Psi(\cap_{i\in\mathbb{N}} B_i)}. \tag{1}$$

By the definition of Ψ (cf. Def. 4.3), (1) holds true if for all $(r(t) \Rightarrow \varphi) \in \nu AX$ and $b : X \to A$ such that $a = b^*(t)$,

$$\forall i \in \mathbb{N} : B_i \models_b \varphi \quad \text{implies} \quad \cap_{i\in\mathbb{N}} B_i \models_b \varphi. \tag{2}$$

We show (2). Let $(r(t) \Rightarrow \varphi) \in \nu AX$ and $b : X \to A$ such that $a = b^*(t)$. By Def. 2.4(3) and (4), there are a $hid\Sigma$-goal $G = (r_1(t_1) \wedge \cdots \wedge r_k(t_k))$ and a goal set ψ such that $\varphi = (G \Rightarrow \psi)$. Since for all $1 \leq i \leq k$, $r_i \in hid\Sigma$, we may assume that $hid\Sigma$ includes the complement $\overline{r_i}$ of $r_i : w$ (cf. Def. 3.1) and thus for all $B \in \mathcal{C}$, $\overline{r_i}^B = T_{\Sigma,w} \setminus r_i^B$. Let $\theta = (\overline{r_1}(t_1) \vee \cdots \vee \overline{r_k}(t_k) \vee \psi)$. Since for all $B \in \mathcal{C}$, $B \models \theta$ iff $B \models \psi$, (2) holds true iff

$$\forall i \in \mathbb{N} : B_i \models_b \theta \quad \text{implies} \quad \cap_{i\in\mathbb{N}} B_i \models_b \theta. \tag{3}$$

Since ψ is image finite, θ is also image finite. Hence (3) follows from Lemma 5.13.

Let $\mathcal{C}$ be the class of Σ-structures whose $\nu\Sigma$-reduct agrees with $A = Her(SP)|_{\nu\Sigma}$ and $B_0 \subseteq B_1 \subseteq B_2 \subseteq \cdots \in \mathcal{C}$. The $(AX \setminus \nu AX)$-consequence operator Θ on A is upward continuous iff for all predicates $r \in \Sigma \setminus \nu\Sigma$,

$$a \in r^{\Theta(\cup_{i\in\mathbb{N}} B_i)} \quad \text{implies} \quad \exists i \in \mathbb{N} : a \in r^{\Theta(B_i)}. \tag{4}$$

By the definition of Θ (cf. Def. 4.3), (4) holds true if there are $(r(t) \Leftarrow \varphi) \in AX$ and $b : X \to A$ such that $a = b^*(t)$ and

$$\cup_{i\in\mathbb{N}} B_i \models_b \varphi \quad \text{implies} \quad \exists i \in \mathbb{N} : B_i \models_b \varphi. \tag{5}$$

Since φ is image finite, (5) follows from Lemma 5.14. ❑

Functionality and continuity are the key properties of a swinging specification that allow us to reason about its Herbrand model via consequence operators:

Lemma 5.16 (stepwise constructions of the Herbrand model) *Let $SP = (\Sigma, AX)$ be a functional and continuous specification and $\Phi, \Psi, \Theta, \Gamma$ be the consequence operators of Def. 4.6 and Lemma 5.10, respectively.*

(1) $Her(SP)|_{hid\Sigma} = \cup_{i\in\mathbb{N}} \Phi^i(\emptyset)$ and for all $hid\Sigma$-atoms p,

$$Her(SP) \models p \iff \exists i \in \mathbb{N} : \Phi^i(\emptyset) \models p \iff SP \vdash_{cut} p \iff p \vdash_{SP} \emptyset.$$

(2) $Her(SP)|_{hid\Sigma} = \cup_{i\in\mathbb{N}}\Gamma^i(\emptyset)$ *and for all hidΣ-atoms p,*

$$Her(SP) \models p \iff \exists i \in \mathbb{N} : \Gamma^i(\emptyset) \models p.$$

(3) $Her(SP)|_{\nu\Sigma} = \cap_{i\in\mathbb{N}}\Psi^i(T_\Sigma)$ *and for all ν-atoms p,*

$$Her(SP) \models p \iff \forall i \in \mathbb{N} : \Psi^i(T_\Sigma) \models p.$$

(4) $Her(SP) = \cup_{i\in\mathbb{N}}\Theta^i(\emptyset)$ *and for all Σ-atoms p,*

$$Her(SP) \models p \iff \exists i \in \mathbb{N} : \Theta^i(\emptyset) \models p.$$

Proof. (1) follows from Lemma 4.7(1) and (2) and Thm. 4.10. (2) is Lemma 5.10. (3) and (4) are immediate consequences of Lemma 4.7(3) and (4). ❏

6 Coinductive axioms

By Lemma 3.10, a visible specification SP is behaviorally consistent. If SP has nonempty hidden, μ- or ν-levels, additional conditions are needed to ensure that SP is behaviorally consistent. We first group the symbols and atoms specified above the visible level of SP (cf. Def. 2.4). A symbol is **non-observing** if it is not an observer.

Given a hidden term t, an atom $\delta(t,a,u)$ is **observing** if δ is a transition predicate or $\delta(t,a,u) = (f(t,a) \equiv u)$ and f is a destructor or $\delta(t,a,u) = r(t,a)$ and r is a separator. An atom $\delta(t,u)$ is **non-observing** if δ is a non-observing dynamic predicate or $\delta(t,u) = (f(t) \equiv u)$ and f is a non-observing defined function or $\delta(t,u) = r(t)$ and r is a non-observing static predicate. A goal is **non-observing** if it consists of non-observing atoms.

Given a term tuple t, **visvar(t)** and **hidvar(t)** denote the sets of visible resp. hidden variables of t.

Definition 6.1 (coinductivity) Let $SP = (\Sigma, AX)$ be a swinging specification. A Σ-normal form t is **strongly normal** if for all $\sigma : X \to NF_\Sigma$ and $u \in NF_\Sigma$, $t\sigma \sim_{SP} u$ implies $t\tau = u$ and $\sigma \sim_{SP} \tau$ for some $\tau : X \to NF_\Sigma$. A co-Horn clause $r(t) \Rightarrow \varphi$ is **coinductive** if t is strongly normal.

A Horn clause $p \Leftarrow \varphi$ is **coinductive** if either $p = \delta(t,u)$ is non-observing and t is strongly normal or $p = \delta(t,a,u)$ is observing,

$$\varphi = G \wedge \delta_1(t_1,a_1,u_1) \wedge G_1 \wedge \cdots \wedge \delta_n(t_n,a_n,u_n) \wedge G_n$$

and the following conditions hold true: Let $V_0 = var(t,a,G)$ and for all $1 \leq i \leq n$, $V_i = V_{i-1} \cup var(a_i,u_i,G_i)$.

(1) t is strongly normal or $t = c(t')$ for a constructor c and a strong normal form t', a is strongly normal, G is weakly modal and non-observing, $var(u) \subseteq V_n$ and $out(G) \cap var(t,a) = \emptyset$.

(2) For all $1 \leq i \leq n$, $\delta_i(t_i,a_i,u_i)$ is observing, (t_i,a_i) is normal, u_i is strongly normal, G_i is weakly modal and non-observing, $var(t_i) \subseteq V_{i-1}$, $(var(u_i) \cup out(G_i)) \cap (V_{i-1} \cup var(a_i,u_i)) = \emptyset$ and $hidvar(a_i) \subseteq var(a)$.

SP is **coinductive** if

(3) for all axioms φ of $SP \setminus visSP$, φ is coinductive or an axiom for a non-observing symbol f such that $\sim_{SP}$ is (zigzag) compatible with f,

(4) for all axioms $p \Leftarrow \varphi$ for observers and all non-observing symbols f occurring in φ, the axioms for f are coinductive and do not contain observers.[11] ❑

Note the different rôles the variables of t, a resp. u play in an observing atom $\delta(t,a,u)$: those of t are *consumed*, those of u are *produced*, $var(a)$ may contain both "input" and "output" variables. Intuitively, conditions 6.1(1) and (2) entail a data flow through a conductive axiom $p \Leftarrow \varphi$ that starts out from t and the "input part" of a, proceeds to the "output part" of a, t_i and the "input part" of a_i, $i > 0$, then propagates from u_i and the "output part" of a_i to t_j and the "input part" of a_j, $j > i$, and finally returns to the "output part" of a and u.

At least the observers must have coinductive axioms if the whole specification shall be coinductive. The conditions on observer axioms are less restrictive than those on axioms for non-observing symbols. This may lead one to declare more symbols as observers. However, more observers increase the number of behavior axioms and thus the number of "cases" generated by unfolding a behavioral equivalence $t \sim t'$ or by applying conduction to a clause of the form $\psi \Rightarrow t \sim t'$ (cf. Section 4).

Functional visible specifications are coinductive because then all ground normal forms are strongly normal and thus all axioms are coinductive. Other coinductive specifications cover usual formats of transition system specifications [37, 20], SOS (= structural operational semantics) rules [67], codatatypes [39] as well as Δ/Γ-*complete* equations[12] [33] or *observer complete* function definitions [16]. Δ/Γ-completeness and observer completeness are simple subcases of coinductivity. They deal with purely functional specifications whose behavioral equality is determined by destructors only and whose axioms are mostly unconditional equations.

An observer complete definition in the sense of [16] admits axioms such as $d(c(x)) \equiv u$ where d is a "context" term consisting of *several* destructors. u may also have subterms of the form $e(c(v))$ such that e is a smaller context that d. Our notion of coinductivity restricts d to a single destructor and e to a variable. Apart form the fact that most examples obey the "restriction" there is a simple way of extending an observer complete specification to a coinductive one that is consistent w.r.t. the former: for each destructor f and each axiom $f(d(c(x))) \equiv u$ where d is a non-variable context, introduce a new constructor, say dc, for the composition $d \circ c$, replace $f(d(c(x))) \equiv u$ by $f(dc(x)) \equiv u$, add $d(c(x)) \equiv dc(x)$ to the set of axioms and iterate this procedure until all axioms are coinductive. It terminates because d is a smaller context that $f \circ d$. As an example consider the following observer complete definition of $blink : stream \to stream$ (cf. Ex. 2.8):

$$head(blink) \equiv 0 \quad head(tail(blink)) \equiv 1 \quad tail(tail(blink)) \equiv tail(blink).$$

While $blink$ denotes the stream of alternating zeros and ones, starting with a zero, $tail(blink))$ stands for the stream of alternating zeros and ones, starting with a one. Hence $tail(blink)$ actually denotes a further constructor, $blink' : stream \to stream$:

$$head(blink) \equiv 0 \quad head(blink') \equiv 1 \quad tail(blink') \equiv blink \quad tail(blink) \equiv blink'.$$

Example 6.2 INFSEQ (cf. Ex. 2.8) is coinductive. Even the following specification of stream comprehension analogously to list comprehension (cf. Ex. 2.1) is coinductive:

$$head(filter(g,s)) \equiv x \quad \Leftarrow \quad head(s) \equiv x \wedge g(x) \equiv true \tag{1}$$

$$head(filter(g,s)) \equiv head(filter(g,tail(s))) \quad \Leftarrow \quad head(s) \equiv x \wedge g(x) \equiv false. \tag{2}$$

[11] This excludes mutually-recursive axiomatizations of observers and non-observers.

[12] Δ and Γ are sets of destructors and constructors, respectively. Δ/Γ-completeness is a special case of the congruence criterion of [69], Thm. 16.

However, only coinductivity *and* functionality imply behavioral consistency (see Thm. 6.5 below), but (1) and (2) are not complete and thus not functional for streams s none of whose elements satisfies g. Correct axioms for $filter$ can only be part of a specification of finite *and* infinite streams such as STREAM (Ex. 6.6). ❏

One may flatten (2) such that the right-hand side of the conclusion consists of non-observing symbols and the resulting axiom is still coinductive and equivalent to the original one:

$$head(filter(g,s)) \equiv y \quad \Leftarrow \quad head(s) \equiv x \wedge g(x) \equiv false \wedge tail(s) \equiv s' \wedge head(filter(g,s')) \equiv y. \qquad (3)$$

Each coinductive axiom can be transformed analogously:

Lemma 6.3 *Given a coinductive specification SP, for each axiom $\psi = (\delta(t,a,u) \Leftarrow \varphi)$ with observing conclusion there is a coinductive axiom $\psi' = (\delta(t,a,u') \Leftarrow \varphi')$ such that u' is normal and SP and $(SP \setminus \{\psi\}) \cup \{\psi'\}$ are inductively equivalent.*

Proof. We show the conjecture by induction on the number k of occurrences of defined functions in u. Since ψ is coinductive,

$$\varphi \quad = \quad G \wedge \delta_1(t_1,a_1,u_1) \wedge G_1 \wedge \cdots \wedge \delta_n(t_n,a_n,u_n) \wedge G_n$$

such that Def. 6.1(1) and (2) hold true. If $k = 0$, the proof is complete with $\psi' = \psi$. Let $k > 0$. Then there is a minimal subterm $f(t)$ of u such that f is a defined function and t is normal. Let $x \in X \setminus var(\psi)$. If f is a destructor, then 6.1(1) and (2) hold true for $n+1$ instead of n, $\delta_1(t_1,a_1,u_1) = (f(t) \equiv x)$, and $G_{n+1} = \emptyset$. If f is non-observing, then 6.1(1) and (2) hold true for $G_n \wedge f(t) \equiv x$ instead of G_n. Hence in the both cases,

$$\psi'' \quad = \quad \delta(t,a,u[x/f(t)]) \;\Leftarrow\; \varphi \wedge f(t) \equiv x$$

and thus $SP' = (SP \setminus \{\psi\}) \cup \{\psi''\}$ are coinductive. Obviously, SP and SP' are inductively equivalent. By induction hypothesis, there is a coinductive axiom $\psi' = (\delta(t,a,u') \Leftarrow \varphi')$ such that u' is normal and SP' and $(SP' \setminus \{\psi''\}) \cup \{\psi'\}$ are inductively equivalent. Since $SP' \setminus \{\psi''\} = SP \setminus \{\psi\}$, SP and $(SP \setminus \{\psi\}) \cup \{\psi'\}$ are inductively equivalent. ❏

Coinductive *definition schemas* should not be confused with coinductive *proof rules* such as fixpoint or hidden coinduction (see Section 4). To emphasize the difference some authors call the former *corecursion schemas* (cf., e.g., [11]). Strong normal forms give rise to rules of term splitting and clash "modulo behavioral equivalence" (see Section 4):

behavioral term splitting $\quad \dfrac{c(t_1,\dots,t_n) \sim c(u_1,\dots,u_n)}{t_1 \sim u_1 \wedge \cdots \wedge t_n \sim u_n} \Updownarrow \quad$ if $c(x)$ is strongly normal

behavioral clash $\quad \dfrac{c(t) \sim d(u)}{FALSE} \Updownarrow \quad$ if $c(x)$ and $d(y)$ are different strong normal forms

Definition and Lemma 6.4 *Given a swinging specification $SP = (\Sigma, AX)$, the* **constructor closure** $\approx$ **of** $\sim_{SP}$ *is the binary relation on T_Σ that is defined inductively as follows:*

- $\sim_{SP} \subseteq \approx$,
- *for all constructors $c : w \to s$ and $t,t' \in T_{\Sigma,w}$, $t \approx t'$ implies $c(t) \approx c(t')$.*

Let t be a strong normal form, $\sigma : X \to NF_\Sigma$ and $u \in NF_\Sigma$ such that $t\sigma \approx u$. Then $t\tau = u$ and $\sigma \approx \tau$ for some $\tau : X \to NF_\Sigma$.

Proof by induction on the size of t. Let $t\sigma \approx u$. If $t\sigma \sim_{SP} u$, then $t\tau = u$ and $\sigma \approx \tau$ for some $\tau : X \to NF_\Sigma$ because t is strongly normal. Otherwise $t\sigma = c(v)$, $v \approx u'$ and $c(u') = u$ for some constructor c and $v, u' \in NF_\Sigma$. If t is a variable, then define $\tau : X \to NF_\Sigma$ by $t\tau = u$ and $\tau =_{X\setminus\{t\}} \sigma$. Otherwise $t = c(t')$ and $v = t'\sigma$ for some $t' \in NF_\Sigma(X)$. Hence $t'\sigma \approx u'$ and thus by induction hypothesis, $t'\tau = u'$ and $\sigma \approx \tau$ for some $\tau : X \to NF_\Sigma$. Hence in both cases, $t\tau = u$ and $\sigma \approx \tau$. ❑

Theorem 6.5 (criteria for behavioral consistency) *A coinductive, functional and continuous specification SP is behaviorally consistent.*

Proof. Let $SP = (\Sigma, AX)$. By Lemma 3.6, $\sim_{SP}$ is compatible with all non-equality symbols of $visSP$ (cf. 2.4) and all behavioral equalities and zigzag compatible with all equality predicates. Def. 6.1(3) and (4) imply that the hidden level of SP splits into three successive sublevels:

- The **1st hidden level** consists of all non-observing symbols of the hidden level of SP and their axioms such that these do not contain observers.
- The **2nd hidden level** consists of all observers of SP and their axioms.
- The **3rd hidden level** consists of all remaining symbols of the hidden level of SP and their axioms.

Let $\approx$ be the constructor closure of $\sim_{SP}$. $\approx$ is compatible with the constructors of Σ. Since $\equiv_{SP}$ satisfies the behavior axioms of SP, $\equiv_{SP}$ is a subset of $\sim_{SP}$ and thus of $\approx$.

At first, we show that $\approx$ satisfies the behavior axioms for visible sorts. Let s be a visible sort and $t \approx_s t'$. We prove $Her(SP) \models t \equiv t'$ by induction on the size of t, t'. If $t \sim_s t'$, then $t \equiv_{SP} t'$. Otherwise $t = c(u)$, $u \approx u'$ and $t' = c(u')$ for a constructor c and term tuples u, u'. By induction hypothesis, $u \equiv_{SP} u'$. Hence $t \equiv_{SP} t'$.

Since $\sim_{SP}$ is the greatest relation satisfying the behavior axioms, we conclude that the restriction of $\approx$ to visible sorts is a subrelation of $\sim_{SP}$ and thus equal to the corresponding restrictions of $\sim_{SP}$ and $\equiv_{SP}$. Hence $\approx$ is compatible with all non-equality symbols of $visSP$.

Let $SP_1 = (\Sigma_1, AX_1)$ be $visSP$ together with the 1st hidden level of SP. Suppose that

$$\approx \text{ is (zigzag) compatible with all symbols specified on the 1st hidden level.} \tag{1}$$

Since $\approx$ is compatible with all non-equality symbols of $visSP$, (1) implies that $\approx$ is (zigzag) compatible with Σ_1. Next we show (1).

Let $rSP_1 = (r\Sigma_1, rAX_1)$ be the relational version of SP_1 (cf. Def. 4.12). Since SP is coinductive, rSP_1 is also coinductive. Since SP is functional, Cor. 4.15 implies that (1) is equivalent to (2): for all non-observing ground $r\Sigma_1$-atoms $\delta(t, u)$ there is $u' \in NF_\Sigma \cup \{\varepsilon\}$ such that

$$Her(rSP_1) \models \delta(t, u) \ \wedge \ t \approx t' \quad \text{implies} \quad Her(rSP_1) \models \delta(t', u') \ \wedge \ u \approx u'. \tag{2}$$

By Lemma 5.16(1), (2) follows from a corresponding property of an approximation of $Her(rSP_1)$: for all non-observing ground $r\Sigma_1$-atoms $\delta(t, u)$ specified on the 1st hidden level and $i \in \mathbb{N}$ there is $u' \in NF_\Sigma \cup \{\varepsilon\}$ such that

$$\Phi^i(\emptyset) \models \delta(t, u) \ \wedge \ t \approx t' \quad \text{implies} \quad \Phi^i(\emptyset) \models \delta(t', u') \ \wedge \ u \approx u' \tag{3}$$

where Φ is the $(rAX_1 \setminus visAX)$-consequence operator on $Her(rSP_1)|_{vis\Sigma}$ and $visSP = (vis\Sigma, visAX)$.

We prove (3) by induction on i. Since for all μ-predicates $r \in r\Sigma_1$, $r^\emptyset = \emptyset$, (3) holds true for $i = 0$. Let $i > 0$. By induction hypothesis, (3) is valid for $i - 1$ and thus

$$\approx \text{ is a behavioral } \Sigma_1\text{-congruence on } \Phi^{i-1}(\emptyset). \tag{4}$$

Let $\Phi^i(\emptyset) \models \delta(t,u)$ and $t \approx t'$. By the definition of Φ and since rSP_1 is coinductive, there are an axiom $\delta(t_0,u_0) \Leftarrow \varphi$ on the 1st hidden level and $\sigma : X \to NF_\Sigma$ such that t_0 is strongly normal, $(t_0,u_0)\sigma = (t,u)$ and $\Phi^{i-1}(\emptyset) \models \varphi\sigma$. Since t_0 is strongly normal and $t_0\sigma = t \approx t'$, Lemma 6.4 implies $t_0\tau = t'$ and $\sigma \approx \tau$ for some $\tau : X \to NF_\Sigma$. By Def. 2.4(b), φ is weakly modal with output Y such that $var(t_0) \cap Y = \emptyset$. Since $\sigma \approx \tau$, (4) and Thm. 3.8(2) imply $\Phi^{i-1}(\emptyset) \models \varphi\tau'$ for some $\tau' \approx \tau$ with $\tau' =_Y \tau$. Hence $\Phi^i(\emptyset) \models \delta(t_0,u_0)\tau'$ and $u = u_0\sigma \approx u_0\tau \approx u_0\tau'$. Since $var(t_0)\cap Y = \emptyset$, $t_0\tau' = t_0\tau = t'$. Hence $\Phi^i(\emptyset) \models \delta(t',u')$ for $u' = u_0\tau' \approx u$.

This completes the proof of (1). Next we show that $\sim_{SP}$ is compatible with all constructors of Σ.

Suppose that $\approx$ satisfies all behavior axioms for Σ (cf. Def. 2.4(3)). Then $\approx$ agrees with $\sim_{SP}$ because behavioral SP-equivalence is the greatest relation satisfying the behavior axioms and because $\sim_{SP}$ is included in $\approx$. Consequently, $\sim_{SP}$ is (zigzag) compatible with all constructors of Σ.

Since we have already shown above that $\approx$ satisfies the behavior axioms for visible sorts, it remains to show that $\approx$ satisfies the behavior axioms for the hidden sorts of Σ. This can be reduced to the following condition (5) because $\approx \circ \equiv_{SP} \circ \approx$ is a subset of $\approx$: for all observing ground atoms $\delta(t,a,u)$ there is $u' \in T_\Sigma \cup \{\varepsilon\}$ such that

$$Her(SP) \models \delta(t,a,u) \ \wedge \ t \approx t' \quad \text{implies} \quad Her(SP) \models \delta(t',a,u') \ \wedge \ u \approx u'. \tag{5}$$

Since SP is functional and δ is compatible with SP-equivalence, we may assume that t, t', a, u, u' are normal forms. Hence by Lemma 5.16(1), (5) is equivalent to (6): for all ground normal forms t, a, u and observing atoms $\delta(t,a,u)$ there is $u' \in NF_\Sigma \cup \{\varepsilon\}$ such that

$$\delta(t,a,u) \vdash_{SP} \emptyset \ \wedge \ t \approx t' \quad \text{implies} \quad \delta(t',a,u') \vdash_{SP} \emptyset \ \wedge \ u \approx u'. \tag{6}$$

Hence it remains to show (6).

By (1), Thm. 3.8(2) and since SP is functional, for all weakly modal Σ_1-goals G and $\sigma, \tau : X \to NF_\Sigma$,

$$\sigma \approx \tau \ \wedge \ G\sigma \vdash_{SP} \emptyset \quad \text{implies} \quad G\tau' \vdash_{SP} \emptyset \ \text{ for some } \tau' : X \to NF_\Sigma \text{ with } \sigma \approx \tau' =_{out(G)} \tau. \tag{7}$$

Let $\delta(t,a,u) \vdash_{SP} \emptyset$ and $t \approx t'$. We show the conclusion of (6) by induction on the length of a shortest reduction R of $\delta(t,a,u)$ into the empty goal. Since SP is coinductive, there are a goal

$$\varphi \;\; = \;\; G_0 \wedge \delta_1(t_1,a_1,u_1) \wedge G_1 \wedge \cdots \wedge \delta_n(t_n,a_n,u_n) \wedge G_n$$

and an axiom $\delta(t_0,a_0,u_0) \Leftarrow \varphi$ on the 2nd hidden level such that Def. 6.1(1) and (2) hold true for t_0, a_0, u_0, G_0 instead of t, a, u, G. Moreover, there is $\sigma : X \to NF_\Sigma$ such that $(t_0,a_0,u_0)\sigma = (t,a,u)$, $G_0\sigma \vdash_{SP} \emptyset$ and for all $1 \le i \le n$ there is an SP-reduction of $\delta_i(t_i,a_i,u_i)\sigma$ into $\emptyset$ that is shorter than R. By the definition of $\approx$, we have one of two cases:

(A) $t \sim_{SP} t'$,

(B) $t = d(v)$, $v \approx v'$ and $d(v') = t'$ for some constructor d and ground terms v and v'.

Case A. $\delta(t,a,u) \vdash_{SP} \emptyset$ implies $Her(SP) \models \delta(t,a,u)$. Suppose that $\delta(t,a,u) = (f(t,a) \equiv u)$ for some destructor $f : w \to s$. Hence $f(t,a) \sim_{SP} f(t',a)$ because $\sim_{SP}$ satisfies the behavior axioms. We conclude $Her(SP) \models (f(t',a) \equiv u') = \delta(t',a,u')$ for $u' = f(t',a) \sim_{SP} f(t,a) = u$. If δ is a separator, then $Her(SP) \models \delta(t',a,u)$ because $\sim_{SP}$ satisfies the behavior axioms. Hence $Her(SP) \models \delta(t',a,u')$ for $u' = u$. If δ is a transition predicate, then $Her(SP) \models \delta(t',a,u')$ for some $u' \sim_{SP} u$ because $\sim_{SP}$ satisfies the behavior axioms.

Hence in all three subcases, $Her(SP) \models \delta(t', a, u')$ for some $u' \sim_{SP} u$. Since $\sim_{SP}$ is a subset of $\approx$, we conclude $u' \approx u$.

Case B. By Def. 6.1(1), there are two subcases, namely B1: t_0 is strongly normal, or B2: $t_0 = c(t'_0)$ for a constructor c and a strong normal form t'_0. In Case B1, $(t_0, a_0)\sigma = (t, a) \approx (t', a)$ and Lemma 6.4 implies $(t_0, a_0)\tau = (t', a)$ and $\sigma \approx \tau$ for some $\tau : X \to NF_\Sigma$. In Case B2, $c(t'_0\sigma) = t_0\sigma = t = d(v)$ and thus $c = d$ and $t'_0\sigma = v$. Hence $(t'_0, a_0)\sigma = (v, a) \approx (v', a)$ and Lemma 6.4 implies $(t'_0, a_0)\tau = (v', a)$ and $\sigma \approx \tau$ for some $\tau : X \to NF_\Sigma$.

We construct a substitution $\tau' : X \to NF_\Sigma$ with

(a) $t_0\tau = t_0\tau'$

and prove by induction on i that for all $0 \le i \le n$,

(b) $a_i\sigma = a_i\tau'$,

(c) $\delta_i(t_i, a_i, u_i)\tau' \vdash_{SP} \emptyset$ if $i > 0$,

(d) $G_i\tau' \vdash_{SP} \emptyset$,

(e) $x\sigma \approx x\tau'$ for all $x \in V_i$.

Define $x\tau' = x\tau$ for all $x \in var(t_0, a_0)$. Then (a) holds true. Since $a_0\tau = a = a_0\sigma$, (b) holds true for $i = 0$. By (7), $\sigma \approx \tau$ and $G_0\sigma \vdash_{SP} \emptyset$ imply $G_0\tau'' \vdash_{SP} \emptyset$ for some $\tau'' : X \to NF_\Sigma$ with $\sigma \approx \tau'' =_{out(G_0)} \tau$. Define $x\tau' = x\tau''$ for all $x \in var(G_0) \setminus var(t_0, a_0)$. Since $out(G_0) \cap var(t_0, a_0) = \emptyset$, we have $x\tau'' = x\tau = x\tau'$ for all $x \in var(G_0) \cap var(t_0, a_0)$. Hence $G_0\tau'' \vdash_{SP} \emptyset$ implies (d) for $i = 0$. Moreover, for all $x \in var(G_0) \setminus var(t_0, a_0)$, $x\sigma \approx x\tau'' = x\tau'$. Hence $V_0 = var(t_0, a_0, G_0)$, (a) and (b) for $i = 0$ imply (e) for $i = 0$.

Let $i > 0$. Suppose that (b)-(e) hold true for $i - 1$. Since $hidvar(a_i) \subseteq var(a_0)$ and $a_0\sigma = a_0\tau'$, we have $x\sigma = x\tau'$ for all $x \in hidvar(a_i)$. Define $x\tau' = x\sigma$ for all $x \in visvar(a_i) \setminus V_{i-1}$. Hence (e) for $i - 1$ implies $x\sigma \approx x\tau'$ and thus $x\sigma \equiv_{SP} x\tau'$ for all $x \in visvar(a_i)$. We conclude $x\sigma = x\tau'$ for all $x \in visvar(a_i)$ because $x\sigma$ and $x\tau'$ are normal and SP is consistent. Hence for all $x \in var(a_i)$, $x\sigma = x\tau'$, and thus (b) holds true.

Since $var(t_i) \subseteq V_{i-1}$ and t_i is normal, (e) for $i - 1$ implies $t_i\sigma \approx t_i\tau'$. Since $\delta_i(t_i, a_i, u_i)\sigma$ has a reduction into $\emptyset$ that is shorter than R, the induction hypothesis (6) implies $\delta_i(t_i\tau', a_i\sigma, u') \vdash_{SP} \emptyset$ and $u_i\sigma \approx u'$ for some $u' \in NF_\Sigma$. Since u_i is strongly normal, u' is normal and $u_i\sigma \approx u'$, Lemma 6.4 implies $u_i\vartheta = u'$ and $\sigma \approx \vartheta$ for some $\vartheta : X \to NF_\Sigma$. Define $x\tau' = x\vartheta$ for all $x \in var(u_i) \setminus (V_{i-1} \cup var(a_i))$. Since $(V_{i-1} \cup var(a_i)) \cap var(u_i) = \emptyset$, $u_i\theta = u'$ implies $u_i\tau' = u'$. Hence by (b), $\delta_i(t_i\tau', a_i\sigma, u') \vdash_{SP} \emptyset$ implies (c).

Define $\eta : X \to NF_\Sigma$ by $x\eta = x\tau'$ for all $x \in V_{i-1} \cup var(a_i, u_i)$ and $x\eta = x\sigma$ otherwise. By (e) for $i - 1$, $x\sigma \approx x\tau' = x\eta$ for all $x \in V_{i-1}$. By (b), $x\sigma = x\tau' = x\eta$ for all $x \in var(a_i)$. Since $x\sigma \approx x\vartheta = x\tau' = x\eta$ for all $x \in var(u_i) \setminus (V_{i-1} \cup var(a_i))$, we conclude $\sigma \approx \eta$. Hence by (7), $G_i\sigma \vdash_{SP} \emptyset$ implies $G_i\tau'' \vdash_{SP} \emptyset$ for some $\tau'' : X \to NF_\Sigma$ with $\sigma \approx \tau'' =_{out(G_i)} \eta$. Define $x\tau' = x\tau''$ for all $x \in var(G_i) \setminus (V_{i-1} \cup var(a_i, u_i))$. Since $out(G_i) \cap (V_{i-1} \cup var(a_i, u_i)) = \emptyset$, we have $x\tau'' = x\eta = x\tau'$ for all $x \in var(G_i) \cap (V_{i-1} \cup var(a_i, u_i))$. Hence $G_i\tau'' \vdash_{SP} \emptyset$ implies (d). Moreover, for all $x \in var(G_i) \setminus (V_{i-1} \cup var(a_i, u_i))$, $x\sigma \approx x\tau'' = x\tau'$, and for all $x \in var(u_i) \setminus (V_{i-1} \cup var(a_i))$, $x\sigma \approx x\vartheta = x\tau'$. Hence $V_i = V_{i-1} \cup var(a_i, u_i, G_i)$, (e) for $i - 1$ and (b) imply (e).

(c) for all $0 \le i \le n$ and (d) for all $1 \le i \le n$ imply $\varphi\tau' \vdash_{SP} \emptyset$. Hence $\delta(t_0, a_0, u_0)\tau' \vdash_{SP} \emptyset$. In Case B1 (see above), (a) and (b) for $i = 0$ imply $(t_0, a_0)\tau' = (t_0\tau, a_0\sigma) = (t', a)$. In Case B2 (see above), (a) and (b) for $i = 0$ imply $(t_0, a_0)\tau' = (t_0\tau, a_0\sigma) = (c(t'_0\tau), a_0\sigma) = (c(v'), a) = (d(v'), a) = (t', a)$.

Since $var(u_0) \subseteq V_n$, (e) for $i = n$ implies $x\sigma \approx x\tau'$ for all $x \in var(u_0)$. By Lemma 6.3 and Prop. 5.6(3), we may assume that u_0 is normal. Hence by (1), $u_0\sigma \approx u_0\tau'$. Therefore, the conclusion of (6) holds true for $u' = u_0\tau'$.

This finishes Case B of the proof of (6) from which we have already concluded that $\sim_{SP}$ is compatible with the constructors of Σ. Since $\sim_{SP}$ is (zigzag) compatible with Σ_1 and all behavioral equalities and since $Her(SP)$ satisfies the behavior axioms for Σ, it remains to show the following properties:

(8) For all destructors $f : sw \to s'$, $t \in T_{\Sigma,s}$ and $a \sim_{SP} a' \in T_{\Sigma,w}$, $f(t,a) \sim_{SP} f(t,a')$.

(9) For all separators $r : sw$ and $t \in T_{\Sigma,s}$, $Her(SP) \models r(t,a) \ \wedge \ a \sim_{SP} a'$ implies $Her(SP) \models r(t,a')$.

(10) For all transition predicates $\delta : sws'$ and $t \in T_{\Sigma,s}$,
$Her(SP) \models \delta(t,a,u) \ \wedge \ a \sim_{SP} a'$ implies $Her(SP) \models \delta(t,a',u') \ \wedge \ u \sim_{SP} u'$ for some u'.

(11) $\sim_{SP}$ is (zigzag) compatible with all symbols specified on the 3rd hidden or a higher level of SP.

Let $rSP = (r\Sigma, rAX)$ be the relational version of SP, $SP_2 = (\Sigma_2, AX_2)$ be the subspecification of rSP consisting of $vis(rSP)$ and the 1st and 2nd hidden level of SP, $hid(rSP) = (\Sigma_3, AX_3)$ and $\nu(rSP) = (\Sigma_4, AX_4)$ (cf. Defs. 2.4 and 4.12). Since $Her(SP)$ satisfies the behavior axioms for Σ, (8)-(10) imply that $\sim_{SP}$ is (zigzag) compatible with Σ_2.

Since SP is coinductive, rSP is also coinductive. Since SP is functional, Cor. 4.15 implies that (8)-(11) can be combined to the following two implications: for all observing ground $r\Sigma$-atoms $\delta(t,a,u)$ there is $u' \in NF_\Sigma \cup \{\varepsilon\}$ such that

$$Her(rSP) \models \delta(t,a,u) \ \wedge \ a \sim_{SP} a' \quad \text{implies} \quad Her(rSP) \models \delta(t,a',u') \ \wedge \ u \sim_{SP} u', \tag{12}$$

and for all non-observing ground $r\Sigma$-atoms $\delta(t,u)$ there is $u' \in NF_\Sigma \cup \{\varepsilon\}$ such that

$$Her(rSP) \models \delta(t,u) \ \wedge \ t \sim_{SP} t' \quad \text{implies} \quad Her(rSP) \models \delta(t',u') \ \wedge \ u \sim_{SP} u'. \tag{13}$$

By Lemma 5.16(1), (3) and (4), (12) and (13) follow from corresponding properties of approximations of $Her(rSP)$: for all observing ground $r\Sigma$-atoms $\delta(t,a,u)$ and $i \in \mathbb{N}$ there is $u' \in NF_\Sigma \cup \{\varepsilon\}$ such that

$$\Delta^i(\emptyset) \models \delta(t,a,u) \ \wedge \ a \sim_{SP} a' \quad \text{implies} \quad \Delta^i(\emptyset) \models \delta(t,a',u') \ \wedge \ u \sim_{SP} u' \tag{14}$$

where Δ is the AX_2-consequence operator on NF_Σ, for all non-observing ground $r\Sigma$-atoms $\delta(t,u)$ and $i \in \mathbb{N}$ there is $u' \in NF_\Sigma \cup \{\varepsilon\}$ such that

$$\Phi^i(\emptyset) \models \delta(t,u) \ \wedge \ t \sim_{SP} t' \quad \text{implies} \quad \exists\, j \in \mathbb{N} : \Phi^j(\emptyset) \models \delta(t',u') \ \wedge \ u \sim_{SP} u' \tag{15}$$

where Φ is the $(AX_3 \setminus AX_2)$-consequence operator on $Her(rSP)|_{\Sigma_2}$, for all ν-predicates $r : w \in \Sigma$, $t, t' \in NF_{\Sigma,w}$ and $i \in \mathbb{N}$,

$$\Psi^i(NF_\Sigma) \models r(t) \ \wedge \ t \sim_{SP} t' \quad \text{implies} \quad \Psi^i(NF_\Sigma) \models r(t') \tag{16}$$

where Ψ is the $(AX_4 \setminus AX_3)$-consequence operator on $Her(rSP)|_{\Sigma_3}$, and for all non-observing ground $r\Sigma$-atoms $\delta(t,u)$ with $\delta \in \Sigma \setminus \Sigma_4$, $t, t' \in NF_{\Sigma,w}$ and $i \in \mathbb{N}$ there is $u' \in NF_\Sigma \cup \{\varepsilon\}$ such that

$$\Theta^i(\emptyset) \models \delta(t,u) \ \wedge \ t \sim_{SP} t' \quad \text{implies} \quad \exists\, j \in \mathbb{N} : \Theta^j(\emptyset) \models \delta(t',u') \ \wedge \ u \sim_{SP} u' \tag{17}$$

where Θ is the $(rAX \setminus AX_4)$-consequence operator on $Her(rSP)|_{\Sigma_4}$.

We prove (14)-(17) by induction on i. Since for all μ-predicates $r \in r\Sigma$, $r^\emptyset = \emptyset$ and for all ν-predicates $r : w \in r\Sigma$, $r^{NF_\Sigma} = NF_{\Sigma,w}$, (14)-(17) hold true for $i = 0$. Let $i > 0$.

Proof of (14). Let $\Delta^i(\emptyset) \models \delta(t,a,u)$ and $a \sim_{SP} a'$. By the definition of Δ and since rSP is coinductive, there are an axiom $\delta(t_0,a_0,u_0) \Leftarrow \varphi$ on the 2nd hidden level as in Def. 6.1 and $\sigma : X \to NF_\Sigma$ such that $(t_0,a_0,u_0)\sigma = (t,a,u)$ and $\Delta^{i-1}(\emptyset) \models \varphi\sigma$. Since a_0 is strongly normal, $a_0\sigma = a \sim_{SP} a'$ implies $a_0\tau = a'$ and $\sigma \sim_{SP} \tau$ for some $\tau : X \to NF_\Sigma$. By Def. 2.4(b), φ is weakly modal with output Y such that $var(t_0,a_0) \cap Y = \emptyset$. Since $\sigma \sim_{SP} \tau$, the induction hypothesis (14) for $i-1$ and Thm. 3.8(2) imply $\Delta^{i-1}(\emptyset) \models \varphi\tau'$ for some $\tau' \sim_{SP} \tau$ with $\tau' =_Y \tau$. Hence $\Delta^i(\emptyset) \models \delta(t_0,a_0,u_0)\tau'$ and $u = u_0\sigma \sim_{SP} u_0\tau \sim_{SP} u_0\tau'$. Since $var(t_0,a_0) \cap Y = \emptyset$, $(t_0,a_0)\tau' = (t_0,a_0)\tau = (t_0\tau,a')$. Hence $\Delta^i(\emptyset) \models \delta(t_0\tau,a',u')$ for $u' = u_0\tau' \sim_{SP} u$. Since $\sim_{SP}=\approx$ satisfies the behavior axioms of Σ (see above), $t_0\tau \sim_{SP} t_0\sigma = t$ implies $\Delta^i(\emptyset) \models \delta(t,a',u'')$ for some $u'' \sim_{SP} u' \sim_{SP} u$.

Proof of (15). Let $\Phi^i(\emptyset) \models \delta(t,u)$ and $t \sim_{SP} t'$. By the definition of Φ and since rSP is coinductive, there are an axiom $\delta(t_0,u_0) \Leftarrow \varphi$ on the 3rd hidden level and $\sigma : X \to NF_\Sigma$ such that $(t_0,u_0)\sigma = (t,u)$, $\Phi^{i-1}(\emptyset) \models \varphi\sigma$ and by 6.1(3), t_0 is strongly normal or $\sim_{SP}$ is zigzag compatible with δ. In the second case, there is $u' \sim_{SP} u$ such that $Her(SP) \models \delta(t',u')$ and thus $\Phi^i(\emptyset) \models \delta(t',u')$ because $\delta(t',u')$ is a Σ_3-atom. In the first case (t_0 is strongly normal), $t_0\sigma = t \sim_{SP} t'$ implies $t_0\tau = t'$ and $\sigma \sim_{SP} \tau$ for some $\tau : X \to NF_\Sigma$. By Def. 2.4(b), φ is weakly modal with output Y such that $var(t_0) \cap Y = \emptyset$. Since $\sigma \sim_{SP} \tau$ and Φ is monotone, the induction hypothesis (15) for $i-1$ and Thm. 3.8(2) imply $\Phi^j(\emptyset) \models \varphi\tau'$ for some $j \in \mathbb{N}$ and $\tau' \sim_{SP} \tau$ with $\tau' =_Y \tau$. Hence $\Phi^{j+1}(\emptyset) \models \delta(t_0,u_0)\tau'$ and $u = u_0\sigma \sim_{SP} u_0\tau \sim_{SP} u_0\tau'$. Since $var(t_0) \cap Y = \emptyset$, $t_0\tau' = t_0\tau = t'$. Hence $\Phi^{j+1}(\emptyset) \models \delta(t',u')$ for $u' = u_0\tau' \sim_{SP} u$.

Proof of (16). Let $\Psi^i(NF_\Sigma) \models r(t)$ and $t \sim_{SP} t'$. By the definition of Ψ,

$$\text{for all } (r(t_0) \Rightarrow \varphi) \in AX \text{ and } \sigma : X \to T_\Sigma,\ t = t_0\sigma \text{ implies } \Psi^{i-1}(NF_\Sigma) \models \varphi\sigma. \tag{18}$$

Suppose that

$$\text{for all } (r(t_0) \Rightarrow \varphi) \in AX \text{ and } \tau : X \to T_\Sigma,\ t' = t_0\tau \text{ implies } \Psi^{i-1}(NF_\Sigma) \models \varphi\tau. \tag{19}$$

By the definition of Ψ, $\Psi^i(NF_\Sigma) \models r(t')$ and thus the proof is complete. It remains to show (19). Let $r(t_0) \Rightarrow \varphi$ be a co-Horn clause of AX and $\tau : X \to T_\Sigma$ such that $t' = t_0\tau$. By 6.1(3), t_0 is strongly normal or $\sim_{SP}$ is compatible with r. In the second case, $Her(SP) \models r(t')$ and thus $\Psi^i(NF_\Sigma) \models r(t')$ because $r(t')$ is a Σ_4-atom. By the definition of Ψ, $\Psi^i(NF_\Sigma) \models r(t_0\tau)$ and $(r(t_0) \Rightarrow \varphi) \in AX$ imply $\Psi^{i-1}(NF_\Sigma) \models \varphi\tau$. In the first case ($t_0$ is strongly normal), there is $\sigma : X \to NF_\Sigma$ such that $t_0\sigma = t$ and $\tau \sim_{SP} \sigma$. By (18), $\Psi^{i-1}(NF_\Sigma) \models \varphi\sigma$. By Def. 2.4(4), φ is poly-modal. Since $t \sim_{SP} t'$, the induction hypothesis (16) for $i-1$ and Thm. 3.8(3) imply $\Psi^{i-1}(NF_\Sigma) \models \varphi\tau$.

Proof of (17). Let $\Theta^i(\emptyset) \models \delta(t,u)$ and $t \sim_{SP} t'$. By the definition of Θ and since rSP is coinductive, there are an axiom $\delta(t_0,u_0) \Leftarrow \varphi$ on the μ-level of AX and $\sigma : X \to NF_\Sigma$ such that $(t_0,u_0)\sigma = (t,u)$, $\Theta^{i-1}(\emptyset) \models \varphi\sigma$ and by 6.1(3), t_0 is strongly normal or $\sim_{SP}$ is zigzag compatible with δ. In the second case, there is $u' \sim_{SP} u$ such that $Her(SP) \models \delta(t',u')$ and thus $\Theta^i(\emptyset) \models \delta(t',u')$. In the first case ($t_0$ is strongly normal), $t_0\sigma = t \sim_{SP} t'$ implies $t_0\tau = t'$ and $\sigma \sim_{SP} \tau$ for some $\tau : X \to NF_\Sigma$. By Def. 2.4(b), φ is weakly modal with output Y such that $var(t_0) \cap Y = \emptyset$. Since $\sigma \sim_{SP} \tau$ and Θ is monotone, the induction hypothesis (17) for $i-1$ and Thm. 3.8(2) imply $\Theta^j(\emptyset) \models \varphi\tau'$ for some $j \in \mathbb{N}$ and $\tau' \sim_{SP} \tau$ with $\tau' =_Y \tau$. Hence $\Theta^{j+1}(\emptyset) \models \delta(t_0,u_0)\tau'$ and $u = u_0\sigma \sim_{SP} u_0\tau \sim_{SP} u_0\tau'$. Since $var(t_0) \cap Y = \emptyset$, $t_0\tau' = t_0\tau = t'$. Hence $\Theta^{j+1}(\emptyset) \models \delta(t',u')$ for $u' = u_0\tau' \sim_{SP} u$. ❑

Example 6.6 In the following stream specification, the destructors *head* and *tail* of INFSEQ (cf. Ex. 2.8) are replaced by a transition predicate $\longrightarrow$: $stream \times entry \times stream$. This allows us to include finite sequences into the stream domain.

STREAM = LISTORD `and` NAT `then`

`hidsorts`	$stream = stream(entry)$
`constructs`	$empty :\to stream$
	$\_\&\_ : entry \times stream \to stream$
	$blink :\to stream(nat)$
	$nats : nat \to stream(nat)$
	$odds, evens : stream \to stream$
	$zip : stream \times stream \to stream$
	$map : (entry \to entry) \times stream \to stream$
	$filter : (entry \to bool) \times stream \to stream$
`separators`	$disabled : stream$
`transpreds`	$\_ \xrightarrow{-} \_ : stream \times entry \times stream$
`static` μ`-preds`	$enabled, finite : stream$
	$exists : (entry \to bool) \times stream$
ν`-preds`	$fair : (entry \to bool) \times stream$
	$infinite : stream$
	$forall : (entry \to bool) \times stream$
`vars`	$n : nat \quad x, y : entry \quad L : list \quad s, s', t, t' : stream$
	$f : entry \to entry \quad g : entry \to bool$
`Horn axioms`	$x\&s \xrightarrow{x} s$
	$blink \xrightarrow{0} 1\&blink$
	$nats(n) \xrightarrow{n} nats(n+1)$
	$odds(s) \xrightarrow{x} odds(t) \;\Leftarrow\; s \xrightarrow{x} s' \wedge s' \xrightarrow{y} t$
	$evens(s) \xrightarrow{x} evens(t) \;\Leftarrow\; s \xrightarrow{y} s' \wedge s' \xrightarrow{x} t$
	$zip(s, s') \xrightarrow{x} zip(s', t) \;\Leftarrow\; s \xrightarrow{x} t$
	$zip(s, s') \xrightarrow{x} zip(s, t) \;\Leftarrow\; disabled(s) \wedge s' \xrightarrow{x} t$
	$map(f, s) \xrightarrow{f(x)} map(f, t) \;\Leftarrow\; s \xrightarrow{x} t$
	$filter(g, s) \xrightarrow{x} filter(g, t) \;\Leftarrow\; s \xrightarrow{x} t \wedge g(x) \equiv true$
	$filter(g, s) \xrightarrow{y} t' \;\Leftarrow\; s \xrightarrow{x} t \wedge g(x) \equiv false \wedge filter(g, t) \xrightarrow{y} t'$
	$enabled(s) \;\Leftarrow\; s \xrightarrow{x} t$
	$disabled(empty)$
	$disabled(odds(s)) \;\Leftarrow\; disabled(s)$
	$disabled(evens(s)) \;\Leftarrow\; disabled(s)$
	$disabled(evens(s)) \;\Leftarrow\; s \xrightarrow{x} t \wedge disabled(t)$
	$disabled(zip(s, s')) \;\Leftarrow\; disabled(s) \wedge disabled(s')$
	$disabled(map(f, s)) \;\Leftarrow\; disabled(s)$
	$disabled(filter(g, s)) \;\Leftarrow\; disabled(s)$
	$disabled(filter(g, s)) \;\Leftarrow\; s \xrightarrow{x} t \wedge g(x) \equiv false \wedge disabled(filter(g, t))$
	$finite(s) \;\Leftarrow\; disabled(s)$
	$finite(s) \;\Leftarrow\; s \xrightarrow{x} t \wedge finite(t)$
	$exists(g, s) \;\Leftarrow\; s \xrightarrow{x} t \wedge g(x) \equiv true$
	$exists(g, s) \;\Leftarrow\; s \xrightarrow{x} t \wedge exists(g, t)$
`co-Horn axioms`	$infinite(s) \;\Rightarrow\; \exists x, t : (s \xrightarrow{x} t \wedge infinite(t))$
	$forall(g, s) \;\Rightarrow\; (s \xrightarrow{x} t \;\Rightarrow\; (g(x) \equiv true \wedge forall(g, t)))$
	$fair(g, s) \;\Rightarrow\; exists(g, s)$
	$fair(g, s) \;\Rightarrow\; (s \xrightarrow{x} t \;\Rightarrow\; fair(g, t))$

In the final STREAM-model, $s \xrightarrow{x} t$ holds true if x is the first entry and t is the rest of s. $disabled$ and $enabled$ separate empty from nonempty streams. $finite$ and $infinite$ distinguish finite from infinite streams. The other function symbols and predicates are interpreted as the synomymous symbols of INFSEQ (cf. Ex. 2.8). The Horn axioms were inspired by transition system specifications given in [70, 44]. CCS-like processes can be specified coinductively in a quite similar way (see [62]). ❑

7 A modal invariance theorem

This section is devoted to the proof of Thm. 7.9. As its forerunner, [14], Thm. 4.18, it depends on a compactness theorem, which, in turn, is based on Łos' Theorem that tells us which model classes are closed under ultraproducts (cf., e.g., [12, 10, 26]). Given a swinging specification SP, we will see that $Mod(SP)$, $Mod_\equiv(SP)$, $Mod_{be}(SP)$ and $Mod_{bc}(SP)$ are all of this kind (cf. Def. 3.1).

Let I be set. $F \subseteq \wp(I)$ is a **filter over** I if

(1) $\emptyset \notin F$,

(2) $A \in F \wedge A \subseteq B \subseteq I$ implies $B \in F$, or, equivalently, $A \cap B \in F$ implies $A, B \in F$,

(3) $A, B \in F$ implies $A \cap B \in F$.

$F \subseteq \wp(I)$ has the **finite intersection property (fip)** iff the intersection of each finite subset of F is nonempty. By (3), all filters have the fip. Conversely, if F has the fip, then

$$\{A \subseteq I \mid B_1 \cap \cdots \cap B_n \subseteq A,\ B_1 \cap \cdots \cap B_n \in F\}$$

is a filter. Hence a subset of $\wp(I)$ can be extended to a filter iff it has the fip. For instance, the set $\{\mathbb{N} \setminus \{i\} \mid i \in \mathbb{N}\}$ has the fip and is thus contained in a filter.

A filter F that is maximal w.r.t. the subset relation on $\wp(I)$ is called an **ultrafilter**. A filter is an ultrafilter iff for all $A \subseteq I$, $A \in F$ or $I \setminus A \in F$.

Lemma 7.1 (Ultrafilter Theorem) *Each filter F over I can be extended to an ultrafilter.*

Proof. Let $\mathcal{F}$ be the set of all filters containing F. $\mathcal{F}$ is partially ordered by set inclusion. It is easy to show that the union of each totally ordered subset of $\mathcal{F}$ is again in $\mathcal{F}$. Hence by Zorn's Lemma, $\mathcal{F}$ has maximal element. ❑

Since $M = \{\mathbb{N} \setminus \{i\} \mid i \in \mathbb{N}\}$ has the fip, Lemma 7.1 implies that M is contained in an ultrafilter, which we denote by F_ω.

A class $\mathcal{C}$ of Σ-structures is **elementary** if there is a closed first-order Σ-formula φ such that $A \in \mathcal{C}$ iff A satisfies φ.

Let $SP = (\Sigma, AX)$ be a swinging specification. $Mod(SP)$, $Mod_{be}(SP)$ and $Mod_{bc}(SP)$ are elementary.

Definition 7.2 (ultraproducts) Let F be an ultrafilter over I, $\{A_i\}_{i \in I}$ be a family of Σ-structures and $A = \prod_{i \in I} A_i$. For all $k \in I$, let $pr_k : A \to A_i$ be the projection sending $(a_i)_{i \in I}$ to a_k. pr_k extends to a function on $\wp(A^+)$ by $pr_k(B) =_{def} \{(pr_k(a_1), \ldots, pr_k(a_n)) \mid (a_1, \ldots, a_n) \in B\}$. The **ultraproduct** A/F **of** A_i, $i \in I$, **modulo** F is the Σ-structure defined as follows:

- For all sorts $s \in \Sigma$, $(A/F)_s = A_s$.
- For all function symbols $f : s_1 \ldots s_n \to s \in \Sigma$, $a = (a_1, \ldots, a_n) \in (A/F)_{s_1 \ldots s_n}$ and $i \in I$, $pr_i(f^{A/F}(a)) = f^{A_i}(pr_i(a_1), \ldots, pr_i(a_n))$.

- For all predicates $r : s_1 \dots s_n \in \Sigma$, $(a_1, \dots, a_n) \in r^{A/F}$ iff $\{i \in I \mid (pr_i(a_1), \dots, pr_i(a_n)) \in r^{A_i}\} \in F$.

If for all $i, j \in I$, $A_i = A_j$, then A/F is called an **ultrapower of** A_i.

Given S-sorted binary relations $\approx_i \subseteq A_i \times A_i$, $i \in I$, the **ultraproduct extension of** $\approx_i$, $i \in I$, **modulo** F is the S-sorted relation $\approx \subseteq A/F \times A/F$ that is defined as follows: for all $a, b \in A/F$,

$$a \approx b \quad \Longleftrightarrow_{def} \quad \{i \in I \mid pr_i(a) \approx_i pr_i(b)\} \in F. \ \square$$

By (2) and (3), $\approx$ is a Σ-congruence if for all $i \in I$, $\approx_i$ is a Σ-congruence.

Def. 7.2 differs from the classical notion of an ultraproduct insofar as the carrier of A/F is not a quotient of $\prod_{i \in I} A_i$, but the product itself. In fact, the usual ultraproduct is the quotient of A/F by the ultraproduct extension of the equality relations on A_i, $i \in I$. These ultraproducts preserve classes of Σ-structures with $\equiv$-equality, such as $Mod_{\equiv}(SP)$. We obtain the same closure property if we first construct an ultraproduct A/F in the sense of Def. 7.2 and then factorize A/F by the ultraproduct extension $\approx$ of the equality relations on the components of A. Since

$$a \approx b \iff \{i \in I \mid pr_i(a) = pr_i(b)\} \in F \iff \{i \in I \mid pr_i(a) \equiv^{A_i} pr_i(b)\} \in F \iff a \equiv^{A/F} b,$$

the quotient of A/F by $\approx$ is indeed a Σ-structure with $\equiv$-equality.

We adapt Łos' Theorem to many-sorted signatures and the ultraproduct definition 7.2:

Theorem 7.3 (Łos' Theorem) *Let F be an ultrafilter, $\{A_i\}_{i \in I}$ be a family of Σ-structures and $A = \prod_{i \in I} A_i$. Let φ be a first-order Σ-formula and b be a valuation in A/F.*

$$A/F \models_b \varphi \quad \textit{iff} \quad \{i \in I \mid A_i \models_{pr_i \circ b} \varphi\} \in F.$$

Proof by induction on the structure of a minimal formula ψ that is equivalent to φ and built up of atoms, negation, conjunction and universal quantification. Let $J = \{i \in I \mid A_i \models_{pr_i \circ b} \psi\}$.

Case 1. ψ is an atom, say $\psi = r(t_1, \dots, t_n)$. Then

$$A/F \models_b \psi \iff (b^*(t_1), \dots, b^*(t_n)) \in r^{A/F} \iff \{i \in I \mid (pr_i(b^*(t_1)), \dots, pr_i(b^*(t_n))) \in r^{A_i}\} \in F$$
$$\iff \{i \in I \mid ((pr_i \circ b)^*(t_1), \dots, (pr_i \circ b)^*(t_n)) \in r^{A_i}\} \in F \iff J \in F.$$

Case 2. $\psi = \neg\vartheta$. Then

$$A/F \models_b \psi \iff A/F \not\models_b \vartheta \overset{ind.hyp.}{\iff} \{i \in I \mid A_i \models_{pr_i \circ b} \vartheta\} \notin F \iff I \setminus \{i \in I \mid A_i \models_{pr_i \circ b} \vartheta\} \in F$$
$$\iff \{i \in I \mid A_i \not\models_{pr_i \circ b} \vartheta\} \in F \iff J \in F.$$

Case 3. $\psi = \vartheta \wedge \delta$. Then

$$A/F \models_b \psi \iff A/F \models_b \vartheta \wedge A/F \models_b \delta \overset{ind.hyp.}{\iff} \{i \in I \mid A_i \models_{pr_i \circ b} \vartheta\} \in F \wedge \{i \in I \mid A_i \models_{pr_i \circ b} \delta\} \in F$$
$$\iff \{i \in I \mid A_i \models_{pr_i \circ b} \vartheta\} \cap \{i \in I \mid A_i \models_{pr_i \circ b} \delta\} \in F \iff J \in F.$$

Case 4. $\psi = \forall x : \vartheta$ for some $x \in X$. Then $J = \{i \in I \mid \forall a_i \in A_i : A_i \models_{(pr_i \circ b)[a_i/x]} \vartheta\} = \{i \in I \mid \forall a \in A/F : A_i \models_{pr_i \circ b[a/x]} \vartheta\}$. Hence for all $a \in A/F$, J is a subset of $J(a) =_{def} \{i \in I \mid A_i \models_{pr_i \circ b[a/x]} \vartheta\}$. Suppose that

$$J \in F \iff \forall a \in A/F : J(a) \in F. \tag{4}$$

Then

$$A/F \models_b \psi \iff \forall a \in A/F : A/F \models_{b[a/x]} \vartheta \overset{ind.hyp.}{\iff} \forall a \in A/F : J(a) \in F \overset{(4)}{\iff} J \in F.$$

Hence it remains to show (4). The "$\Rightarrow$"-part follows from $J \subseteq J(a)$. Suppose that $J \notin F$. Then $I \setminus J \in F$. For all $i \in I \setminus J$ there is $a_i \in A_i$ such that $A_i \models_{(pr_i \circ b)[a_i/x]} \neg\vartheta$. Let $a \in A/F$ such that for all $i \in I \setminus J$, $pr_i(a) = a_i$. Then $pr_i \circ b[a/x] = (pr_i \circ b)[a_i/x]$ and thus

$$I \setminus J \;\subseteq\; \{i \in I \mid A_i \models_{(pr_i \circ b)[a_i/x]} \neg\vartheta\} \;=\; \{i \in I \mid A_i \models_{pr_i \circ b[a/x]} \neg\vartheta\} \;=\; I \setminus J(a).$$

Hence by (2), $I \setminus J(a) \in F$ and thus $J(a) \notin F$. This completes the proof of the "$\Leftarrow$"-part of (4). ❏

An immediate consequence is the following:

Corollary 7.4 *All elementary classes of Σ-structures are closed under ultraproducts.* ❏

Corollary 7.5 (Compactness Theorem) *Let Γ be a set of first-order Σ-formulas and $\mathcal{C}$ be a class Σ-structures that is closed under ultraproducts.*

(1) If for all finite subsets Γ' of Γ there are $A \in \mathcal{C}$ and $b : X \to A$ such that $A \models_b \Gamma'$, then there are $B \in \mathcal{C}$ and $c : X \to B$ such that $B \models_c \Gamma$.

(2) Let φ be a first-order Σ-formula. If $\mathcal{C} \models \bigwedge\Gamma \Rightarrow \varphi$, then there is a finite subset Γ' of Γ such that $\mathcal{C} \models \bigwedge\Gamma' \Rightarrow \varphi$.

Proof. (1) Let Γ^+ be the set finite conjunctions of elements of Γ. By assumption, for all $\varphi \in \Gamma^+$ there are $A_\varphi \in \mathcal{C}$ and $b_\varphi : X \to A_\varphi$ such that $A_\varphi \models_{b_\varphi} \varphi$. Let $A = \prod_{\varphi \in \Gamma^+} A_\varphi$ and for all finite conjunctions φ of elements of Γ, let $D_\varphi = \{\psi \in \Gamma \mid A_\psi \models_{b_\psi} \varphi\}$. Since for all $\varphi_1, \ldots, \varphi_n \in \Gamma^+$, $D_{\varphi_1} \cap \cdots \cap D_{\varphi_n} = D_{\varphi_1 \wedge \cdots \wedge \varphi_n}$, $\mathcal{S} = \{D_\varphi \mid \varphi \in \Gamma^+\}$ has the fip and thus can be extended to an ultrafilter F. We define $c : X \to A/F$ by $pr_\varphi \circ c = b_\varphi$ for all $\varphi \in \Gamma^+$. By Thm. 7.3,

$$A/F \models_c \varphi \iff \{\psi \in \Gamma \mid A_\psi \models_{pr_\psi \circ c} \varphi\} = \{\psi \in \Gamma \mid A_\psi \models_{b_\psi} \varphi\} = D_\varphi \in F.$$

But $D_\varphi \in F$ follows from the construction of F. Hence (1) holds true for $B = A/F$.

(2) Suppose that for all finite subsets Γ' of Γ there are $A \in \mathcal{C}$ and $b : X \to A$ such that $A \not\models_b \bigwedge\Gamma' \Rightarrow \varphi$ and thus $A \models_b \bigwedge\Gamma' \wedge \neg\varphi$. Then for all finite subsets Γ' of $\Gamma \cup \{\neg\varphi\}$ there are $A \in \mathcal{C}$ and $b : X \to A$ with $A \models_b \bigwedge\Gamma'$. Hence by (1), there are $B \in \mathcal{C}$ and $c : X \to B$ such that $B \models_c \Gamma \cup \{\neg\varphi\}$ and thus $B \not\models_c \bigwedge\Gamma \Rightarrow \varphi$. We conclude $\mathcal{C} \models \bigwedge\Gamma \Rightarrow \varphi$. ❏

A Σ-structure A is **ω-saturated** if for each countable set Γ of first-order Σ-formulas the following holds true: if for all finite subsets Γ' of Γ there is $b : X \to A$ such that $A \models_b \Gamma'$, then there is $c : X \to A$ such that $A \models_c \Gamma$.

Given Σ-structures A and B, an injective S-sorted function $h : A \to B$ is an **elementary embedding of A in B** if for all first-order Σ-formulas φ and valuations b in A, $A \models_b \varphi$ iff $B \models_{h \circ b} \varphi$. We say that A is **elementarily embedded in B**.

Theorem 7.6 *Each Σ-structure A is elementarily embedded in an ω-saturated ultrapower of A.*

Proof.[13] Since the set $\{\mathbb{N} \setminus \{i\} \mid i \in \mathbb{N}\}$ has the fip, it can be extended to an ultrafilter F. The function $h : A \to A^{\mathbb{N}}$ defined by $h(a) = (a, a, a, \ldots)$ embeds A in $A^{\mathbb{N}}/F$. h is an elementary embedding because by Thm. 7.3, for all first-order formulas φ and $b : X \to A$,

$$A^{\mathbb{N}}/F \models_{h \circ b} \varphi \iff \{i \in \mathbb{N} \mid A \models_{pr_i \circ h \circ b} \varphi\} \in F \iff \{i \in \mathbb{N} \mid A \models_b \varphi\} \in F \overset{\emptyset \notin F,\ I \in F}{\iff} A \models_b \varphi.$$

[13] The proof proceeds along the lines of the proofs of [12], Lemma 2.3, and [26], Thm. 8.5.

We claim that $A^{\mathbb{N}}/F$ is ω-saturated. Let $\Gamma = \{\varphi_0, \varphi_1, \varphi_2, \ldots\}$ be a countable set of first-order formulas. Suppose that for all finite subsets Γ' of Γ there is $b : X \to A^{\mathbb{N}}/F$ such that $A^{\mathbb{N}}/F \models_b \Gamma'$. Then, in particular, for all $k \in \mathbb{N}$ there is $b_k : X \to A^{\mathbb{N}}/F$ such that $A^{\mathbb{N}}/F \models_{b_k} \varphi_0 \wedge \cdots \wedge \varphi_k$.

Let $k \in \mathbb{N}$. By Thm. 7.3, $\{i \in \mathbb{N} \mid A \models_{pr_i \circ b_k} \varphi_0 \wedge \cdots \wedge \varphi_k\} \in F$. Since $\emptyset \notin F$, there is $f(k) \in \mathbb{N}$ such that $A \models_{pr_{f(k)} \circ b_k} \varphi_0 \wedge \cdots \wedge \varphi_k$. We define $c : X \to A^{\mathbb{N}}/F$ by $pr_i \circ c = pr_{f(i)} \circ b_i$ for all $i \in \mathbb{N}$. Since k was chosen arbitrarily, we obtain

$$\forall i \geq k : A \models_{pr_i \circ c} \varphi_k. \tag{5}$$

Moreover, by Thm. 7.3,

$$A^{\mathbb{N}}/F \models_c \varphi_k \iff D_k =_{def} \{i \in \mathbb{N} \mid A \models_{pr_i \circ c} \varphi_k\} \in F. \tag{6}$$

Since for all $i \in \mathbb{N}$, $\mathbb{N} \setminus \{i\} \in F$, $E_k =_{def} \{i \in \mathbb{N} \mid i \geq k\} = \bigcap_{i=0}^{k-1} (\mathbb{N} \setminus \{i\}) \in F$. By (6), E_k is a subset of D_k. Hence $E_k \in F$ implies $D_k \in F$ and thus by (7), $A^{\mathbb{N}}/F \models_c \varphi_k$. We conclude that $A^{\mathbb{N}}/F$ is ω-saturated. ❑

From now on we follow the proof of Benthem's Invariance Theorem ([14], Thm. 4.18) in order to obtain our modal invariance theorem.

Given a Σ-structure A and $a \in A$, a modal formula $\varphi(x)$ with $A \models_{a/x} \varphi(x)$ is called a **modal theorem of** a (cf. Def. 2.3). **mod(a)** denotes the set of all modal theorems of a. Given Σ-structures A and B, $a \in A$ and $b \in B$, a and b are **modally equivalent** if $mod(a) = mod(b)$.

Lemma 7.7 *a and b are modally equivalent iff $mod(a) \subseteq mod(b)$ or $mod(b) \subseteq mod(a)$.*

Proof. W.l.o.g. let $mod(a) \subseteq mod(b)$. Assume that there is $\varphi(x) \in mod(b) \setminus mod(a)$. Then $A \models_{a/x} \neg\varphi(x)$. Hence $\neg\varphi(x)$ is a modal theorem of a. Since $mod(a) \subseteq mod(b)$, we conclude that $\neg\varphi(x)$ is a modal theorem of b, which contradicts the assumption that $\varphi(x)$ is also a modal theorem of b. Hence $mod(a) = mod(b)$. ❑

Lemma 7.8 *Let A and B be ω-saturated Σ-structures. Then $\approx \subseteq A \times B$ defined by: $a \approx b$ iff $mod(a) = mod(b)$ is a bisimulation (cf. Def. 2.3).*

Proof. Let $s_1, \ldots, s_n \in S$, $1 \leq i \leq n$, $a \in A_{s_i}$, $b \in B_{s_i}$ and $t_j \in T_{\Sigma, s_j}$ for all $1 \leq j \neq i \leq n$ such that $a \approx b$, i.e. for all modal formulas $\varphi(x)$, $A \models_{a/x} \varphi(x)$ iff $B \models_{b/x} \varphi(x)$.

Let $f : s_1 \ldots s_n \to s$ be a function symbol. Then for all modal formulas $\varphi(x)$,

$$\begin{aligned} & A \models_{f^A(t_1^A, \ldots, a, \ldots, t_n^A)/x} \varphi(x) \iff A \models_{a/x} \varphi(f(t_1, \ldots, x, \ldots, t_n)) \\ & \iff B \models_{b/x} \varphi(f(t_1, \ldots, x, \ldots, t_n)) \iff B \models_{f^B(t_1^B, \ldots, a, \ldots, t_n^B)/x} \varphi(x). \end{aligned}$$

Hence $f^A(t_1^A, \ldots, a, \ldots, t_n^A) \approx f^B(t_1^B, \ldots, b, \ldots, t_n^B)$.

Let $r : s_1 \ldots s_n$ be a static predicate. Since $\varphi(x) = r(t_1, \ldots, x, \ldots, t_n)$ is a modal formula, $A \models_{a/x} \varphi(x)$ iff $B \models_{b/x} \varphi(x)$. Hence $(t_1^A, \ldots, a, \ldots, t_n^A) \in r^A$ iff $(t_1^B, \ldots, b, \ldots, t_n^B) \in r^B$.

Let $\delta : s_1 \ldots s_n s$ be a dynamic predicate, $a' \in A_s$ and $b' \in B_s$. We must show

$$(t_1^A, \ldots, a, \ldots, t_n^A, a') \in \delta^A \quad \text{implies} \quad \exists\, b' \in B : (t_1^B, \ldots, b, \ldots, t_n^B, b') \in \delta^B \ \wedge\ a' \approx b', \tag{1}$$

$$(t_1^B, \ldots, b, \ldots, t_n^B, b') \in \delta^B \quad \text{implies} \quad \exists\, a' \in A : (t_1^A, \ldots, a, \ldots, t_n^A, a') \in \delta^A \ \wedge\ a' \approx b'. \tag{2}$$

We show (1). (2) can be proved analogously. Let $(t_1^A, \ldots, a, \ldots, t_n^A, a') \in \delta^A$. By Def. 2.3, for all modal theorems $\varphi(y)$ of a', $\exists y(\delta(x, y) \wedge \varphi(y))$ is a modal theorem of a. Since $a \approx b$, $A \models_{a/x} \exists y(\delta(x, y) \wedge \varphi(y))$ implies $B \models_{b/x} \exists x(\delta(x, y) \wedge \varphi(y))$. Hence for all $\varphi(y) \in mod(a')$ there is $b_\varphi \in B$ such that $B \models_{b_\varphi/y} \varphi(y)$. Since B

is ω-saturated, there is $b' \in B$ such that for all $\varphi \in mod(a')$, $B \models_{b'/y} \varphi(y)$. Hence all modal theorems of a' are modal theorems of b' and thus by Lemma 7.7, a' and b' are modally equivalent. ❑

Theorem 7.9 (Modal Invariance Theorem) *Let φ be a unary first-order formula that is bisimulation invariant in an elementary class $\mathcal{C}$ of Σ-structures (with or without $\equiv$-equality). Then φ is modal in $\mathcal{C}$.*

Proof. Let $\varphi = \varphi(x)$ and Γ be the set of modal formulas $\psi = \psi(x)$ such that $\mathcal{C}$ satisfies $\varphi \Rightarrow \psi$. Suppose that $\mathcal{C}$ satisfies $\bigwedge \Gamma \Rightarrow \varphi$. Then by Cor. 7.5(2), there is a finite subset $\{\psi_1, \dots, \psi_n\}$ of Γ such that $\mathcal{C}$ satisfies $(\psi_1 \wedge \cdots \wedge \psi_n) \Rightarrow \varphi$. By the definition of Γ, we conclude that φ and $\psi_1 \wedge \cdots \wedge \psi_n$ are equivalent in $\mathcal{C}$. Hence it remains to show $\mathcal{C} \models \bigwedge \Gamma \Rightarrow \varphi$.

Let $A \in \mathcal{C}$ and $a \in A$ such that $A \models_{a/x} \Gamma$. Suppose that

$$\text{for all finite subsets } \Phi \text{ of } mod(a) \text{ there are } B \in \mathcal{C} \text{ and } b \in B \text{ such that } B \models_{b/x} \varphi \wedge \bigwedge \Phi. \tag{1}$$

By Cor. 7.5(1), (1) implies $B \models_{b/x} mod(a) \cup \{\varphi\}$ for some $B \in \mathcal{C}$ and $b \in B$. Hence $mod(a) = mod(b)$. By Thm. 7.6, A and B are second-order embedded in ω-saturated extensions A^+ resp. B^+. Since $\mathcal{C}$ is second-order definable, $A, B \in \mathcal{C}$ implies $A^+, B^+ \in \mathcal{C}$. Moreover, $A \models_{a/x} mod(a)$ implies $A^+ \models_{g(a)/x} mod(a)$ and $B \models_{b/x} mod(a) \cup \{\varphi\}$ implies $B^+ \models_{h(b)/x}$ mod $(a) \cup \{\varphi\}$ where g and h are the embeddings of A and B in A^+ resp. B^+. Hence $mod(g(a)) = mod(a) = mod(b) = mod(h(b))$, i.e. $h(b)$ and $g(a)$ are modally equivalent and thus by Lemma 7.8, $(h(b), g(a))$ belongs to a bisimulation. Since φ is bisimulation invariant in $\mathcal{C}$ and $A^+, B^+ \in \mathcal{C}$, $B^+ \models_{h(b)/x} \varphi$ implies $A^+ \models_{g(a)/x} \varphi$ and thus $A \models_{a/x} \varphi$.

It remains to show (1). Assume that there is a finite subset Φ of $mod(a)$ such that for all $B \in \mathcal{C}$ and $b \in B$, $B \not\models_{b/x} \varphi \wedge \bigwedge \Phi$. Then $B \models_{b/x} \varphi \Rightarrow \neg \bigwedge \Phi$. Hence $\neg \bigwedge \Phi \in \Gamma$ and thus $A \models_{a/x} \neg \bigwedge \Phi$ because $A \models_{a/x} \Gamma$. But $\Phi \subseteq mod(a)$ implies $A \models_{a/x} \bigwedge \Phi$ and thus $A \not\models_{a/x} \neg \bigwedge \Phi$, which contradicts $A \models_{a/x} \neg \bigwedge \Phi$.
❑

Corollary 7.10 *Let φ be a unary first-order formula that is bisimulation invariant in $Mod(SP)$, $Mod_{\equiv}(SP)$, $Mod_{be}(SP)$ or $Mod_{bc}(SP)$. Then φ is modal in $Mod(SP)$, $Mod_{\equiv}(SP)$, $Mod_{be}(SP)$ or $Mod_{bc}(SP)$, respectively.* ❑

8 Conclusion

We have introduced swinging types as a specification formalism that covers functional, relational and state-oriented "transitional" techniques. The approach developed here differs considerably from the preliminary versions given in [58, 61]. Swinging types combine the dominant algebraic touch of other data type presentations with concepts, results and methods obtained in relational semantics, modal logic, higher-order functional programming and Horn clause rewriting. The integration of functions and relations becomes particularly evident in the possibility to use defined functions *or* static *or* dynamic predicates as observers that determine the behavior axioms and thus the interpretation of behavioral equality.

Since the number of observers raises the number of behavior axioms and thus the number of cases produced by unfolding behavioral equivalences, only a few functions or predicates should be declared as observers. For most behavioral equalities, one or two observers turn out to be sufficient (cf. [62]). Behavioral consistency and behavioral term replacement require that $\sim$ is a weak congruence. For this purpose we have established coinductivity as a—mainly syntactic—property of a swinging specification that ensures weak congruence and covers most other congruence criteria to be found in the literature on hidden/observational or process types. From a practical point of view, more general weak-congruence criteria than coinductivity are not needed.

However, special cases should be distinguished from each other and establish a classification of "coinductive program schemas" as part of a design methodology for swinging types. Different schemas may correspond to different application areas and lead to tailor-made verification and transformation rules.

Coinductivity is accompanied by the other indispensable requirement, namely functionality, which means intuitively that all data presented by the type have unique normal form representations (w.r.t. structural equivalence). By Thm. 4.10, criteria for functionality reduce to criteria for confluence (see, e.g., [60], Sect. 10.5). Functionality is indispensible for verifying defined functions of a swinging type: applications of fixpoint induction on defined functions, term unfolding, term splitting and clash may not be correct if the specification is not functional (see Section 4).

The third main condition besides functionality and coinductivity is image finiteness, which ensures that the consequence operators that build up the Herbrand model are continuous and thus admit inductive arguments on predicates such as, for instance, in the proof of Theorem 6.5.

The syntactic structure of swinging specifications is motivated by the intended applications as well as the goal to obtain simple Herbrand models that both reflect the specifier's intuitive models and can be reasoned about formally with the help of powerful proof rules. These provide the basis for test and proof procedures that are still to be worked out and implemented. Their development should be guided by case studies along the lines of [62] and integrated into the development of design methods based on swinging types. Case studies are also needed for investigating the range of traditional methods and applications that could be covered by this approach.

The paper also defines and discusses hierarchical relationships between several swinging types such as (relative) completeness, monotonicity, consistency and inductive equivalence (see Section 5). Lemma 5.7 and Cor. 5.8 show when and how they can be reduced to inductive theorems of the involved types. Consistency criteria based on confluence (cf. Thm. 4.10) can be found in, e.g., [60], Section 5. In particular, a functional specification is complete, monotone and consistent with respect to its three sublevels (Lemma 5.9). Structured types involving several swinging specifications, in particular specifications *with import* and refinements, are the topic of [64]. They admit, for instance, the specification of defined functions in terms of ν-predicates or to implement visible by hidden sorts and structural by behavioral equalities.

An open question are the practical consequences of Theorems 3.8(1) and 7.9 stating that a first-order formula over a swinging signature Σ is modal iff it is bisimulation invariant in a given elementary class of Σ-structures.

References

[1] *Abstract State Machines (aka Gurevich Machines or Evolving Algebras)*, http://www.eecs.umich.edu/gasm

[2] K.R. Apt, H.A. Blair, A. Walker, *Towards a Theory of Declarative Knowledge*, in: J. Minker, ed., Deductive Databases and Logic Programming, Morgan Kaufmann (1988) 89-148

[3] M.A. Arbib, E.G. Manes, *Parametrized Data Types Do Not Need Highly Constrained Parameters*, Information and Control 52 (1982) 139-158

[4] E. Astesiano, M. Broy, G. Reggio, *Algebraic Specification of Concurrent Systems*, in [6]

[5] E. Astesiano, A. Giovini, G. Reggio, *Observational Structures and their Logic*, Theoretical Computer Science 96 (1992) 249-283

[6] E. Astesiano, H.-J. Kreowski, B. Krieg-Brückner, eds., *Algebraic Foundations of Systems Specification*, IFIP State-of-the-Art Report, Springer 1999

[7] E. Astesiano, G. Reggio, *Algebraic Specification of Concurrency*, Proc. WADT'91, Springer LNCS 655 (1993) 1-39

[8] E. Astesiano, M. Wirsing, *Bisimulation in Algebraic Specifications*, in: H. Ait-Kaci, M. Nivat, eds., Resolution of Equations in Algebraic Structures 1, Academic Press (1989) 1-31

[9] J.C.M. Baeten, W.P. Weijland, *Process Algebra*, Cambridge University Press 1990

[10] D.W. Barnes, J.M. Mack, *Algebraic Introduction to Mathematical Logic*, Springer 1975

[11] J. Barwise, L. Moss, *Vicious Circles: On the Mathematics of Non-Wellfounded Phenomena*, CSLI Publications, Stanford 1996

[12] J.L. Bell, A.B. Slomson, *Models and Ultraproducts: An Introduction*, North-Holland 1969

[13] J. van Benthem, J. Bergstra, *Logic of Transition Systems*, J. Logic, Language, and Information 3 (1995) 247-283

[14] J. van Benthem, *Exploring Logical Dynamics*, CSLI Publications, Stanford 1996

[15] M. Bidoit, R. Hennicker, *Proving the Correctness of Behavioural Implementations*, Proc. AMAST '95, Springer LNCS 936 (1995) 152-168

[16] M. Bidoit, R. Hennicker, *Observer Complete Definitions are Behaviourally Coherent*, Report, University of Munich (1999)

[17] M. Bidoit, R. Hennicker, M. Wirsing, *Behavioural and Abstractor Specifications*, Science of Computer Programming 25 (1995) 149-186

[18] M. Broy, M. Wirsing, *Partial Abstract Types*, Acta Informatica 18 (1982) 47-64

[19] The CoFI Task Group on Language Design, *CASL: The Common Algebraic Specification Language*, 1998, http://www.brics.dk/Projects/CoFI/Documents/CASL/Summary

[20] A. Corradini, R. Heckel, U. Montanari, *From SOS Specifications to Structured Coalgebras: How to Make a Bisimulation a Congruence*, Proc. CMCS '99, Elsevier ENTCS 19 (1999)

[21] G. Costa, G. Reggio, *Specification of Abstract Dynamic Data Types: A Temporal Logic Approach*, Theoretical Computer Science 173 (1997) 513-554

[22] R. Diaconescu, K. Futatsugi, *CafeOBJ Report*, AMAST Series in Computing 6, World Scientific 1998

[23] H. Ehrig, H.-J. Kreowski, B. Mahr, P. Padawitz, *Algebraic Implementation of Abstract Data Types*, Theoretical Computer Science 20 (1982) 209-263

[24] H. Ehrig, B. Mahr, *Fundamentals of Algebraic Specification 1*, Springer 1985

[25] H. Ehrig, F. Orejas, *Dynamic Abstract Data Types: An Informal Proposal*, EATCS Bulletin 53 (June 1994) 162-169

[26] P.C. Eklof, *Ultraproducts for Algebraists*, in: J. Barwise, ed., Handbook of Mathematical Logic, North-Holland (1977) 105-137

[27] E.A. Emerson, *Temporal and Modal Logic*, in: J. van Leeuwen, ed., Handbook of Theoretical Computer Science, Elsevier (1990) 995-1072

[28] V. Giarratana, F. Gimona, U. Montanari, *Observability Concepts in Abstract Data Type Specifications*, Proc. MFCS '76, Springer LNCS 45 (1976) 576-587

[29] J.A. Goguen, R. Diaconescu, *Towards an Algebraic Semantics for the Object Paradigm*, Proc. 9th ADT Workshop, Springer LNCS 785 (1994) 1-29

[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, 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

[32] J.A. Goguen, G. Malcolm, *A Hidden Agenda*, UCSD Technical Report CS97-538, San Diego 1997

[33] J.A. Goguen, G. Malcolm, *Hidden Coinduction: Behavioral Correctness Proofs for Objects*, to appear in MSCS

[34] 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

[35] 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

[36] A.D. Gordon, *A Tutorial on Co-induction and Functional Programming*, Proc. Functional Programming Glasgow 1994, Springer (1995) 78-95

[37] J.F. Groote, F. Vaandrager, *Structured Operational Semantics and Bisimulation as a Congruence*, Information and Computation 100 (1992) 202-260

[38] J. Guttag, E. Horowitz, D.R. Musser *Abstract Data Types and Software Validation*, Report ISI/RR-76-48, University of Southern California 1976

[39] T. Hagino, *Codatatypes in ML*, J. Symbolic Computation (1989) 629-650

[40] M. Hennessy, R. Milner, *Algebraic Laws for Nondeterminism and Concurrency*, J. of the ACM 32 (1985) 137-161

[41] R. Hennicker, M. Bidoit, *Observational Logic*, Proc. AMAST '98, Springer LNCS 1548 (1998) 263-277

[42] C.A.R. Hoare, *Proof of Correctness of Data Representations*, Acta Informatica 1 (1972) 271-281

[43] B. Jacobs, *Behaviour-Refinement of Coalgebraic Specifications with Coinductive Correctness Proofs*, Proc. TAPSOFT '97, Springer LNCS 1214 (1997) 787-802

[44] B. Jacobs, J. Rutten, *A Tutorial on (Co)Algebras and (Co)Induction*, EATCS Bulletin 62 (June 1997) 222-259

[45] J.-P. Jouannaud, H. Kirchner, *Completion of a Set of Rules Modulo a Set of Equations*, SIAM J. Computing 15 (1986) 1155-1194

[46] S. Kamin, *Final Data Type Specifications: A New Data Type Specification Method*, ACM TOPLAS 5 (1983) 97-123

[47] U. Kühler, C.-P. Wirth, *Conditional Equational Specifications of Data Types with Partial Operations for Inductive Theorem Proving*, Proc. RTA '97, Springer LNCS 1232 (1997) 38-52

[48] K.G. Larsen, *Proof Systems for Hennessy-Milner Logic with Recursion*, Proc. CAAP '88, Springer LNCS 299 (1988) 215-230

[49] J.-L. Lassez, V.L. Nguyen, E.A. Sonenberg, *Fixed Point Theorems and Semantics: A Folk Tale*, Information Processing Letters 14 (1982) 112-116

[50] G. Malcolm, J.A. Goguen, *Proving Correctness of Refinement and Implementation*, Technical Monograph PRG-114, Oxford University Computing Lab 1994

[51] 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

[52] J. Meseguer, *Membership Algebra as a Logical Framework for Equational Specification*, Proc. WADT '97, Springer LNCS 1376 (1998) 18-61

[53] B. Möller, A. Tarlecki, M. Wirsing, *Algebraic Specifications of Reachable Higher-Order Algebras*, Proc. 5th ADT Workshop, Springer LNCS 332 (1988) 154-169

[54] E. Moggi, *Notions of Computation and Monads*, Information and Computation 93 (1991) 55-92

[55] H.-J. Ohlbach, *Semantic-Based Translation Methods for Modal Logics*, J. Logic and Computation 1 (1991) 691-746

[56] P. Padawitz, *Computing in Horn Clause Theories*, Springer 1988

[57] P. Padawitz, *Deduction and Declarative Programming*, Cambridge University Press 1992

[58] P. Padawitz, *Swinging Data Types: Syntax, Semantics, and Theory*, Proc. WADT '95, Springer LNCS 1130 (1996) 409-435

[59] P. Padawitz, *Inductive Theorem Proving for Design Specifications*, J. Symbolic Computation 21 (1996) 41-99

[60] P. Padawitz, *Proof in Flat Specifications*, in [6]

[61] P. Padawitz, *Towards the One-Tiered Design of Data Types and Transition Systems*, Proc. WADT '97, Springer LNCS 1376 (1998) 365-380

[62] P. Padawitz, *Sample Swinging Types*, Report, University of Dortmund 1998, http://ls5.cs.uni-dortmund.de/~peter/BehExa.ps.gz

[63] P. Padawitz, *Theorie der Programmierung*, Course Notes, University of Dortmund 1998, http://ls5.cs.uni-dortmund.de/~peter/TdP96.ps.gz

[64] P. Padawitz, *Modular Swinging Types*, Report, University of Dortmund 1999, http://ls5.cs.uni-dortmund.de/~peter/MST.ps.gz

[65] D. Park, *Fixpoint Induction and Proofs of Program Properties*, in: B. Meltzer, D. Michie, eds., Machine Intelligence 5, Elsevier (1969) 59-78

[66] G.D. Plotkin, *Building-in Equational Theories*, in: B. Meltzer, D. Michie, eds., Machine Intelligence 7, Elsevier (1972) 73-90

[67] G.D. Plotkin, *An Operational Semantics for CSP*, in: D. Bjørner, ed., Proc. IFIP TC-2 Working Conf. Formal Description of Programming Concepts II, North-Holland (1983) 199-225

[68] H. Reichel, *An Approach to Object Semantics based on Terminal Coalgebras*, Math. Structures in Comp. Sci. 5 (1995) 129-152

[69] G. Roşu, J. Goguen, *Hidden Congruent Deduction*, Proc. First-Order Theorem proving - FTP'98, Vienna (1998) 213-223

[70] J.J.M.M. Rutten, *Universal Coalgebra: A Theory of Systems*, Report CS-R9652, CWI, SMC Amsterdam 1996

[71] D. Sannella, A. Tarlecki, *Toward Formal Development of Programs from Algebraic Specifications: Implementations Revisited*, Acta Informatica 25 (1988) 233-281

[72] M. Stickel, *Automated Deduction by Theory Resolution*, J. Automated Reasoning 1 (1985) 333-356

[73] C. Stirling, *Modal and Temporal Logics*, in: S. Abramsky et al., eds., Handbook of Logic in Computer Science, Clarendon Press (1992) 477-563

[74] C. Stirling, *The Joys of Bisimulation*, Proc. MFCS '98, Springer LNCS 1450 (1998) 142-151

[75] M. Wand, *Final Algebra Semantics and Data Type Extensions*, J. Computer and System Sciences 19 (1979) 27-44

[76] M. Wand, *Specifications, Models, and Implementations of Data Abstractions*, Theoretical Computer Science 20 (1982) 3-32

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