Structural Induction in Institutions

Răzvan Diaconescu

Simion Stoilow Institute of Mathematics of the Romanian Academy

Abstract

We develop a general logic-independent structural induction proof method at the level of abstract institutions. This provides a solid and uniform mathematical foundations to induction proof methodologies for a wide variety of actual logic-based formal specification frameworks. Our development is based technically upon an axiomatic approach to substitutions within institution theory.

Key words: Structural induction, Institution theory, Algebraic specification

1. Introduction

Since its introduction within computing science by Burstall [5] structural induction has become a major method for performing inductive proofs, which constitute one of the most important formal verification trends. Originally structural induction was confined to proving properties of abstract data types, specified within many-sorted algebra (MSA). But over the past decades due to the population explosion of underlying logics for specification formalisms, the meaning and scope of structural induction has been extended to logical systems that are increasingly sophisticated and different from MSA. However these structural induction proof methodologies are often developed on a rather ad-hoc basis without clear mathematical foundations, a situation that in our opinion ultimately undermines the credibility of the associated formal methods.

Here we develop a generic method for proving inductive properties, that is directly applicable to wide variety of logic based specification formalisms, already in existence or that may be developed in the future. The genericity of our structural induction method is given by the fact that it is developed at the level of abstract institutions, and it therefore lacks a commitment to a particular logical system.

Institution theory [21] is a categorical abstract model theory that arose within specification theory as a response to the explosion in the population of logics in use there, its original aim being to develop as much computing science as possible in a general uniform way independently of particular logical systems. While this, often known as 'institution-independent computing science', has been achieved to an impressive extent, probably greater than originally thought, in parallel (but not disconnected) a similar 'institution-independent' development has happened this time fuelled by model theoretic motivations [14]. From this perspective, our work may be seen as part of the 'institution-independence' program that has been undertaken since three decades in computing science and in model theory.

Apart from providing a generic logic-independent proof method for inductive properties that is based upon solid and clear mathematical foundations, we think that an important aspect of our work is a logic-independent clarification of the essence of the structural induction and its relation to the model theory of induction. This may be described quite informally in a simplified manner as follows:

Email address: Razvan.Diaconescu@imar.ro (Răzvan Diaconescu)

– Let 0_Γ denote the initial model (when it exists!) of a given specification Γ (considered as a set of sentences in a fixed logical system). An *inductive property* for Γ is any sentence ρ that holds in 0_Γ, i.e. $0_\Gamma \models \rho$.

– Then in the case of universally quantified sentences $(\forall X)\rho$ (N.B. here ρ may be any sentence, *not* necessarily a quantifier-free one) the checking of inductive property gets reduced to a (possibly infinite) set of 'ordinary' deductions:

$$0_\Gamma \models (\forall X)\rho \quad \text{if} \quad \Gamma \models \theta(\rho) \ \text{ for all 'substitutions' } \ \theta : X \to T_\Sigma \tag{1}$$

where here T_Σ denotes the 'set of terms' for the signature of Γ. The actual concept of 'substitution' is of course dependent upon the underlying logical system. The problem with the condition of (1) is that in general it represents an *infinite* set of proof tasks.

– The structural induction method is just a sufficient condition for the condition of (1) but which in actual situations represents a *finitary* proof method, its very essence being a Peano induction on the 'depth' of the 'substitutions' θ. This may involve pure methodological artifacts, most notably the so-called 'sub-signatures of constructors', playing a role only for the efficiency of the proof method.

Contributions and structure of the paper

1. The first technical section briefly recalls institution theory concepts that are necessary for our work here. We also introduce a number of examples of institutions that will constitute concrete benchmark examples for the concepts and results about structural induction in abstract institutions that are developed in this paper.
2. The next section develops an axiomatic theory of substitutions for abstract institutions that is based upon and refines the general institution-independent concept of substitution introduced in [13] (see also [14]). This serves as the technical ground for the development of our institution-independent structural induction method.
3. The core result of this work, namely the structural induction theorem (Thm. 4.1), constitutes the topic of the first part of the third technical section. The second part of this section is devoted to instances of this result in actual logical systems, all representing rigorous formulations of concrete induction proof methodologies. A particularly important feature of these methodologies that differs from other formulations of structural induction in the literature (in fact mostly within MSA) is that they allow *simultaneous induction on several variables*. This owes to the fact that we do not restrict X of $(\forall X)\rho$ to a single variable, it may rather represent a block of variables. This comes naturally from approaching the concepts of variable and substitution from an abstract institution theoretic perspective.
4. The next section is devoted to establishing the relation (1) above within the abstract institution theoretic setting and to a theory of 'constructors', at the same level of generality.
5. The final technical section is devoted to the illustration of the practical applicability of our theoretical results through several examples of formal verification proof scores written in CafeOBJ and Maude languages. These proof scores are based directly and rigidly upon some of the concrete instances of our institution-independent structural induction method.

2. Institution-theoretic preliminaries

This section is meant to recall the institution-theoretic concepts that are necessary for our work here. Its contents are as follows.

1. We recall the definition of institutions.
2. We give a brief presentation of five examples of institutions that are relevant to computing science and formal specification and that will constitute the benchmark of concrete examples for our abstract developments.
3. We recall the concept of model amalgamation in institutions.

2.1. Categories

Institution theory relies heavily upon category theory. We assume the reader is familiar with basic notions and standard notations from category theory. With few exceptions, in general we follow the terminology and the notations of [27]. With respect to notational conventions, $|\mathbb{C}|$ denotes the class of objects of a category $\mathbb{C}$, $\mathbb{C}(A, B)$ the set of arrows (morphisms) with domain A and codomain B, and composition is denoted by ";" and in diagrammatic order. The category of sets (as objects) and functions (as arrows) is denoted by $\mathbb{S}et$, and $\mathbb{CAT}$ is the category of all categories.[1]

2.2. Institutions

Institutions have been defined by Goguen and Burstall in [6, 21]. Below we recall the concept of institution which formalizes the intuitive notion of logical system, including syntax, semantics, and the satisfaction between them.

Definition 2.1 (Institutions). *An* institution $\mathcal{I} = (\mathbb{S}ig^{\mathcal{I}}, \mathsf{Sen}^{\mathcal{I}}, \mathsf{Mod}^{\mathcal{I}}, \models^{\mathcal{I}})$ *consists of*

1. *a category* $\mathbb{S}ig^{\mathcal{I}}$*, whose objects are called* signatures*,*
2. *a functor* $\mathsf{Sen}^{\mathcal{I}} : \mathbb{S}ig^{\mathcal{I}} \to \mathbb{S}et$*, giving for each signature a set whose elements are called* sentences *over that signature,*
3. *a functor* $\mathsf{Mod}^{\mathcal{I}} : (\mathbb{S}ig^{\mathcal{I}})^{\mathrm{op}} \to \mathbb{CAT}$ *giving for each signature* Σ *a category whose objects are called* Σ-models*, and whose arrows are called* Σ*-(*model*)* homomorphisms*, and*
4. *a relation* $\models^{\mathcal{I}}_{\Sigma} \subseteq |\mathsf{Mod}^{\mathcal{I}}(\Sigma)| \times \mathsf{Sen}^{\mathcal{I}}(\Sigma)$ *for each* $\Sigma \in |\mathbb{S}ig^{\mathcal{I}}|$*, called* Σ-satisfaction*,*

such that for each morphism $\varphi : \Sigma \to \Sigma'$ *in* $\mathbb{S}ig^{\mathcal{I}}$*, the* satisfaction condition

$$M' \models^{\mathcal{I}}_{\Sigma'} \mathsf{Sen}^{\mathcal{I}}(\varphi)(\rho) \text{ if and only if } \mathsf{Mod}^{\mathcal{I}}(\varphi)(M') \models^{\mathcal{I}}_{\Sigma} \rho$$

holds for each $M' \in |\mathsf{Mod}^{\mathcal{I}}(\Sigma')|$ *and* $\rho \in \mathsf{Sen}^{\mathcal{I}}(\Sigma)$*. We denote the* reduct *functor* $\mathsf{Mod}^{\mathcal{I}}(\varphi)$ *by* $\_{\restriction_\varphi}$ *and the sentence translation* $\mathsf{Sen}^{\mathcal{I}}(\varphi)$ *by* $\varphi(\_)$*. When* $M = M'{\restriction_\varphi}$ *we say that* M *is a* φ-reduct of M'*, and that* M' *is a* φ-expansion of M*. When there is no danger of ambiguity, we may skip the superscripts from the notations of the entities of the institution; for example* $\mathbb{S}ig^{\mathcal{I}}$ *may be simply denoted* $\mathbb{S}ig$*.*

General assumption: We assume that model isomorphisms preserve the satisfaction of all sentences of the institutions, i.e. if M and N are isomorphic (denoted $M \cong N$) then for each sentence ρ we have that $M \models \rho$ if and only if $N \models \rho$. The high level of abstraction of Dfn. 2.1 allows examples in which model isomorphisms do not preserve satisfaction; however these have a rather artificial nature. This assumption holds in most concrete examples of interest for specification and programming, such as the ones we present thereafter in Sect. 2.3.

[1] Strictly speaking, this is only a quasi-category living in a higher set-theoretic universe.

2.3. Examples of institutions

Many examples of logics, coming both from logic and computing, are captured as institutions, see [14] for some of them. In fact the thesis underlying institution theory is that anything that deserves to be called logic can be captured as institution. In the following we recall five of them that will be used all over the paper for reflecting our abstract developments at a concrete level.

Example 2.1 (Many-sorted algebra). Let MSA denote the institution of *many-sorted algebra*. This is perhaps the most notorious logical system in computing science, and it is also the original framework for structural induction.

Its *signatures* are pairs (S, F) consisting of

- a set of sort symbols S, and
- a family $F = \{F_{w\to s} \mid w \in S^*, s \in S\}$ of sets of function symbols indexed by arities (for the arguments) and sorts (for the results).

Signature morphisms map the two components in a compatible way. This means that a signature morphism $\varphi\colon (S, F) \to (S', F')$ consists of

- a function $\varphi^{\mathrm{st}}\colon S \to S'$, and
- a family of functions $\varphi^{\mathrm{op}} = \{\varphi^{\mathrm{op}}_{w\to s}\colon F_{w\to s} \to F'_{\varphi^{\mathrm{st}}(w)\to\varphi^{\mathrm{st}}(s)} \mid w \in S^*, s \in S\}$.

Models M for a signature (S, F), called (S, F)-*algebras*, interpret each sort symbol s as a set M_s, and each function symbol σ as a function M_σ from the product of the interpretations of the argument sorts to the interpretation of the result sort. In order to avoid the existence of empty interpretations of the sorts, which may complicate unnecessarily our presentation, we assume that each signature has at least one *constant* (i.e. function symbol with empty arity) for each sort. An homomorphism of (S, F)-algebras, called (S, F)-homomorphism and denoted $h\colon M \to M'$, is an indexed family of functions $\{h_s\colon M_s \to M'_s\}_{s\in S}$ such that for each $\sigma \in F_{w\to s}$ and each $m \in M_w$,

$$h_s(M_\sigma(m)) = M'_\sigma(h_w(m))$$

where $h_w\colon M_w \to M'_w$ is the canonical component-wise extension of h, i.e. $h_w(m_1, \ldots, m_n) = (h_{s_1}(m_1), \ldots, h_{s_n}(m_n))$ for $w = s_1 \ldots s_n$ and $m_i \in M_{s_i}$.

For each signature morphism φ, the *reduct* $M'{\restriction_\varphi}$ of an (S', F')-algebra M' is defined by $(M'{\restriction_\varphi})_x = M'_{\varphi(x)}$ for each sort or function symbol x from the domain signature of φ.

The many-sorted set of (S, F)-*terms* is denoted $T_{(S,F)}$. This is canonically endowed with an (S, F)-algebra structure, denoted $0_{(S,F)}$, which in fact is the *initial* (S, F)-algebra.

Sentences are the usual first-order sentences built from equational atoms of the form $t = t'$, for t and t' being (S, F)-terms of the same sort, by iterative application of Boolean connectives and quantifiers. Sentence translations along signature morphisms just rename the sorts, function, and relation symbols according to the respective signature morphisms. They can be formally defined by recursion on the structure of the sentences. While the recursion step is straightforward for the case of the Boolean connectives it needs a bit of attention for the case of the quantifiers. For any signature morphism $\varphi\colon (S, F) \to (S', F')$,

$$\mathsf{Sen}^{MSA}(\varphi)((\forall X)\rho) = (\forall X^\varphi)\mathsf{Sen}^{MSA}(\varphi')(\rho)$$

for each finite set X of *variables for* (S, F). The variables need to be disjoint from the constants of the signature, also we have to ensure that Sen^{MSA} thus defined is indeed functorial and that there is no overloading

of variables (which in certain situations would cause a failure of the Satisfaction Condition). These may be formally achieved by considering that a variable for (S, F) is a triple of the form $(x, s, (S, F))$ where x is the *name of the variable* and $s \in S$ is the *sort of the variable* and that two different variables in X have different names. Then we let $(S, F + X)$ be the extension of (S, F) such that $(F + X)_{w \to s} = F_{w \to s}$ when w is non-empty and $(F + X)_{\to s} = F_{\to s} \cup \{(x, s, (S, F)) \mid (x, s, (S, F)) \in X\}$ and we let $\varphi' \colon (S, F + X) \to (S', F' + X^{\varphi})$ be the canonical extension of φ that maps each variable $(x, s, (S, F))$ to $(x, \varphi(s), (S', F'))$. When there is no danger of confusion, for variables we may use the short notation x instead of the full notation $(x, s, (S, F))$.

The satisfaction of sentences by models is the usual Tarskian satisfaction defined recursively on the structure of the sentences:

- $M \models_{(S,F)} t = t'$ if and only if $M_t = M_{t'}$, where M_t denotes the evaluation of a term t in the algebra M which may be defined recursively on the structure of t by $M_{\sigma(t_1,\ldots,t_n)} = M_\sigma(M_{t_1}, \ldots, M_{t_n})$.
- $M \models_{(S,F)} \rho_1 \wedge \rho_2$ if and only if $M \models_{(S,F)} \rho_1$ and $M \models_{(S,F)} \rho_2$, and similary for the other Boolean connectives $\vee$, $\Rightarrow$, $\neg$, etc.
- $M \models_{(S,F)} (\forall X)\rho$ if and only if $M' \models_{(S,F+X)} \rho$ for each $(S, F + X)$-expansion M' of M, and similarly for the existential quantifications.

Example 2.2 (Preordered algebra). Preordered algebras are used for formal specification and verifications of algorithms [16], for automatic generation of case analysis [16], and in general about reasoning about transitions between states of systems. They constitute an unlabeled form of rewriting logic of [30]. Let POA denote the institution of preordered algebras.

The POA signatures are just the MSA signatures. The POA models are *preordered algebras* which are interpretations of the signatures into the category of preorders $\mathbb{P}re$ rather than the category of sets $\mathbb{S}et$. This means that each sort gets interpreted as a preorder, and each function symbol as a monotonic function. A *preordered algebra homomorphism* is just a family of monotonic functions which is an algebra homomorphism.

The initial (S, F)-algebra $0_{(S,F)}$ is also initial in the category of preordered (S, F)-algebras when the preorder relations on $0_{(S,F)}$ are the discrete ones, i.e. $t \leq t'$ if and only if $t = t'$.

The sentences have two kinds of atoms: equations $t = t'$ like in MSA and preorder atoms $t \leq t'$. A preorder atom $t \leq t'$ is satisfied by a preordered model M when the interpretations of the terms are in the preorder relation of the carrier, i.e. $M_t \leq M_{t'}$. The sentences are formed like in MSA from these atoms by Boolean connectives and quantifications over variables.

Example 2.3 (Multiple-valued logic). This institution denoted MVL, of great tradition in non-classical logic [19, 28, 32] and logical basis for 'fuzzy' developments, generalizes ordinary logic based upon the two Boolean truth values, *true* and *false*, to larger sets of truth values that are structured by the concept of *residuated lattices*. A *residuated lattice* [18, 25, 38] L is a bounded lattice (with $\leq$ denoting the underlying partial order that has infimum $\wedge$, supremum $\vee$, biggest $\top$ and lowest $\bot$ elements) and which comes equipped with an additional commutative and associative binary operation $\otimes$ which has $\top$ as identity and such that for all elements x, y and z

- $(x \otimes y) \leq (x \otimes z)$ if $y \leq z$, and
- there exists an element $x \Rightarrow z$ such that $y \leq (x \Rightarrow z)$ if and only if $x \otimes y \leq z$.

The first condition above just means that $x \otimes -$ is a functor on the partial order $(L, \leq)$, and the second condition means that it has a left adjoint $x \Rightarrow -$. The ordinary two-valued situation can be recovered when L is the two values Boolean algebra with $\otimes$ being the conjunction. Then $\Rightarrow$ is the ordinary Boolean implication. There is a myriad of interesting examples of residuated lattices used for multiple-valued logics for which $\otimes$ gets an interpretation rather different from the ordinary conjunction. One famous such example is the so-called *Łukasiewicz arithmetic conjunction* on the closed interval $[0, 1]$ defined by $x \otimes y = 1 - \min\{1, 2 - (x + y)\}$. In this example $x \Rightarrow y = \min\{1, 1 - x + y\}$.

Let us fix a residuated lattice L that is also complete, i.e. it has infimum and supremum for any sets of elements. *MVL signatures* are triples (S, F, P) such that (S, F) is an MSA signature and P is an S^*-indexed family of relation symbols, with P_w denoting the set of relation symbols of arity w. Signature morphisms map the three components in a compatible way, similar to the MSA signature morphisms.

The (S, F, P)-*sentences* are pairs (ρ, x) where ρ is a *pre-sentence* and x is any element of L. The (S, F, P)-pre-sentences are very much like the MSA sentences, but instead of equational atoms they are constructed from relational atoms $\pi(t_1, \ldots, t_n)$ (with t_i being terms of appropriate sorts) by the connectives $\bot, \top, \wedge, \vee, \otimes, \Rightarrow$ and by universal $(\forall X)$ and existential $(\exists X)$ quantifications for finite sets X of variables.

An MVL (S, F, P)-*model* M is an (S, F)-algebra together with an interpretation of each relation symbol $\pi \in P_w$ as an L-*fuzzy relation*, i.e. a function $M_\pi \colon M_w \to L$. A model homomorphism $h \colon M \to N$ is an (S, F)-algebra homomorphism such that $M_\pi(m) \leq N_\pi(h_w(m))$ for each $\pi \in P_w$ and each $m \in M_w$. It is easy to note that each MVL signature (S, F, P) has an initial model $0_{(S,F,P)}$ such that its underlying (S, F)-algebra is the initial (term) (S, F)-algebra and which interprets each relation symbol $\pi \in P_w$ by $(0_{(S,F,P)})_\pi(t_1, \ldots, t_n) = \bot$.

For each (S, F, P)-model M and each (S, F, P)-pre-sentence ρ we define a value $M[\rho]$ in L as follows:

- $M[\pi(t_1, \ldots, t_n)] = M_\pi(M_{t_1}, \ldots, M_{t_n})$ for relational atoms,
- $M[\rho_1 \wedge \rho_2] = M[\rho_1] \wedge M[\rho_2]$ and similarly for the other connectives $\vee$, $\otimes$ and $\Rightarrow$,
- $M[(\forall X)\rho] = \bigwedge\{M'[\rho] \mid M'\restriction_{(S,F,P)} = M\}$ and $M[(\exists X)\rho] = \bigvee\{M'[\rho] \mid M'\restriction_{(S,F,P)} = M\}$.

The translation of sentences and the model reducts along signature morphisms are defined like in MSA; we skip these details here. Then the MVL satisfaction relation is defined by

$$M \models^{MVL}_{(S,F,P)} (\rho, x) \text{ if and only if } x \leq M[\rho].$$

Example 2.4 (Many-sorted algebra with predefined types). This institution, denoted $MSA^{@}$, underlies specification and programming with predefined types in MSA. Its origins go back to [23] which gave a model theoretic semantics for predefined types within the context of the (many-sorted) equational logic programming paradigm and this idea has been gradually developed at the level of abstract institutions in [10, 11, 14].

An $MSA^{@}$ signature is a pair $((S, F), A)$ consisting of an MSA signature (S, F) and an (S, F)-algebra A. A signature morphism $(\varphi, h) \colon ((S, F), A) \to ((S', F'), A')$ consists of an MSA signature morphism $\varphi \colon (S, F) \to (S', F')$ and an (S, F)-algebra homomorphism $h \colon A \to A'\restriction_\varphi$.

The category of the $((S, F), A)$-models is the comma category $A/\mathsf{Mod}^{MSA}(S, F)$. This means that an $((S, F), A)$-model is an (S, F)-algebra homomorphism $m \colon A \to M$ and an $((S, F), A)$-homomorphism $h \colon (m \colon A \to M) \to (m' \colon A \to M')$ is a homomorphism $h \colon M \to M'$ such that $m; h = m'$. The category of the models of each signature $((S, F), A)$ has $1_A \colon A \to A$ as initial model.

The $((S,F),A)$-sentences are formed from equational atoms $a = a'$, with a and a' being elements of A of the same sort, by iterations of the usual Boolean connectives and of quantifications. The definition of $MSA^{@}$ quantifications requires a bit of work as follows.

We let (S,F_A) be the extension of (S,F) which adds the elements of A as new constants of corresponding sorts. For any set X of variables for (S,F) by $A[X]$ we denote the set of all normal forms in $T_{(S,F_A+X)}$ with respect to the rewrite system

$$E_A = \{\sigma(a) \to A_\sigma(a) \mid \sigma \in F_{w\to s},\ a \in A_w,\ w \in S^*,\ s \in S\}.$$

Then $A[X]$ can be endowed canonically with an $(S, F+X)$-algebra structure by defining

$$A[X]_\sigma(t_1,\ldots,t_n) = \mathsf{nf}(\sigma(t_1,\ldots,t_n))$$

where by $\mathsf{nf}(t)$ we denote the normal form of the term t. We extend the notation $A[X]$ also to this $(S, F+X)$-algebra. If ρ is an $((S,F+X),A[X])$-sentence then both $(\forall X)\rho$ and $(\exists X)\rho$ are $((S,F),A)$-sentences.

$MSA^{@}$ satisfaction is defined by recursion on the structure of the sentences as follows:

- $(m\colon A \to M) \models (a_1 = a_2)$ if and only if $m(a_1) = m(a_2)$.
- The satisfaction of the Boolean connectives is defined as in MSA, POA.
- $(m\colon A \to M) \models_{((S,F),A)} (\forall X)\rho$ if and only if $m' \models_{((S,F+X),A[X])} \rho$ for each $m'\colon A[X] \to M$ such that $i; m' = m$ where $i\colon A \to A[X]$ denotes the canonical inclusion.

Given an $MSA^{@}$ signature morphism $(\varphi,h)\colon ((S,F),A) \to ((S',F'),A')$ the corresponding reduct of an $((S',F'),A')$-model $m'\colon A' \to M'$ is defined as $h; m'{\restriction_\varphi}$. The translation of the sentences requires more elaborated work, however this follows the corresponding ideas from the MSA institution. In brief, each equation $a_1 = a_2$ gets translated to $h(a_1) = h(a_2)$, the translation preserves the Boolean connectives, and each quantified sentence $(\forall X)\rho$ gets translated to $(\forall X^\varphi)\mathsf{Sen}^{MSA^{@}}(\varphi', h')(\rho)$ where X^φ and φ' are like for the translations along MSA signature morphisms, and $h'\colon A[X] \to A'[X^\varphi]{\restriction_{\varphi'}}$ is the canonical extension of h.

Example 2.5 (Partial algebra). Here we consider the institution PA of partial algebra as employed by the specification language CASL [1].

A *PA signature* is a tuple (S, TF, PF), where TF is a family of sets of *total* function symbols and PF is a family of sets of *partial* function symbols such that $TF_{w\to s} \cap PF_{w\to s} = \emptyset$ for each arity w and each sort s. Signature morphisms map the three components in a compatible way.

A *partial algebra* is just like an MSA algebra, but interpreting the function symbols of PF as partial rather than total functions. A *partial algebra homomorphism* $h\colon A \to B$ is a family of (total) functions $\{h_s\colon A_s \to B_s\}_{s\in S}$ indexed by the set of sorts S of the signature such that $h_s(A_\sigma(a)) = B_\sigma(h_w(a))$ for each function symbol $\sigma \in TF_{w\to s} \cup PF_{w\to s}$ and each string of arguments $a \in A_w$ for which $A_\sigma(a)$ is defined.

The sentences have three kinds of atoms: *definedness* $\mathsf{def}(t)$, *strong* equality $t \overset{s}{=} t'$, and *existence* equality $t \overset{e}{=} t'$. For any set T of terms we let $\mathsf{def}(T)$ denote the set $\{\mathsf{def}(t) \mid t \in T\}$. The definedness $\mathsf{def}(t)$ of a term t holds in a partial algebra A when the interpretation A_t of t is defined. The strong equality $t \overset{s}{=} t'$ holds when the evaluations of both terms are undefined or both of them are defined and are equal. The existence equality $t \overset{e}{=} t'$ holds when the evaluations of both terms are defined and are equal[2]. The

[2]Note that $\mathsf{def}(t)$ is equivalent to $t \overset{e}{=} t$ and that $t \overset{s}{=} t'$ is equivalent to $(t \overset{e}{=} t') \vee (\neg\mathsf{def}(t) \wedge \neg\mathsf{def}(t'))$.

sentences are formed from these atoms by Boolean connectives and quantifications over total variables (i.e variables that are always defined). The satisfaction relation extends from atoms to all sentences in the usual Tarskian way, like in MSA, POA, etc. The translation of sentences and the model reducts along signature morphisms are defined like in MSA; we skip these details here.

In order to have a healthy theory of substitutions for partial algebras we need to refine PA to the institution of *partial algebra with definability constraints*, denoted PA', and defined in the following.

A PA' *signature* is a pair $((S, TF, PF), C)$ consisting of a PA signature (S, TF, PF) and a set of terms $C \subseteq T_{(S,TF+PF)}$ called *definability constraints*. A PA' signature morphism $\varphi\colon\ ((S, TF, PF), C) \to ((S', TF', PF'), C')$ is a PA signature morphism $\varphi\colon\ (S, TF, PF) \to (S', TF', PF')$ which preserves the definability constraints, i.e. $\mathsf{def}(C') \models \mathsf{def}(\varphi(C))$. The $((S, TF, PF), C)$-sentences are formed from atomic (S, TF, PF)-sentences by Boolean connectives and quantifications of the form $((\forall(X, C'))\rho$ or $((\exists(X, C'))\rho$ where X is a set of (total) variables for (S, TF, PF), C' is a set of $(S, TF + X, PF)$-terms such that $\mathsf{def}(C') \models \mathsf{def}(C)$, and ρ is any $((S, TF + X, PF), C')$-sentence. The $((S, TF, PF), C)$-models are the (S, TF, PF)-algebras A such that $A \models \mathsf{def}(C)$.

The satisfaction relation between $((S, TF, PF), C)$-models and $((S, TF, PF), C)$-sentences is defined by recursion on the structure of the sentence like in PA. Note that because of the PA' quantifications there is a sense in which one may say that PA' has 'more' sentences than PA.

Each PA' signature $((S, TF, PF), C)$ has an initial model $0_{((S,TF,PF),C)}$ defined as follows:

- its carrier consists of the least subset of $T_{(S,TF+PF)}$ containing all constants of TF and all terms and subterms of terms of C and which is closed under application of operation symbols from TF, and
- $(0_{((S,TF,PF),C)})_\sigma(t_1, \ldots, t_n) = \begin{cases} \sigma(t_1, \ldots, t_n) & \text{when } \sigma(t_1, \ldots, t_n) \text{ belongs to the carrier,} \\ \text{undefined} & \text{otherwise.} \end{cases}$

2.4. Model amalgamation

The crucial role of model amalgamation for the semantics studies of formal specifications comes up in very many works in the area, a few early examples being [17, 29, 34, 35]. The model amalgamation property is a necessary condition in many institution-independent model theoretic results (see [14]), thus being one of the most desirable properties for an institution. It can be considered even as more fundamental than the satisfaction condition since in institutions with quantifications it is used in one of its weak forms in the proof of the satisfaction condition at the induction step corresponding to quantifiers (see [14] for the details). Model amalgamation properties for institutions formalize the possibility of amalgamating models of different signatures when they are consistent on some kind of generalized 'intersection' of signatures.

Definition 2.2 (Model amalgamation). *A commutative square of signature morphisms*

$$\begin{array}{ccc} \Sigma & \xrightarrow{\varphi_1} & \Sigma_1 \\ {\scriptstyle\varphi_2}\downarrow & & \downarrow{\scriptstyle\theta_1} \\ \Sigma_2 & \xrightarrow[\theta_2]{} & \Sigma' \end{array}$$

is an amalgamation square *if and only if for each Σ_1-model M_1 and a Σ_2-model M_2 such that $M_1{\restriction_{\varphi_1}} = M_2{\restriction_{\varphi_2}}$, there exists a unique Σ'-model M', denoted $M_1 \otimes_{\varphi_1,\varphi_2} M_2$, or $M_1 \otimes M_2$ for short when there is no danger of ambiguity, such that $M'{\restriction_{\theta_1}} = M_1$ and $M'{\restriction_{\theta_2}} = M_2$. When we drop off the uniqueness requirement we call this a* weak model amalgamation square.

In most of the institutions formalizing conventional or non-conventional logics, pushout squares of signature morphisms are model amalgamation squares [14, 17].

Definition 2.3. *An institution* has (weak) model amalgamation *when each pushout square of signatures is a (weak) amalgamation square. A* semi-exact institution *is an institution with the model amalgamation property extended also to model homomorphisms.*

The literature considers also extensions of model amalgamation from pushouts to arbitrary co-limits, but these are not needed here.

Example 2.6. The categories of the signatures of all the five examples of institutions presented in this paper have pushouts. In all these cases the existence of pushouts may be obtained by using the general result on existence of pushouts in Grothendieck categories from [37]. A direct proof that MSA has small co-limits of signatures may be found in [14]. A special mention should be made for the case of $MSA^{@}$ where the existence of pushouts of signatures (in [2, 3] called 'free amalgamated products') is obtained by a double application of the above mentioned general result. The second such application involves also some non-trivial results about MSA models, such as existence of free constructions and that the category of MSA models for a given signature has pushouts. Both these results are well known in the algebraic specification literature, proofs may be found for example in [14]. Pushouts of signatures in $MSA^{@}$ may be also derived as an MSA instance of a general institution-independent result from [11], namely the existence of the so-called 'amalgamated sums'.

All the five examples of institutions presented in this paper are also semi-exact. For a proof of the semi-exactness of MSA the reader may consult [14]. The semi-exactness of MSA may be extended easily to POA. The work [8] proves the model amalgamation property for MVL, and this may be easily extended to semi-exactness. The institution $MSA^{@}$ is also semi-exact. Its model amalgamation property may be derived as an MSA instance of a general result from [9]. The proof that PA is semi-exact may follow the same way as the proof of the semi-exactness of MSA. Then the semi-exactness may be extended from PA to PA' by using a general institution-independent result that lifts model amalgamation and semi-exactness from signatures to theories (see [14]).

3. Abstract substitutions

In this section we develop an axiomatic approach to substitutions within abstract institutions, which is meant to support our institution-independent study of structural induction. The contents of the section are as follows.

1. We recall the institution-independent concepts of variables and of universal quantifications.
2. We recall the general institution-independent concept of substitution.
3. Our axiomatic development of systems of substitutions for structural induction consists first of a designation of a sub-class substitutions equipped with a function to the natural numbers giving the 'depth' of the substitutions, and then with a sub-designation of a class of so-called 'atomic' substitutions. The intention here is that each substitution used in structural induction should be presented as a finite composition of 'atomic' substitutions.

3.1. Variables and universal quantification in abstract institutions

The following terminology has been introduced in [13] and it is also used in [14].

Definition 3.1 (Variables). *For any signature Σ of an institution, a signature morphism $X : \Sigma \to \Sigma'$ is called a* Σ-variable. *Usually we will denote Σ', the target signature of X, by $\Sigma(X)$.*

We should be aware of the following slight terminological mismatch. As we will see in examples below, in actual situations a Σ-variable may in fact mean a set of actual variables for Σ. Moreover, in some cases, (e.g. PA'; see Ex. 3.4) the institution-independent variables in the sense of Dfn. 3.1 appear as pairs (X, C') with X a set of variables in the ordinary sense and C' a set of terms.

The following represents the standard approach to quantification at the level of abstract institutions [14]; it has been first introduced in [36].

Definition 3.2 (Universal quantification). *Given any Σ-variable X, any $\Sigma(X)$-sentence ρ and any Σ-model M we let $M \models (\forall X)\rho$ denote that $M' \models \rho$ for any X-expansion M' of M.*

We say that the institution has universal X-quantifications *when for each $\Sigma(X)$-sentence ρ there exists a Σ-sentence ρ' such that for each Σ-model M we have that $M \models (\forall X)\rho$ if and only if $M \models \rho'$.*

Example 3.1 (Variables and universal quantification in MSA, POA). Usual sets of variables for any MSA signatures are typical examples for Dfn. 3.1. Thus given a set X of variables for any MSA signature (S, F), we may overload X to denote the signature inclusion morphism $(S, F) \hookrightarrow (S, F + X)$. In the light of Dfn. 3.1, the signature $(S, F + X)$ may be denoted as $(S, F)(X)$. Note that MSA has universal X-quantification. Moreover it is not difficult to prove that MSA has universal X-quantification for any injective signature morphism X such that $\Sigma(X)$ adds only constants to the image of Σ through X.

The great generality of the institution-independent concepts of variable given by Dfn. 3.1 and universal quantification given by Dfn. 3.2 accomodate also other concepts of variables and quantifications corresponding to various extensions of MSA, such as infinite sets of first-order variables (by allowing X above, considered as set of variables, to be infinite) or sets of second-order variables (by considering signature extensions also with sort and operation symbols that are not constants).

This discussion about MSA variables and quantification is also valid as it is for POA.

Example 3.2 (Variables and universal quantification in MVL). The MVL concepts of variables and universal quantification, resp., arise as examples of Dfn. 3.1 and Dfn. 3.2, resp., as follows. Like in MSA and POA, for each set X of variables for a signature (S, F, P) we let X also denote the corresponding signature inclusion $(S, F, P) \hookrightarrow (S, F + X, P)$. That MVL has universal X-quantifications in the sense of Dfn. 3.2 follows from the easy result below.

Fact 3.1. *For each (S, F, P)-model M, each $(S, F + X, P)$-sentence ρ and each $k \in L$,*

$$M \models (\forall X)(\rho, k) \text{ if and only if } M \models ((\forall X)\rho, k).$$

Example 3.3 (Variables and universal quantification in $MSA^{@}$). Any set X of variables for an MSA signature (S, F) together with any (S, F)-algebra A may be regarded as an $((S, F), A)$-variable in $MSA^{@}$ (in the sense of Dfn. 3.1), namely the 'inclusion' $((S, F), A) \hookrightarrow ((S, F + X), A[X])$ in which the signature morphism component is the MSA signature inclusion $(S, F) \hookrightarrow (S, F+X)$ and the algebra homomorphism component is the canonical subalgebra homomorphism $A \hookrightarrow A[X]\restriction_{(S,F)}$.

If we denote the 'inclusion' $((S, F), A) \hookrightarrow ((S, F + X), A[X])$ also by X, then we may note that $MSA^{@}$ has universal X-quantifications (in the sense of Dfn. 3.2).

Example 3.4 (Variables and universal quantification in PA, PA'). The PA variables X for a signature (S, TF, PF) arise as an example of Dfn. 3.1 by considering the signature inclusion $(S, TF, PF) \hookrightarrow (S, TF + X, PF)$, also denoted by X. Then PA has universal X-quantifications in the sense of Dfn. 3.2.

The PA' variables (X, C') are captured as variables in the sense of Dfn. 3.1 by considering the PA' signature 'inclusions' $((S, TF, PF), C) \to ((S, TF + X, PF), C')$. If we denote the latter signature 'inclusion' also by (X, C') then we may note that PA' has (X, C')-quantifications in the sense of Dfn. 3.2.

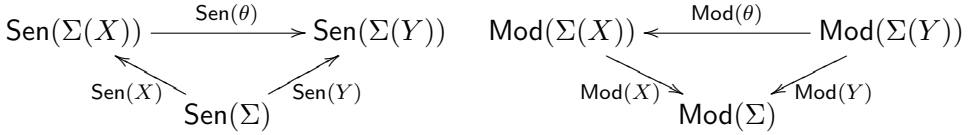
3.2. Substitutions in abstract institutions

The following general concept of substitution has been introduced in [13] and is also used in several places in [14].

Definition 3.3 (Substitution). *For any signature Σ of an institution, and any Σ-variables X and Y, a* Σ-substitution *θ from X to Y, denoted $\theta : X \dashrightarrow Y$, consists of a pair $(\mathsf{Sen}(\theta), \mathsf{Mod}(\theta))$, where*

- $\mathsf{Sen}(\theta) : \mathsf{Sen}(\Sigma(X)) \to \mathsf{Sen}(\Sigma(Y))$ *is a function, and*
- $\mathsf{Mod}(\theta) : \mathsf{Mod}(\Sigma(Y)) \to \mathsf{Mod}(\Sigma(X))$ *is a functor*

such that both of them preserve Σ, i.e., the following diagrams commute:



and such that for each $\Sigma(Y)$-model M'' and each $\Sigma(X)$-sentence ρ' the following satisfaction condition holds:

$$\mathsf{Mod}(\theta)(M'') \models \rho' \text{ if and only if } M'' \models \mathsf{Sen}(\theta)(\rho').$$

Like for signature morphisms, for any substitution θ we may denote $\mathsf{Mod}(\theta)(M'')$ by $M''|_\theta$ and $\mathsf{Sen}(\theta)(\rho')$ by $\theta(\rho')$.

Fact 3.2 (Composition of substitutions). *For any Σ-substitutions $\theta : X \dashrightarrow Y$ and $\psi : Y \dashrightarrow Z$ their* composition *$\theta; \psi : X \dashrightarrow Z$ defined by $\mathsf{Sen}(\theta; \psi) = \mathsf{Sen}(\theta); \mathsf{Sen}(\psi)$ and $\mathsf{Mod}(\theta; \psi) = \mathsf{Mod}(\psi); \mathsf{Mod}(\theta)$ is a Σ-substitution. Moreover this composition yields a category, called the* category of the Σ-substitutions*, in which the objects are the Σ-variables and the arrows are the Σ-substitutions.*

This categorical view of substitutions is similar in spirit to that of the works on abstract categorical unification of Goguen [20] and Burstall [33] in that a substitution (for a given fixed signature) is an arrow whose domain and target are 'objects of variables' which are not necessarily the same.

Example 3.5 (First-order MSA, POA substitutions). Let X and Y be sets of variables for an MSA signature (S, F). Any (many-sorted) function $\theta : X \to T_{(S,F+Y)}$ that preserves the sorts (i.e. $\theta(x, s, (S, F))$ is a term of sort s) determines an (S, F)-substitution (in the sense of Dfn. 3.3)

$$\theta^\sharp : (X : (S, F) \hookrightarrow (S, F + X)) \dashrightarrow (Y : (S, F) \hookrightarrow (S, F + Y))$$

defined for each $(S, F + Y)$-algebra M' and each $(S, F + X)$-sentence ρ by

- $\mathsf{Mod}(\theta^\sharp)(M')_z = \begin{cases} M'_z & \text{when } z \in S \text{ or } z \in F_{w\to s}, \\ M'_{\theta(z)} & \text{when } x \in X, \end{cases}$

- Informally, $\mathsf{Sen}(\theta^\sharp)(\rho)$ is the $(S, F+Y)$-sentence obtained by replacing all variables x from X in ρ by the term $\theta(x)$. This may be defined formally by recursion on the structure of ρ, however we skip this here.

The satisfaction condition for the substitution $\theta^\sharp$ follows by induction on the structure of ρ by using the fact that for each term t

$$\mathsf{Mod}(\theta^\sharp)(M')_t = M'_{\theta^\sharp(t)} \tag{2}$$

where $\theta^\sharp\colon T_{(S,F+X)} \to T_{(S,F+Y)}$ is the unique extension of θ to an (S,F)-homomorphism $0_{(S,F+X)} \to 0_{(S,F+Y)}$.

All these also constitute the example of first-order substitutions in POA, modulo the fact that the sentences are also formed from preorder atoms and the models are preordered algebras rather than (ordinary) algebras.

Example 3.6 (First-order substitutions in MVL). For any sets of variables X and Y for an MVL signature (S,F,P) each function $\theta\colon X \to T_{(S,F+Y)}$ that preserves the sorts determines a substitution

$$\theta\colon (X\colon (S,F,P) \hookrightarrow (S,F+X,P)) \dashrightarrow (Y\colon (S,F,P) \hookrightarrow (S,F+Y,P))$$

in a way very similar to Ex. 3.5 of first-order substitutions in MSA and POA. In this case the satisfaction condition is obtained from the fact that for any $(S,F+Y,P)$-model M' and for each $(S,F+X,P)$-pre-sentence ρ

$$\mathsf{Mod}(\theta^\sharp)(M')[\rho] = M'[\theta^\sharp(\rho)]$$

where by $\theta^\sharp(\rho)$ we denote the $(S,F+X,P)$-pre-sentence obtained by replacing each $x \in X$ by $\theta(x)$ in ρ. This is based upon the relation (2) of Ex. 3.5 which also holds in MVL.

Example 3.7 (First-order $MSA^@$ substitutions). This extends Ex. 3.5 from MSA to $MSA^@$ as follows. Let X and Y be sets of variables for an MSA signature (S,F) and let A be any (S,F)-algebra. Any many-sorted function (i.e. that preserves the sorts) $\theta\colon X \to A[Y]$ determines an $((S,F),A)$-substitution

$$\theta^\sharp\colon (X\colon ((S,F),A) \hookrightarrow ((S,F+X),A[X])) \dashrightarrow (Y\colon ((S,F),A) \hookrightarrow ((S,F+Y),A[Y]))$$

defined for each $((S,F+Y),A[Y])$-model $m'\colon A[Y] \to M'$ and each $((S,F+X),A[X])$-sentence ρ by

- $\mathsf{Mod}(\theta^\sharp)(m') = (m\colon A[X] \to M)$ where $M_z = \begin{cases} M'_z & \text{when } z \in S \text{ or } z \in F_{w\to s}, \\ m'(\theta(z)) & \text{when } x \in X. \end{cases}$
  and $m(t) = m'(A[\theta](t))$ where here $A[\theta]$ denotes the canonical extension of θ to an (S,F)-algebra homomorphism $A[X] \to A[Y]$.

- Informally, $\mathsf{Sen}(\theta^\sharp)(\rho)$ is the $(S,F+Y)$-sentence that extends the mapping $A[\theta]\colon A[X] \to A[Y]$ to sentences.

Example 3.8 (First-order PA' substitutions). Let (X, C') and (Y, C'') be variables for a PA' signature $((S, TF, PF), C)$. Any (many-sorted) function $\theta\colon\ X \to T_{(S,TF+PF+Y)}$ that preserves the sorts and such that $\mathsf{def}(C'') \models \mathsf{def}(\theta(X \cup C'))$ determines an $((S, TF, PF), C)$-substitution

$$\theta^\sharp\colon\ ((X, C')\colon\ ((S, TF, PF), C) \hookrightarrow ((S, TF + X, PF), C')) \dashrightarrow ((Y, C'')\colon\ ((S, TF, PF), C) \hookrightarrow ((S, TF + Y, PF$$

that is defined like in Ex. 3.5. Note that the definition of $\mathsf{Mod}(\theta^\sharp)(M')_x$ for $x \in X$ relies crucially upon the condition $\mathsf{def}(C'') \models \mathsf{def}(\theta(X))$. What happens here is that because x is total then $\mathsf{Mod}(\theta^\sharp)(M')_x$ should always be defined, which is guaranteed by the fact that $M'_{\theta(x)}$ is defined. In general this situation cannot be achieved in PA, and this is the reason behind refining it to the institution PA'.

The high generality of Dfn. 3.3 supports also examples that go beyond first-order substitutions, such as second-order substitutions. Since second-order substitutions will not constitute an example of our axiomatization of substitutions for structural examples, we avoid here the rather heavy technicalities of a fully general presentation of second order MSA substitutions (that can be read in [14]), and instead give a concrete example. Moreover, this idea may also be exported to other institutions, including those discussed in this paper.

Example 3.9 (Second-order MSA substitutions). Let (S, F) be an MSA signature such that S consists of a single sort, i.e. $S = \{s\}$, and F consists of one unary function symbol $f\colon\ s \to s$ and one constant $a\colon\ \to s$. Let X and Y, resp., be extensions of (S, F) with unary function symbols $x\colon\ s \to s$ and $y\colon\ s \to s$, resp.

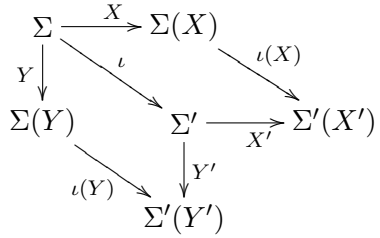
The canonical extension of the mapping $\theta\colon\ T_{(S,F+x)} \to T_{(S,F+y)}$ defined by

$$\theta(\sigma(t_1, \ldots, t_n)) = \begin{cases} \sigma(\theta(t_1), \ldots, \theta(t_n)) & \text{when } \sigma \text{ in } F \\ y(f(\theta(t_1), \ldots, \theta(t_n)) & \text{when } \sigma = x. \end{cases}$$

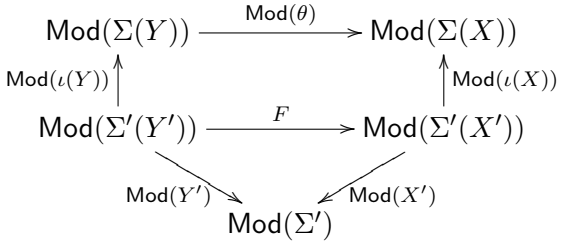
to a function $\mathsf{Sen}(S, F + x) \to \mathsf{Sen}(S, F + y)$ together with the functor $\mathsf{Mod}(S, F + y) \to \mathsf{Mod}(S, F + x)$ that maps any $(S, F + y)$-algebra A'' to an $(S, F + y)$-algebra A' such that both A' and A'' share the same reduct to (S, F) and such that $A'_x(z) = A''_y(A_f(z))$ yields an (S, F)-substitution (in the sense of Dfn. 3.3), which may be described in λ-notation as $\theta(x) = \lambda z.y(f(z))$.

The following result will only play a technical role below in the paper.

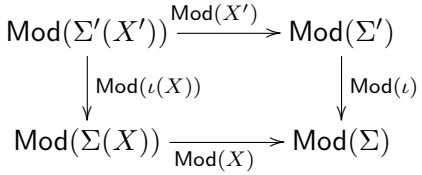
Proposition 3.1. *In any semi-exact institution, for any Σ-substitution $\theta\colon\ X \dashrightarrow Y$ and for any couple of signature pushouts as in the diagram below*



there exists a unique functor F making the diagram below commute



Proof. By the semi-exactness property of the institution we have that the following is a pullback in $\mathbb{C}at$.



Because $\iota; Y' = Y; \iota(Y)$ we have that

$$\mathsf{Mod}(\iota(Y)); \mathsf{Mod}(Y) = \mathsf{Mod}(Y'); \mathsf{Mod}(\iota). \tag{3}$$

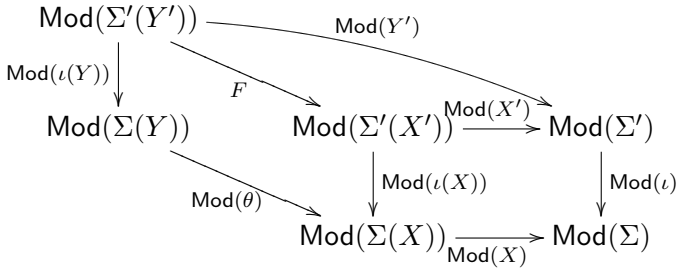
By the substitution condition for θ we have that

$$\mathsf{Mod}(Y) = \mathsf{Mod}(\theta); \mathsf{Mod}(X). \tag{4}$$

From (3) and (4) we have that

$$\mathsf{Mod}(\iota(Y)); \mathsf{Mod}(\theta); \mathsf{Mod}(X) = \mathsf{Mod}(Y'); \mathsf{Mod}(\iota). \tag{5}$$

From (5) by the pullback property there exists a unique functor F making the diagram below commute



□

3.3. Systems of substitutions

The basic Dfn. 3.3 recalled above together with the Dfns. 3.4, 3.5, and 3.6 introduced below constitute our axiomatic approach to institution-independent substitutions for structural induction.

Definition 3.4 (System of substitutions). *A* system of substitutions *in a given institution consists of a $|\mathbb{S}ig|$-indexed family $\mathcal{S} = \{\mathcal{S}_\Sigma \mid \Sigma \in |\mathbb{S}ig|\}$ such that for each $\Sigma \in |\mathbb{S}ig|$, $\mathcal{S}_\Sigma$ is a sub-category of the category of the Σ-substitutions (cf. Fact 3.2) and such that*

1. $1_\Sigma \in |\mathcal{S}_\Sigma|$,
2. *for each* $X \in |\mathcal{S}_\Sigma|$ *and any signature morphism* $\iota\colon\ \Sigma \to \Sigma'$ *there exists a pushout of signature morphisms*

$$\begin{array}{ccc} \Sigma & \xrightarrow{\ \iota\ } & \Sigma' \\ {\scriptstyle X}\downarrow & & \downarrow{\scriptstyle X'} \\ \Sigma(X) & \xrightarrow[\iota(X)]{} & \Sigma'(X') \end{array}$$

   such that $X' \in |\mathcal{S}_{\Sigma'}|$,
3. *for any* $X, Y \in |\mathcal{S}_\Sigma|$ *and any functor* F *making the diagram below commute*

$$\begin{array}{ccc} \mathsf{Mod}(\Sigma(Y)) & \xrightarrow{\ F\ } & \mathsf{Mod}(\Sigma(X)) \\ & {\scriptstyle \mathsf{Mod}(Y)}\searrow \quad \swarrow{\scriptstyle \mathsf{Mod}(X)} & \\ & \mathsf{Mod}(\Sigma) & \end{array}$$

   there exists a unique $\theta \in \mathcal{S}_\Sigma$ *such that* $F = \mathsf{Mod}(\theta)$.

The Σ*-substitutions that belong to* $\mathcal{S}_\Sigma$ *are called* $\mathcal{S}_\Sigma$-substitutions.

Example 3.10 (Systems of substitutions in MSA, POA, MVL). The standard system of first-order substitutions in MSA, denoted $\mathcal{S}^\omega$ is defined as follows:

- $|\mathcal{S}^\omega(S,F)| = \{X \mid X \text{ finite set of variables for } (S,F)\}$,
- $\mathcal{S}^\omega(S,F)(X,Y) = \{\theta^\sharp \mid \theta\colon\ X \to T_{(S,F+Y)}\}$, (see Ex. 3.5 and 3.6)
- for any (finite) set X of variables for (S,F) and any signature morphism $\iota\colon\ (S,F) \to (S',F')$ we let $X' = \{(x, \iota^{\mathrm{sort}}(s), (S',F')) \mid (x,s,(S,F)) \in X\}$; then $\iota(X)$ is the extension of ι that maps each $(x,s,(S,F))$ to $(x, \iota^{\mathrm{sort}}(s), (S',F'))$,
- for any functor $F\colon\ \mathsf{Mod}(S, F+Y) \to \mathsf{Mod}(S, F+X)$ such that $F(M')\restriction_{(S,F)} = M'\restriction_{(S,F)}$ for each $(S, F+Y)$-algebra M', we define $\theta\colon\ X \to T_{(S,F+Y)}$ by

$$\theta(x) = F(0_{(S,F+Y)})_x.$$

  Note that $\mathsf{Mod}(\theta^\sharp) = F$ and θ is unique with this property.

Another system of (first-order) substitutions in MSA extends $\mathcal{S}^\omega$ by allowing infinite sets of variables; let us denote this one by $\mathcal{S}^\infty$.

This example may be easily upgraded with only 'cosmetic' changes to POA and MVL by upgrading from MSA signatures, sentences and models, respectively, to POA and MVL signatures, sentences and models, respectively, and by reading $0_{(S,F+Y)}$ above as the initial preordered (S,F)-algebra in the case of POA and as the initial models $0_{(S,F,P)}$ in the case of MVL.

Example 3.11 (Systems of substitutions in $MSA^@$). The MSA system of substitutions presented in Ex. 3.10 may be extended to a system of substitutions in $MSA^@$ as follows:

- $|\mathcal{S}^\omega((S,F),A)| = |\mathcal{S}^\omega(S,F)|$ (of Ex. 3.10),

- $\mathcal{S}^{\omega}((S,F),A)(X,Y) = \{\theta^{\sharp} \mid \theta\colon\ X \to A[Y]\}$ (see Ex. 3.7),
- for any (finite) set X of variables for (S,F) and any $MSA^{@}$ signature morphism $(\iota,h)\colon\ ((S,F),A) \to ((S',F'),A')$ we let X' and $\iota(X)$ be defined like in Ex. 3.10 and we define $h[X]\colon\ A[X] \to A'[X']\restriction_{\iota(X)}$ by

$$h[X](\sigma(t_1,\ldots,t_n)) = \begin{cases} \iota(\sigma)(h[X](t_1),\ldots,h[X](t_n)) & \text{when } \sigma \text{ is in } F, \\ h(\sigma) & \text{when } \sigma \text{ is an element of } A, \\ \iota(X)(\sigma) & \text{when } \sigma \text{ is in } X. \end{cases}$$

  Then $(\iota,h)(X)$ is $(\iota(X), h[X])$.
- for any functor $F\colon\ \mathsf{Mod}((S,F+Y),A[Y]) \to \mathsf{Mod}((S,F+X),A[X])$ such that the diagram below consisting of F and two reduct functors commutes

$$\begin{array}{ccc} \mathsf{Mod}((S,F+Y),A[Y]) & \xrightarrow{\quad F\quad} & \mathsf{Mod}((S,F+X),A[X]) \\ & \searrow \quad \swarrow & \\ & \mathsf{Mod}((S,F),A) & \end{array}$$

  we define $\theta\colon\ X \to A[Y]$ as the restriction of $F(1_{A[X]})\colon\ A[X] \to A[Y]$. It is easy to check that $\mathsf{Mod}(\theta^{\sharp}) = F$ and θ is unique with this property.

Example 3.12 (System of substitutions in PA'). The PA' system of substitutions presented in Ex. 3.8 may be extended to a system of substitutions in PA' as follows:

- The objects of the category $\mathcal{S}^{\omega}((S,TF,PF),C)$ are the pairs (X,C') such that X is a finite set of total variables for (S,TF,PF) and $C' \subseteq T_{(S,TF+PF+X)}$ and $\mathsf{def}(C') \models \mathsf{def}(C)$,
- The morphisms from (X,C') to (Y,C'') are the mappings $\{\theta^{\sharp} \mid \theta\colon\ X \to T_{(S,TF+PF+Y)}\}$ such that $\mathsf{def}(C'') \models \mathsf{def}(\theta(X \cup C'))$,
- for any $(X,C') \in |\mathcal{S}^{\omega}((S,TF,PF),C)|$ and any PA' signature morphism $\iota\colon\ ((S,TF,PF),C) \to ((S',TF',PF'),D)$ we let X' and $\iota(X)$ be defined like in Ex. 3.10 and we let $D' = D \cup \iota(X)(C')$; then under the notations of Dfn. 3.4 we define $(X,C')' = (X',D')$ and $\iota(X,C') = (\iota(X)\colon\ ((S,TF,PF),C') \to ((S',TF',PF'),D'))$.

$$\begin{array}{ccc} ((S,TF,PF),C) & \xrightarrow{\ \iota\ } & ((S',TF',PF'),D) \\ {\scriptstyle (X,C')}\downarrow & & \downarrow{\scriptstyle (X',D')} \\ ((S,TF,PF),C') & \xrightarrow[\iota(X)]{} & ((S',TF',PF'),D') \end{array}$$

- for any functor $F\colon\ \mathsf{Mod}((S,TF+Y,PF),C'') \to \mathsf{Mod}((S,TF+X,PF),C')$ such that the diagram below consisting of F and two reduct functors commutes

$$\begin{array}{ccc} \mathsf{Mod}((S,TF+Y,PF),C'') & \xrightarrow{\quad F\quad} & \mathsf{Mod}((S,TF+X,PF),C') \\ & \searrow \quad \swarrow & \\ & \mathsf{Mod}((S,TF,PF),C) & \end{array}$$

we define $\theta\colon\ X \to T_{(S,TF+PF+Y)}$ by $\theta(x) = F(0_{(S,TF+Y,PF),C''})_x$. That $\theta(x)$ thus defined is indeed an $(S, TF + PF + Y)$-term follows by the help of the diagram above. In order to establish that $\theta^\sharp$ is substitution we have to show that $\mathsf{def}(C'') \models \mathsf{def}(\theta(X \cup C'))$. For this we use of the fact that for each term $t' \in T_{(S,TF+PF+Y)}$ and each $M'' \in |\mathsf{Mod}((S, TF + Y, PF), C'')|$ we have that

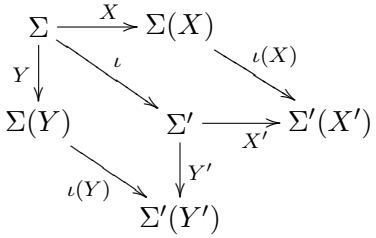
$$t' \in 0_{(S,TF+Y,PF),C''} \text{ implies } M'' \models \mathsf{def}(t')$$

which follows by applying the unique homomorphism $0_{(S,TF+Y,PF),C''} \to M''$ to t'. Therefore we are left with the task to show that $\theta(X \cup C') \subseteq 0_{(S,TF+Y,PF),C''}$. This follows because

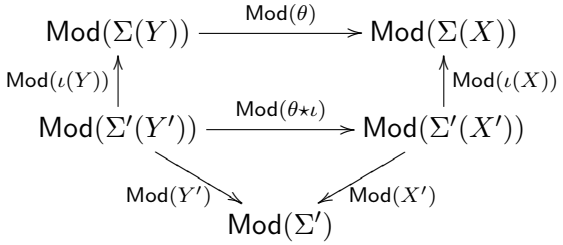
$$t \in 0_{(S,TF+X,PF),C'} \text{ implies } \theta(t) \in 0_{(S,TF+Y,PF),C''}$$

which can be proved by induction on the structure of t as follows. If $t = \sigma(t_1, \ldots, t_n) \in 0_{(S,TF+X,PF),C'}$ then when $\sigma = x \in X$ the conclusion follows by the definition of $\theta(x)$ and for the other cases since it follows that $t_1, \ldots, t_n \in 0_{(S,TF+X,PF),C'}$ we may apply the induction hypothesis.

Corollary 3.1 (Translation of substitutions along signature morphism). *In any semi-exact institution with a system of substitutions $\mathcal{S}$, for any $\mathcal{S}_\Sigma$-substitution $\theta\colon\ X \dashrightarrow Y$ and any couple of signature morphism pushouts as shown below*



such that $X', Y' \in |\mathcal{S}_{\Sigma'}|$ there exists a unique $\mathcal{S}_{\Sigma'}$-substitution $\theta \star \iota\colon\ X' \dashrightarrow Y'$ such that the diagram below commutes:



Proof. Directly from Prop. 3.1 and Dfn. 3.4. □

Example 3.13. For the *MSA* system of substitutions $\mathcal{S}^\omega$ of Ex. 3.5, for any $\theta\colon\ X \to T_{(S,F+Y)}$ and any morphism of signatures $\iota\colon\ (S, F) \to (S', F')$ the translation $\theta^\sharp \star \iota$ is the (S', F')-substitution $\psi^\sharp$ determined by $\psi\colon\ X' \to T_{(S',F'+Y')}$ where

- X' and Y' are defined like in Ex. 3.10, i.e. $X' = \{(x, \iota^{\mathrm{sort}}(s), (S', F')) \mid (x, s, (S, F)) \in X\}$ and similarly for Y', and
- for each $(x, s, (S, F)) \in X$, $\psi(x, \iota^{\mathrm{sort}}(s), (S', F'))$ is the $(S', F' + Y')$-term obtained by replacing in $\theta(x, s, (S, F))$ each symbol z of F by $\iota(z)$ and each $(y, s, (S, F)) \in Y$ by $(y, \iota^{\mathrm{sort}}(s), (S', F'))$.

The translations of POA, PA' and MVL substitutions are defined similarly.

For the $MSA^{@}$ system of substitutions $\mathcal{S}^{\omega}$ of Ex. 3.7, for any $\theta\colon X \to A[Y]$ and any morphism of signatures $\iota\colon ((S,F),A) \to ((S',F'),A')$ the translation $\theta^{\sharp} \star \iota$ is the $((S',F'),A')$-substitution $\psi^{\sharp}$ determined by $\psi\colon X' \to A[Y']$ where

- X' and Y' are defined like in the MSA example above, and
- for each $(x,s,(S,F)) \in X$, $\psi(x, \iota^{\mathrm{sort}}(s), (S',F'))$ is the $(S', F'_{A'} + Y')$-term obtained by replacing in $\theta(x,s,(S,F))$
  - each symbol z of F by $\iota(z)$,
  - each $a \in A$ by $h(a) \in A'$, and
  - each $(y,s,(S,F)) \in Y$ by $(y, \iota^{\mathrm{sort}}(s), (S',F'))$.

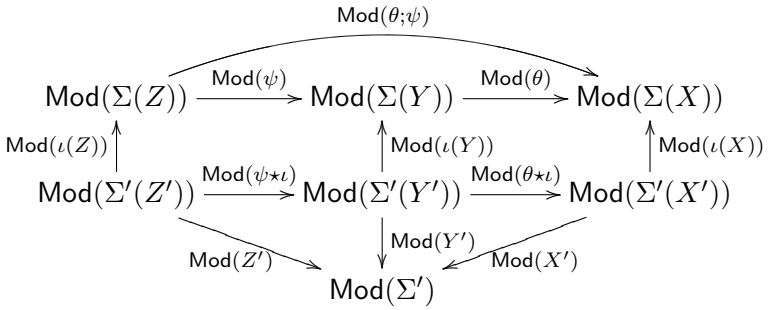
Corollary 3.2. *In any semi-exact institution with a system of substitutions $\mathcal{S}$, for any $\mathcal{S}_\Sigma$-substitutions $\theta\colon X \dashrightarrow Y$ and and $\psi\colon Y \dashrightarrow Z$ and any pushouts of signature morphisms as shown below*

$$\begin{array}{ccc} \Sigma & \xrightarrow{\iota} & \Sigma' \\ {\scriptstyle X}\downarrow & & \downarrow{\scriptstyle X'} \\ \Sigma(X) & \xrightarrow[\iota(X)]{} & \Sigma'(X') \end{array} \qquad \begin{array}{ccc} \Sigma & \xrightarrow{\iota} & \Sigma' \\ {\scriptstyle Y}\downarrow & & \downarrow{\scriptstyle Y'} \\ \Sigma(Y) & \xrightarrow[\iota(Y)]{} & \Sigma'(Y') \end{array} \qquad \begin{array}{ccc} \Sigma & \xrightarrow{\iota} & \Sigma' \\ {\scriptstyle Z}\downarrow & & \downarrow{\scriptstyle Z'} \\ \Sigma(Z) & \xrightarrow[\iota(Z)]{} & \Sigma'(Z') \end{array}$$

such that $X', Y', Z' \in |\mathcal{S}_{\Sigma'}|$ we have that

$$\mathsf{Mod}((\theta;\psi)\star\iota) = \mathsf{Mod}(\psi\star\iota);\mathsf{Mod}(\theta\star\iota).$$

Proof. Immediately from Cor. 3.1 by chasing the diagram below



□

Definition 3.5 (Substitutions with depth). *A* depth measure *d for a system $\mathcal{S}$ of substitutions in an institution is a family of functions from the substitutions to the set ω of the natural numbers, $d = \{d_\Sigma\colon \mathcal{S}_\Sigma \to \omega \mid \Sigma \in |\mathbb{S}ig|\}$, such that*

1. *$d(1_X) = 0$ for any $X \in |\mathcal{S}_\Sigma|$, and*
2. *for any $\theta\colon X \dashrightarrow Y$ and $\theta'\colon Y \dashrightarrow Z$ in $\mathcal{S}_\Sigma$ we have that $d(\theta;\theta') \leq d(\theta) + d(\theta')$.*

The substitutions θ with $d(\theta) = 0$ are called flat substitutions.

Example 3.14. In MSA, for the system of substitutions $\mathcal{S}^\omega$ of Ex. 3.10, we define a depth measure d as follows:

1. for each term $t = \sigma(t_1, \ldots, t_n)$ we define recursively a depth measure
$$d(t) = \begin{cases} 0 & \text{when } n = 0, \\ 1 + \max\{d(t_i) \mid 1 \leq i \leq n\} & \text{when } 0 < n. \end{cases}$$
2. for each $\theta : X \to T_{(S,F+Y)}$
$$d(\theta^\sharp) = \max\{d(\theta(x)) \mid x \in X\}.$$

Note that while this definition of depth measure works for $\mathcal{S}^\omega$ it does not work for $\mathcal{S}^\infty$, the finiteness of the sets of variables being crucial.

We may note immediately that this also functions as a depth measure when $\mathcal{S}^\omega$ is read as system of substitutions in POA. Based upon the fact that all $\mathcal{S}^\omega$-substitutions in MVL, $MSA^@$ and PA' defined above admit canonical representations as mappings between finite sets of variables and terms (although in the latter two cases not all such mappings do represent substitutions!) we may define similar depth measures for all these cases.

Definition 3.6 (Atomic substitutions). *In an institution, given a system of substitutions $\mathcal{S}$ with a depth measure d, a designated subclass $\mathsf{At}_\Sigma \subseteq \mathcal{S}_\Sigma$, for any signature Σ, is called a subclass of* atomic substitutions *when*

1. *any flat substitution $\theta : X \dashrightarrow 1_\Sigma$ is atomic,*
2. *for each non flat $\mathcal{S}_\Sigma$-substitution θ there are $\mathcal{S}_\Sigma$-substitutions Q and T such that $\theta = Q;T$, $Q \in \mathsf{At}_\Sigma$, and $d(T) < d(\theta)$.*

Example 3.15. In continuation of Ex. 3.14, we define the atomic substitutions of $\mathcal{S}^\omega$ as those substitutions $Q : X \dashrightarrow Y$ such that

1. $d(Q) \leq 1$,
2. $var(Q(x_1)) \cap var(Q(x_2)) = \emptyset$ for any $x_1 \neq x_2 \in X$, and
3. $Y = \bigcup\{var(Q(x)) \mid x \in X\}$,

where by $var(Q(z))$ we denote the subset of Y of the variables that actually occur in $Q(z)$.

For any non-flat $\mathcal{S}^\omega$-substitution $\theta : X \dashrightarrow Y$, for each $x \in X$, if $\theta(x) = \sigma(t_1, \ldots, t_n)$ such that $\sigma \notin Y$, we choose a set $Z_x = \{z_1, \ldots, z_n\}$ of variables such that the sort of z_k is the sort of t_k, for each $k \in \{1, \ldots, n\}$. Note that if $\theta(x)$ is a constant symbol (excluding variables) then $n = 0$ and consequently $Z_x = \emptyset$. If $\sigma \in Y$, then we let $Z_x = \{z_0\}$. We also choose the sets Z_x above such that $Z_{x_1} \cap Z_{x_2} = \emptyset$ when $x_1 \neq x_2$. We let $Z = \bigcup_{x \in X} Z_x$. We define the substitution $Q : X \dashrightarrow Z$ by

$$Q(x) = \begin{cases} \sigma(z_1, \ldots, z_n) & \text{when } \theta(x) = \sigma(t_1, \ldots, t_n),\ \sigma \notin Y \\ z_0 & \text{when } \sigma \in Y \text{ and } Z_x = \{z_0\} \end{cases}$$

and the substitution $T : Z \dashrightarrow Y$ by

$$T(z) = \begin{cases} t_k & \text{when } z = z_k \in Z_x,\ \theta(x) = \sigma(t_1, \ldots, t_n),\ \sigma \notin Y \\ \sigma & \text{when } z = z_0,\ Z_x = \{z_0\},\ \theta(x) = \sigma \in Y. \end{cases}$$

This definition of atomic substitutions is valid for all our benchmark examples MSA, POA, MVL, $MSA^@$, and PA'. However in the latter case, due to the specific nature of PA' variables and substitutions, we need some additional structure as follows.

In PA' the substitution θ has to be considered between (X, C') and (Y, C''). Then Z above should be rather defined as the pair $(Z, \mathsf{def}(Q(X \cup C'))$. While that Q satisfies the requirements for a substitution $(X, C') \dashrightarrow (Z, \mathsf{def}(Q(X \cup C'))$ is rather obvious, it is slightly less so for T as substitution $(Z, \mathsf{def}(Q(X \cup C')) \dashrightarrow (Y, C'')$. For this we have to show that

$$\mathsf{def}(C'') \models \mathsf{def}(T(Z \cup Q(X \cup C'))) = \mathsf{def}(T(Z)) \cup \mathsf{def}(\theta(X \cup C'))$$

The above relation follows from the fact that $\theta\colon (X, C') \dashrightarrow (Y, C'')$ and because $\mathsf{def}(\theta(X)) \models \mathsf{def}(T(Z))$, the latter relation being a direct consequence of the fact that for each term $\sigma(t_1, \ldots, t_n)$ we have that $\mathsf{def}(\sigma(t_1, \ldots, t_n)) \models \mathsf{def}(t_k)$ for each $k \in \{1, \ldots, n\}$.

4. Structural induction in abstract institutions

This section is devoted to the core result of our work, namely the institution-independent structural induction theorem. Its applicability is illustrated in the second part of the section by a series of actual instances.

Theorem 4.1 (Structural induction). *Let us consider a semi-exact institution with pushouts of signatures, equipped with:*

- *a system of substitutions $\mathcal{S}$,*
- *a depth measure d for $\mathcal{S}$,*
- *a system of atomic substitutions At for $\mathcal{S}$ and d, and*
- *a binary relation $\sqsubset$ on each set $\mathsf{At}(X, Y)$ such that*

$$\psi \sqsubset Q \text{ implies } \psi \text{ is flat and } Q \text{ is not flat.}$$

Let $\iota\colon \Omega \to \Sigma$ be a signature morphism, let $X \in |\mathcal{S}_\Omega|$ and let $X' \in |\mathcal{S}_\Sigma|$ be defined by the following pushout square:

$$\begin{array}{ccc} \Omega & \xrightarrow{\iota} & \Sigma \\ {\scriptstyle X}\downarrow & & \downarrow{\scriptstyle X'} \\ \Omega(X) & \xrightarrow[\iota(X)]{} & \Sigma(X') \end{array}$$

Let Γ be a set of Σ-sentences and ρ be a $\Sigma(X')$-sentence such that, for every atomic $\mathcal{S}_\Omega$-substitution $Q\colon X \dashrightarrow Z$ and every pushout square:

$$\begin{array}{ccc} \Omega & \xrightarrow{\iota} & \Sigma \\ {\scriptstyle Z}\downarrow & & \downarrow{\scriptstyle Z'} \\ \Omega(Z) & \xrightarrow[\iota(Z)]{} & \Sigma(Z') \end{array}$$

with $Z' \in |\mathcal{S}_\Sigma|$ we have:

$$Z'(\Gamma) \cup \{(\psi \star \iota)(\rho) \mid \psi \sqsubset Q\} \models_{\Sigma(Z')} (Q \star \iota)(\rho).$$

Then for all $\mathcal{S}_\Omega$-substitutions $\theta\colon X \dashrightarrow 1_\Omega$:

$$\Gamma \models_\Sigma (\theta \star \iota)(\rho).$$

Proof. We prove the conclusion of the theorem by induction on $d(\theta)$.

Let us assume $d(\theta) = 0$. We take $Q = \theta$. By the defining conditions on $\sqsubset$ we have that $\{\psi \mid \psi \sqsubset Q\} = \emptyset$. Because $Q = \theta$ we also have that $Z = 1_\Omega$, hence without any loss of generality we may consider $Z' = 1_\Sigma$. Thus, under this situation, the condition of the theorem reads as

$$\Gamma \models_\Sigma (\theta \star \iota)(\rho)$$

which represents the conclusion of the theorem.

At the induction step let us consider a $\mathcal{S}_\Omega$-substitution $\theta : X \dashrightarrow 1_\Omega$ such that $d(\theta) > 0$. Let $\theta = Q; T$ such that Q is atomic, $T : Z \dashrightarrow 1_\Omega$, and such that $d(T) < d(\theta)$. By the hypothesis we have that

$$Z'(\Gamma) \cup \{(\psi \star \iota)(\rho) \mid \psi \sqsubset Q\} \models_{\Sigma(Z')} (Q \star \iota)(\rho). \tag{6}$$

Let M be any Σ-model such that $M \models_\Sigma \Gamma \cup \{((\psi; T) \star \iota)(\rho) \mid \psi \sqsubset Q\}$. Let $M{\restriction_\iota}{\restriction_T} \otimes M$ be the $\Sigma(Z')$-model which is the amalgamation between the $\Omega(Z)$-model $M{\restriction_\iota}{\restriction_T}$ and M. From the substitution condition for $(T \star \iota) : Z' \dashrightarrow 1_\Sigma$ we have that $\mathsf{Mod}(T \star \iota); \mathsf{Mod}(Z') = 1_{\mathsf{Mod}(\Sigma)}$. This implies

$$M{\restriction_{T \star \iota}}{\restriction_{Z'}} = M. \tag{7}$$

From Cor. 3.1 for the substitution $(T \star \iota) : Z' \dashrightarrow 1_\Sigma$ we have that $\mathsf{Mod}(T \star \iota); \mathsf{Mod}(\iota(Z)) = \mathsf{Mod}(\iota); \mathsf{Mod}(T)$. This implies

$$M{\restriction_{T \star \iota}}{\restriction_{\iota(Z)}} = M{\restriction_\iota}{\restriction_T}. \tag{8}$$

By the uniqueness of model amalgamation from (7) and (8) we obtain

$$M{\restriction_{T \star \iota}} = M{\restriction_\iota}{\restriction_T} \otimes M. \tag{9}$$

By Cor. 3.2 we have that $\mathsf{Mod}((\psi; T) \star \iota) = \mathsf{Mod}(T \star \iota); \mathsf{Mod}(\psi \star \iota)$, hence from (9) we obtain

$$(M{\restriction_\iota}{\restriction_T} \otimes M){\restriction_{\psi \star \iota}} = M{\restriction_{(\psi; T) \star \iota}}. \tag{10}$$

Because $M \models_\Sigma \Gamma$, by the satisfaction condition of the institution we obtain that

$$M{\restriction_\iota}{\restriction_T} \otimes M \models_{\Sigma(Z')} Z'(\Gamma). \tag{11}$$

By the choice of M we know that for each $\psi \sqsubset Q$ we have that $M \models_\Sigma ((\psi; T) \star \iota)(\rho)$. From (10), by the satisfaction condition for the substitutions $(\psi; T) \star \iota$ and $\psi \star \iota$ we obtain that

$$M{\restriction_\iota}{\restriction_T} \otimes M \models_{\Sigma(Z')} (\psi \star \iota)(\rho) \quad \text{for each } \psi \sqsubset Q. \tag{12}$$

From (11) and (12), by the hypothesis (6) we obtain that $M{\restriction_\iota}{\restriction_T} \otimes M \models_{\Sigma(Z')} (Q \star \iota)(\rho)$ and further by the satisfaction conditions for the substitutions $Q \star \iota$ and $(Q; T) \star \iota$, respectively, that $M \models_\Sigma ((Q; T) \star \iota)(\rho)$. Thus we may conclude with

$$\Gamma \cup \{((\psi; T) \star \iota)(\rho) \mid \psi \sqsubset Q\} \models_\Sigma ((Q; T) \star \iota)(\rho) = (\theta \star \iota)(\rho). \tag{13}$$

We have that $d(\psi; T) \leq d(\psi) + d(T) = d(T) < d(\theta)$. Thus we may now use the induction hypothesis to get that for each $\psi \sqsubset Q$:

$$\Gamma \models_\Sigma ((\psi; T) \star \iota)(\rho). \tag{14}$$

From (13) and (14) we obtain the desired conclusion $\Gamma \models_\Sigma (\theta \star \iota)(\rho)$. □

Let us make the following comments with respect to Thm. 4.1.

1. In the applications ι represents the so-called 'sub-signatures of constructors'. A general treatment at the level of abstract institutions of this rather well-established concept, followed by examples, is given in Sect. 5 below. Constructors have only a pure methodological role, namely that of reducing the complexity of the proof process, a 'smaller' Ω leading to a smaller number of substitutions Q and hence a smaller number of proof goals. If we disregarded this efficiency aspect, then we could very well do without constructors, a situation that corresponds to setting ι of Thm. 4.1 to the identity 1_Σ. In such a case, the statement of Thm. 4.1 gets simplified with $\Omega = \Sigma$, $\iota(X)$ and $\iota(Z)$ being identities, $X = X'$ and $Z = Z'$.
2. The parameter $\sqsubset$ represents the main heuristic aspect of Thm. 4.1 and in actual situations the setting of its value is a key factor in defining actual structural induction methodologies. In setting $\sqsubset$ one should consider that a smaller $\sqsubset$ means fewer hypotheses for the proof goals of the associated structural induction methodology, a situation that may result in severe difficulties in the proof process. On the other hand, it is crucial to ensure the finiteness of the proof process through the finiteness of the set $\{\psi \mid \psi \sqsubset Q\}$. In the concrete instances of Thm. 4.1 presented below in this section $\sqsubset$ is set in a rather uniform way, which means that at the abstract level of Thm. 4.1 at this moment the parameter $\sqsubset$ may be seen mostly as an axiomatization device. However it seems a promising subject of further research to come up with concrete values for $\sqsubset$ leading to concrete structural induction methodologies alternative to those presented below in this section.
3. A crucial aspect of Thm. 4.1 is that it is supposed to represent a *finitary* proof process. We have already discussed one of the conditions for this, namely the finiteness of $\{\psi \mid \psi \sqsubset Q\}$. The other condition is the finiteness of the number of the (atomic) substitutions Q (from the statement of the theorem), which in the actual cases may be guaranteed by the finiteness of the signatures and by the atomicity of the substitutions Q (see the examples below in this section). Note also that the latter finiteness condition should be considered modulo isomorphism classes of Z and of $\Sigma(Z')$.

The intention of the examples presented below in this section is both to clarify aspects that are treated abstractly in Thm. 4.1 and to illustrate the concrete methodological power of the mathematical result of Thm. 4.1.

Since in all examples below the conditions of Thm. 4.1 on pushouts of signatures and on semi-exactness are fullfilled through Ex. 2.6, below we will skip them.

Example 4.1. Let us instantiate Thm. 4.1 for the following setting of its parameters:

- the institution is MSA,
- the system of substitutions is $\mathcal{S}^\omega$ of Ex. 3.10,
- the depth measure d is that defined Ex. 3.14,
- the system of atomic substitutions is defined in Ex. 3.15,
- the relation $\sqsubset$ defined as follows: for any atomic $\mathcal{S}^\omega_{(S,F)}$-substitutions $\psi, Q\colon X \to T_{(S,F+Z)}$ we have that $\psi \sqsubset Q$ if and only if
  - Q is not flat, and
  - $\psi\colon X \to Z$, i.e. ψ is *function* between sets of variables, and $var(\psi(x)) \subseteq var(Q(x))$ for each $x \in X$,

  and

- the signature morphism ι is a sub-signature inclusion $(S, F^c) \subseteq (S, F)$.

This yields the following method for structural induction in MSA:

Corollary 4.1 (Structural induction in MSA). *Let X be a finite set of variables for a signature (S, F) and let ρ be any $(S, F + X)$-sentence. Let (S, F^c) be sub-signature of (S, F) (i.e. $F^c_{w\to s} \subseteq F_{w\to s}$ for all arities w and sorts s) and a set Γ of (S, F)-sentences.*

If for any sort preserving mapping $Q\colon X \to F^c$ (i.e. the sort of Q_x is the sort of x),

$$\Gamma \cup \{\psi(\rho) \mid \psi\colon X \to Z = \cup_{x\in X} Z_x \text{ with } \psi(x) \in Z_x\} \models_{(S,F+Z)} Q^\sharp(\rho) \tag{15}$$

where

- *Z_x are strings of variables for the arguments of Q_x such that $Z_{x1} \cap Z_{x2} = \emptyset$ for $x1 \neq x2 \in X$, and*
- *$Q^\sharp$ is the substitution $X \to T_{(S,F^c+Z)}$ defined by $Q^\sharp(x) = Q_x(Z_x)$,*

then

$$\Gamma \models_{(S,F)} \theta(\rho) \text{ for all substitutions } \theta\colon X \to T_{(S,F^c)}. \tag{16}$$

Typical applications of Cor. 4.1 require that (S, F^c) is a *sub-signature of constructors for* Γ. This concept will be discussed in Sect. 5 below both at an abstract level and at the level of concrete logic, and (as has been mentioned above) has a pure methodological role, namely that of reducing the complexity of the proof task since a smaller F^c determines fewer mappings Q, hence fewer proof goals. Note that the finiteness of the proof task may be guaranteed by the finiteness of F^c (which since X is finite implies a finite number of mappings Q). The fact that (S, F^c) and (S, F) share the same set of sorts has a double significance. On the one hand their (sets of) variables coincide, and on the other hand, as we will see in Sect. 5 below, we would be able to have a proof theoretic characterization for sub-signatures of constructors.

Another important aspect of Cor. 4.1 that makes the corresponding methodology practically viable is that due to the finiteness of the arities of the operations and of the finiteness of X, if Γ is finite then we always have only a finite set of premises for each of the proof goals (because there is only a finite number of functions $\psi\colon X \to Z$).

A similar structural induction method may be obtained for POA and MVL, resp., just by changing the above setting of the institution from MSA to POA and MVL, resp. In both cases, in practice the role of the signature morphisms ι of Thm. 4.1 is played by 'sub-signatures of constructors'; this concept will be clarified in Sect. 5. At this moment, for this, we may just consider *any* sub-signatures.

Corollary 4.2 (Structural induction in POA). *The same statement as Cor. 4.1, but read within the POA framework.*

Corollary 4.3 (Structural induction in MVL). *Statement similar to Cor. 4.1, read within the MVL framework and with the role of the ιs being played by signature inclusions of the form $(S, F^c, P) \subseteq (S, F, P)$.*

Example 4.2. An instance of Thm. 4.1 within $MSA^@$ may be obtained along the lines of Cor. 4.1 by considering in the role of ι signature 'inclusions' of the form

$$((S, F^c), A^c) \subseteq ((S, F), A)$$

where $(S, F^c) \subseteq (S, F)$ is an inclusion of MSA signatures and $A^c \subseteq A{\restriction}_{(S,F^c)}$ is a sub-algebra relation. This constitutes the basic format for the concept of 'sub-signature of constructors' for $MSA^{@}$ that will be discussed in Sect. 5.

Then within the framework of $MSA^{@}$ the mappings Q of Cor. 4.1 become mappings $Q\colon\ X \to F^c_{A^c}$ (where $F^c_{A^c}$ denotes the extension of F^c with the elements of A^c considered as new constants), and instead of the substitutions $\theta\colon\ X \to T_{(S,F^c)}$ (of relation (16)) we have to consider substitutions $\theta\colon\ X \to A^c$.

Corollary 4.4 (Structural induction in $MSA^{@}$). *Statement similar to Cor. 4.1, read within the $MSA^{@}$ framework under the upgrades discussed above.*

Note that Cor. 4.1 may appear as a special case of Cor. 4.4 when $A = 0_{(S,F)}$ and $A^c = 0_{(S,F^c)}$. For this we need to reduce the mappings $Q\colon\ X \to F^c_{0_{(S,F^c)}}$ to the mappings $Q\colon\ X \to F^c$, which is based upon the remark that both sets of mappings give rise to the same set of substitutions.

Example 4.3. An instance of Thm. 4.1 within PA' may be obtained along the lines of Cor. 4.1 by considering in the role of ι signature 'inclusions' of the form

$$((S, TF^c, PF^c), C^c) \subseteq ((S, TF, PF), C)$$

where $(S, TF^c, PF^c) \subseteq (S, TF, PF)$ is an inclusion of PA signatures and $\mathsf{def}(C) \models \mathsf{def}(C^c)$. This constitutes the basic format for the concept of 'sub-signature of constructors' for PA' that will be discussed in Sect. 5. In the role of X of Thm. 4.1 let us consider (X, C^c). Then with these settings we have that:

- X' of Thm. 4.1 is (X, C),
- the mappings Q of Cor. 4.1 are upgraded to mappings $Q\colon\ X \to TF^c + PF^c$,
- the relation (15) gets upgraded to

$$\Gamma \cup \{\psi(\rho) \mid \psi\colon\ X \to Z = \cup_{x \in X} Z_x \ \text{ with } \ \psi(x) \in Z_x\} \cup \mathsf{def}(C \cup Q^\sharp(X)) \ \models_{(S,TF+Z,PF)} \ Q^\sharp(\rho)$$

  and

- the relation (16) gets upgraded to

$$\Gamma \cup \mathsf{def}(C) \models_{(S,TF,PF)} \theta(\rho) \quad \text{for all } \ \theta\colon\ X \to T_{(S,TF^c+PF^c)} \ \text{ with } \ \mathsf{def}(C^c) \models \mathsf{def}(\theta(X)).$$

Corollary 4.5 (Structural induction in PA'). *Statement similar to Cor. 4.1, read within the PA' framework under the upgrades discussed above.*

Note that Cor. 4.1 appears as a special case of Cor. 4.5 when PF is empty.

5. From inductive properties to structural induction

In this section we provide an institution-independent study of the relation (1) (see the Introduction), which represents the justification for using the structural induction method of Thm. 4.1 for proving inductive properties. The section consists of two parts:

1. A general treatment of the concept of constructors at the level of abstract institutions.
2. The development of an institution-independent approach and proof of the relation (1) above.

5.1. Constructors

The concept of constructor as a methodological device for induction is rather well established in the literature, one of the most elegant (in our opinion) theoretical treatments of constructors being found in [24]. The definition below abstracts the classical many-sorted algebra concept of constructors to abstract institutions.

Definition 5.1 (Sub-signature of constructors). *In any institution, for any class $\mathcal{E}$ of model homomorphisms, a signature morphism $\iota\colon\ \Omega \to \Sigma$ is a* sub-signature of $\mathcal{E}$-constructors for a set Γ *of Σ-sentences when*

- Γ *has an initial model* 0_Γ*,*
- *the signature Ω has an initial model 0_Ω, and*
- *the unique Ω-homomorphism $(0_\Omega \to 0_\Gamma{\restriction_\iota})$ is in $\mathcal{E}$.*

We have already mentioned that having a sub-signature of constructors as small as possible leads to substantial reduction in the size of the inductive proof scores. However there is also the minimal approach to constructors that is illustrated by the following fact and which in actual examples corresponds to the situations when all elements of the signature are considered constructors.

Fact 5.1. *If Γ is a set of Σ-sentences having an initial model 0_Γ such that the unique homomorphism $(0_\Sigma \to 0_\Gamma) \in \mathcal{E}$ then 1_Σ is a sub-signature of $\mathcal{E}$-constructors for Γ.*

In the following we present examples of sub-signatures of constructors that are relevant in the applications.

Example 5.1 (Constructors in MSA, POA). Given an MSA signature (S, F) it is well known that each set Γ of conditional equations for (S, F) (i.e. sentences of the form $(\forall X)H \Rightarrow C$ with H finite conjunctions of equations and C single equation) has an initial model 0_Γ. Let us set $\mathcal{E}$ of Dfn. 5.1 to the class of all surjective signature morphisms.

Proposition 5.1. *In MSA, a sub-signature (S, F^c) of (S, F) (i.e. $F^c_{w\to s} \subseteq F_{w\to s}$ for all arities w and sorts s) is a* sub-signature of $\mathcal{E}$-constructors for Γ *if and only if for each (S, F)-term t there exists an (S, F^c)-term t' such that $\Gamma \models_{(S,F)} t = t'$.*

Proof. By noting that (S, F^c) is a sub-signature of $\mathcal{E}$-constructors for Γ if and only if for each $a \in 0_\Gamma$ there exists an (S, F^c)-term t such that $a = (0_\Gamma)_t$. □

Note that the alternative formulation for MSA constructors given by Prop. 5.1 has the advantage (towards the one of Dfn. 5.1) of being more general in that it does not rely upon existence of initial models, which means that Γ may not be restricted only to conditional equations.

The same situation of constructors may be replicated to POA as follows. We know (from [14], for example) that any set Γ of POA Horn sentences of the form $(\forall X)H \Rightarrow C$ where H is any conjunction of atoms (either transitions $t \leq t'$ or equations $t = t'$) and C is a single atom, has an initial model. Like in MSA, let $\mathcal{E}$ be the class of the surjective preordered algebra homomorphisms. Then we have a situation similar to that in MSA, the proposition below sharing the same proof with Prop. 5.1.

Proposition 5.2. *In POA, a sub-signature (S, F^c) of (S, F) is a* sub-signature of $\mathcal{E}$-constructors for Γ *if and only if for each (S, F)-term t there exists an (S, F^c)-term t' such that $\Gamma \models_{(S,F)} t = t'$.*

Example 5.2 (Constructors in MVL). In MVL any set Γ of sentences of the form $(\forall X)H \Rightarrow C$ where H is a quantifier-free sentence formed from atoms and the connectives $\wedge$, $\vee$, and $\otimes$ admits an initial model 0_Γ (see [15]). A standard choice for $\mathcal{E}$ in this case is the class of all surjective model homomorphisms. Unlike for MSA and POA, in general MVL does not support a proof-theoretic characterization of constructors in the style of Prop. 5.1. However we conjecture that if Γ contains the theory of (fuzzy) L-equalities $\approx$ (see [25]) then it may be possible to have such a characterization with the equality $t = t'$ replaced by $(\top, t \approx t')$.

Example 5.3 (Constructors in $MSA^@$). In $MSA^@$ any set Γ of conditional equations for a signature $((S, F), A)$ admits an initial model 0_Γ (which appears as a quotient (S, F)-homomorphism $A \to B$); this can be obtained by methods similar to those from MSA, i.e. by quotienting A through the congruence determined by Γ.

We let $\mathcal{E}$ be the class of the surjective model homomorphisms. Note that for any sub-signature $((S, F^c), A^c) \subseteq ((S, F), A)$ like in Ex. 4.4, the condition $0_\Omega \to 0_\Gamma\restriction_\iota \in \mathcal{E}$ of Dfn. 5.1 just means that the composition of homomorphisms $(A^c \subseteq A\restriction_{(S,F^c)}); (0_\Gamma\restriction_{(S,F^c)})$ is surjective. The proof-theoretic characterization of constructors given by Prop. 5.1 gets extended to $MSA^@$ constructors as follows (the rather straightforward proof is ommited).

Proposition 5.3. *In $MSA^@$ $((S, F^c), A^c) \subseteq ((S, F), A)$ is a sub-signature of $\mathcal{E}$ constructors for Γ if and only if for each $a \in A$ there exists $a' \in A^c$ such that $\Gamma \models_{((S,F),A)} a' = a$.*

Example 5.4 (Constructors in PA'). From the literature of partial algebras, e.g. [4], it is well known that each set Γ of *QE-equations* for a signature (S, TF, PF), i.e. sentences of the form $(\forall X)H \Rightarrow C$ where H is a finite conjunction of existence equations $t \stackrel{e}{=} t'$ and C is a single existence equation, admits an initial algebra 0_Γ. The unique (S, TF, PF)-homomorphism $0_{(S,TF,PF)} \to 0_\Gamma$ is an epimorphism, and in PA the epimorphisms may *not* be surjective in general. Recall that an *epimorphism* of partial algebras $f\colon A \to B$ is characterized by the fact that the closed sub-algebra generated by the image $f(A)$ is B. A sub-algebra B' of B is *closed* when for all $b'_1, \ldots, b'_n \in B'$ and σ in PF if $B_\sigma(b'_1, \ldots, b'_n)$ is defined then $B_\sigma(b'_1, \ldots, b'_n) \in B'$.

Let $\mathcal{E}$ denote the class of epimorphisms and $\mathcal{E}_1$ the class of the surjective homomorphisms. The following proof theoretic characterization of constructors in PA' extends Prop. 5.1 (the rather straightforward proof is omitted here).

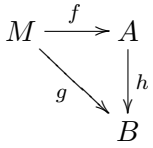
Proposition 5.4. *Let a set Γ of QE-equations for a PA'-signature $((S, TF, PF),\ C)$ and $((S, TF^c, PF^c),\ C^c)$ such that $(S, TF^c, PF^c) \subseteq (S, TF, PF)$ and $\mathsf{def}(C) \models \mathsf{def}(C^c)$. We have the following equivalences:*

- *The inclusion $((S, TF^c, PF^c),\ C^c) \subseteq ((S, TF, PF),\ C)$ is a sub-signature of $\mathcal{E}$-constructors for Γ if, and only if, for every term $t \in T_{(S,TF+PF)}$ such that $\Gamma \cup \mathsf{def}(C) \models \mathsf{def}(t)$, there exists a term $t' \in T_{(S,TF^c+PF^c)}$ such that $\Gamma \cup \mathsf{def}(C) \models t \stackrel{e}{=} t'$.*
- *The inclusion $((S, TF^c, PF^c),\ C^c) \subseteq ((S, TF, PF),\ C)$ is a sub-signature of $\mathcal{E}_1$-constructors for Γ if, and only if, for each term $t \in T_{(S,TF+PF)}$ such that $\Gamma \cup \mathsf{def}(C) \models \mathsf{def}(t)$, there exists a term $t' \in T_{(S,TF^c+PF^c)}$ such that $\mathsf{def}(C^c) \models \mathsf{def}(t')$ and $\Gamma \cup \mathsf{def}(C) \models t \stackrel{e}{=} t'$.*

5.2. Inductive satisfaction via ordinary deduction

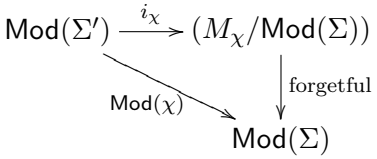
The following is a standard category theory concept (see [26, 31]).

Definition 5.2. *In any institution, given any class $\mathcal{E}$ of model homomorphisms, a model M is* $\mathcal{E}$-projective *when for each homomorphism $(h\colon A \to B) \in \mathcal{E}$ and each homomorphism $g\colon M \to B$ there exists a homomorphism $f\colon M \to A$ such that $f;h = g$.*



The following definition captures the essence of 'first-order' variables at the level of abstract institutions. It has been introduced in [12] and has subsequently been used quite a lot in institution-independent model theory studies (see [14]).

Definition 5.3 (Representable signature morphisms). *In any institution, a signature morphism $\chi\colon \Sigma \to \Sigma'$ is* representable *if and only if there exists a Σ-model M_χ (called the* representation of χ*) and an isomorphism i_χ of categories such that the following diagram commutes:*



(Recall that $M_\chi/\mathsf{Mod}(\Sigma)$ is the comma-category with Σ-homomorphisms $M_\chi \to M$ as objects and with Σ-homomorphisms $h\colon M \to N$ such that $f;h = g$ as arrows $(f\colon M_\chi \to M) \to (g\colon M_\chi \to N)$.)

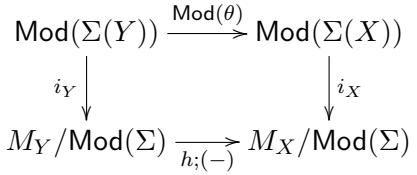
The nature of the representations M_χ may be understood better when recalling the following straightforward property.

Fact 5.2. *For any representable signature morphism $\chi\colon \Sigma \to \Sigma'$ we have that $\mathsf{Mod}(\Sigma')$ has an initial model $0_{\Sigma'}$ such that $M_\chi = 0_{\Sigma'}{\restriction_\chi}$.*

The following represents a slight upgrade of a corresponding definition from [13] and [14].

Definition 5.4 (Representable substitutions). *An institution with a system of substitutions $\mathcal{S}$* has representable $\mathcal{S}$-substitutions *when*

1. *each $\mathcal{S}$-variable $X \in |\mathcal{S}_\Sigma|$ is representable, and*
2. *for each $X, Y \in |\mathcal{S}_\Sigma|$ and $h\colon M_X \to M_Y$ there exists a $\mathcal{S}_\Sigma$-substitution $\theta\colon X \dashrightarrow Y$ such that the following diagram commutes:*



Example 5.5. MSA, POA, MVL, $MSA^{@}$, and PA' have representable $\mathcal{S}^{\omega}$-substitutions for the corresponding systems of susbtitutions $\mathcal{S}^{\omega}$. Fact 5.2 gives us the representations M_X for the Σ-variables $X \in |\mathcal{S}_\Sigma|$; in all these cases it is rather straighforward to check the property stated by Dfn. 5.3. The second condition of Dfn. 5.4 is also rather easy to check in the mentioned examples. For this we have only to note that each homomorphism $h\colon M_X \to M_Y$ determines a map $X \to |M_Y|$ (where here X is read as a set of variables and $|M_Y|$ denotes the underlying set of M_Y) and that the substitution determined by this map satisfies that $i_X(M'\restriction_\theta) = h; i_Y(M')$ for each $\Sigma(Y)$-model M'.

Proposition 5.5. *In any institution*

1. *with model amalgamation,*
2. *with a designated class $\mathcal{E}$ of model homomorphisms, and*
3. *with a system $\mathcal{S}$ of representable substitutions such that M_X is $\mathcal{E}$-projective for each $X \in |\mathcal{S}_\Sigma|$,*

let $\iota\colon \Omega \to \Sigma$ be a sub-signature of $\mathcal{E}$-constructors for a set Γ of Σ-sentences and let $X\colon \Omega \to \Omega(X) \in |\mathcal{S}_\Omega|$. Let E be a set of Σ-sentences such that $0_\Gamma \models E$. Then for any pushout square of signature morphisms such that $X' \in |\mathcal{S}_\Sigma|$

$$\begin{array}{ccc} \Omega & \xrightarrow{\iota} & \Sigma \\ {\scriptstyle X}\downarrow & & \downarrow{\scriptstyle X'} \\ \Omega(X) & \xrightarrow[\iota(X)]{} & \Sigma(X') \end{array}$$

if for some $\Sigma(X')$-sentence ρ we have that $\Gamma \cup E \models (\theta \star \iota)(\rho)$ for each $\mathcal{S}_\Omega$-substitution $\theta\colon X \dashrightarrow 1_\Omega$ then $0_\Gamma \models (\forall X')\rho$.

Proof. Let us assume the hypothesis of the proposition and let us consider any X'-expansion B of 0_Γ. We need to prove that $B \models \rho$. Since M_X is $\mathcal{E}$-projective there exists h such that the diagram below commutes:

$$\begin{array}{ccc} M_X & \xrightarrow{h} & 0_\Omega \\ & {\scriptstyle i_X(B\restriction_{\iota(X)})}\searrow & \downarrow \\ & & 0_\Gamma\restriction_\iota \end{array}$$

Let $\theta\colon X \dashrightarrow 1_\Omega$ be the $\mathcal{S}_\Omega$-substitution represented by $h\colon M_X \to 0_\Omega = M_{1_\Omega}$. We have that $(0_\Gamma\restriction_\iota)\restriction_\theta = B\restriction_{\iota(X)}$ which from Cor. 3.1 it implies

$$0_\Gamma\restriction_{\theta\star\iota}\restriction_{\iota(X)} = B\restriction_{\iota(X)}. \tag{17}$$

By the substitution condition on $\theta \star \iota$ we have

$$0_\Gamma\restriction_{\theta\star\iota}\restriction_{X'} = 0_\Gamma\restriction_1 = 0_\Gamma = B\restriction_{X'}. \tag{18}$$

By the uniqueness of model amalgamation in the institution, from (17) and (18) we obtain that $0_\Gamma\restriction_{\theta\star\iota} = B$. Then we have that $\Gamma \models (\theta \star \iota)(\rho)$ and $0_\Gamma \models E$ implies $0_\Gamma \models (\theta \star \iota)(\rho)$ and by the satisfaction condition of the substitution $\theta \star \iota$ we obtain that $B = 0_\Gamma\restriction_{\theta\star\iota} \models \rho$. □

In practice the set E of Prop. 5.5 above plays the role of 'lemmas', which means that Thm. 4.1 is applied in combination with Prop. 5.5 with $\Gamma \cup E$ in the role of Γ.

Example 5.6. Prop. 5.5 may be easily instantiated to our benchmark examples in MSA, POA, MVL, $MSA^{@}$, and PA' as follows:

- from Ex. 2.6, we know that all these institutions have model amalgamation,
- in all these cases $\mathcal{E}$ is set to be the class of the surjective model homomorphisms, and
- from Ex. 5.5 we know that $\mathcal{S}^{\omega}$ are systems of representable substitutions. The projectivity of M_X follows in all these cases from the surjectivity of the homomorphisms in $\mathcal{E}$ by the following straighforward result.

  Fact 5.3. *For any representable signature morphism $\chi : \Sigma \to \Sigma'$ the following are equivalent:*

  1. *M_χ is $\mathcal{E}$-projective.*
  2. *For each $(h : A \to B) \in \mathcal{E}$ and each χ-expansion B' of B there exists a χ-expansion $h' : A' \to B'$ of h.*

Note that this scheme may not work for PA' with $\mathcal{E}$ the class of epimorphisms (i.e. $\mathcal{E}_1$ in Ex. 5.4) because epimorphisms of partial algebras are not necessarily surjective. The difficulty of this case may also be understood if we noted that identities 1_Σ in general may not be sub-signatures of constructors for Γ set of Σ-QE-equations

6. Examples of structural induction proof scores

This section is devoted to some proof scores written in actual formal specification and verification languages that represent a direct implementation of our structural induction method and theory. The terminology 'proof score' is due to Joseph Goguen and designates script-like specifications of the proof structure of a formal verification process, including lemmas, conditions and proof tasks to be executed by the system. For this we use as languages CafeOBJ [16] and Maude [7]. Since both CafeOBJ and Maude notations are close enough to the ordinary mathematical notation we may skip here the introduction to these notations, which may be found in the corresponding literature.

6.1. An MSA structural induction proof score

Let us consider the following specification (written in CafeOBJ notation) of natural numbers with a semantic equality relation (the sort `Bool` and the constants `true` and `false` come from a data type of Booleans that is imported tacitly).

```
mod! PNAT {
  [ Nat ]
  op 0 :  -> Nat
  op s_ : Nat -> Nat
  op _=_ : Nat Nat -> Bool {comm}
  vars M N : Nat
  eq ((s M) = 0) = false .
  eq (0 = 0) = true .
  eq (s M = s N) = (M = N) .
}
```

The following defines a strict 'less than' relation on the natural numbers.

```
mod! PNAT< {
  protecting(PNAT)
  op _<_ : Nat Nat -> Bool
  vars M N : Nat
  eq 0 < s M = true .
  eq M < 0 = false .
  eq (s M < s N) = M < N .
}
```

Let us consider the following total order property:

$$(\forall \mathtt{M},\mathtt{N})\ \mathtt{(M < N) or (N < M) or (M = N)}. \tag{19}$$

We match this to the notations from our theory above as follows:

- Γ is the set of axioms of `PNAT<` (including the imports),
- X is the set of variables $\{\mathtt{M},\mathtt{N}\}$,
- ρ is `(M < N) or (N < M) or (M = N)`, and
- Ω (of Thm. 4.1) and (S, F^c) (of Cor. 4.1) is the sub-signature formed by `0`, `s`, `true`, and `false`; it is rather straightforward to show through Prop. 5.1 that this constitutes a sub-signature of constructors for Γ (i.e. `PNAT<`).

Then the denotation of `PNAT<` consists of the initial model 0_Γ.

Under the above matching of notations, that (19) is a property of `PNAT<` reads as $0_\Gamma \models (\forall X)\rho$. In order to prove this we first apply (the *MSA* instance of) Prop. 5.5 (with E set to $\emptyset$) and next Cor. 4.1. Under the notations of Cor. 4.1 there are four such mappings $Q\colon\ X \to F^c$. The proof scores for the corresponding four proof goals are given below; they consist of simple reductions by rewriting.

```
open PNAT< .
  ops m n :  -> Nat .
```

1. The case $Q_\mathtt{M} = \mathtt{0}$, $Q_\mathtt{N} = \mathtt{0}$ (then $Z_\mathtt{M} = Z_\mathtt{N} = \emptyset$, and no ψ):

   ```
   red (0 < 0) or (0 < 0) or (0 = 0) .
   ```

2. The case $Q_\mathtt{M} = \mathtt{0}$, $Q_\mathtt{N} = \mathtt{s}$ (then $Z_\mathtt{M} = \emptyset$, $Z_\mathtt{N} = \{\mathtt{n}\}$, and no ψ):

   ```
   red (0 < s n) or (s n < 0) or (0 = s n) .
   ```

3. The case $Q_\mathtt{M} = \mathtt{s}$, $Q_\mathtt{N} = \mathtt{0}$ (then $Z_\mathtt{M} = \{\mathtt{m}\}$, $Z_\mathtt{N} = \emptyset$, and no ψ):

   ```
   red (s m < 0) or (0 < s m) or (s m = 0) .
   ```

4. The case $Q_\mathtt{M} = \mathtt{s}$, $Q_\mathtt{N} = \mathtt{s}$ (then $Z_\mathtt{M} = \{\mathtt{m}\}$, $Z_\mathtt{N} = \{\mathtt{n}\}$, and one ψ given by $\psi(\mathtt{M}) = \mathtt{m}$ and $\psi(\mathtt{N}) = \mathtt{n}$):
   we first introduce the premise:

   ```
   eq (m < n) or (n < m) or (m = n) = true .
   ```

and next we give the goal:

```
red (s m < s n) or (s n < s m) or (s m = s n) .
close
```

Note that in this example we have performed simultaneous induction on two variables, a particular strength of our structural induction methodology that has been emphasized above at the theoretical level.

6.2. Another MSA structural induction

The following is a classical specification of the addition of natural numbers by recursion:

```
mod! PNAT+ {
  protecting(PNAT)
  op _+_ : PNat PNat -> PNat
  vars M N : PNat
  eq N + (s M) = s(N + M) .
  eq N + 0 = N .
}
```

Then Γ is the set of axioms of `PNAT+`, and $\Omega/(S, F^c)$ is the same sub-signature of constructors like in the previous example of *MSA* proof score, i.e. determined by `s`, `0`, `true`, and `false`. Let us consider the commutativity of the addition

$$(\forall \mathtt{M},\mathtt{N})\ \mathtt{M + N = N + M}. \tag{20}$$

We first apply Prop. 5.5 with E being the set of two sentences

$$E = \{(\forall \mathtt{N})\ \mathtt{0 + N = N},\ (\forall \mathtt{M},\mathtt{N})\ \mathtt{(s\ M) + N = s(M + N)}\}.$$

and next we apply Cor. 4.1 (with $\Gamma \cup E$ as results from Prop. 5.5 in the role of Γ). We also set X to $\{\mathtt{N}\}$ and consequently ρ to $(\forall \mathtt{M})\ \mathtt{M + N = N + M}$. Note that in principle there is also another choice, namely X set to $\{M, N\}$, however the first choice is appropriate here. The following is the CafeOBJ proof score:

```
open PNAT+
```

We introduce the lemmas of E:

```
  vars M' N' : PNat .
  eq 0 + N' = N' .
  eq (s M') + N' = s (M' + N') .
```

Since $X = \{\mathtt{N}\}$ we have only two mappings $Q\colon\ X \to F^c$ from the statement of Cor. 4.1, that give two proof goals.

1. The case $Q_{\mathtt{N}} = 0$ (then $Z = Z_{\mathtt{N}} = \emptyset$ and there is no ψ):

   ```
   op M : -> PNat .
   red M + 0 = 0 + M .
   ```

   Note that here we have transformed the proof goal $(\forall \mathtt{M})\ \mathtt{M + 0 = 0 + M}$ into a quantifier-free goal in the corresponding signature extended with `M` by using the well known so-called 'Generalization Rule' which in this particular case takes the following form

   $$\Gamma \cup E \models_{\Sigma}\ (\forall \mathtt{M})\ \mathtt{M + 0 = 0 + M} \quad \text{if and only if} \quad \Gamma \cup E \models_{\Sigma+\{\mathtt{M}\}}\ \mathtt{M + 0 = 0 + M}.$$

2. The case $Q_{\mathrm{N}} = \mathtt{s}$ (then $Z = Z_{\mathrm{N}} = \{\mathtt{n}\}$ and there is only one ψ defined by $\psi(\mathtt{N}) = \mathtt{n}$):
   We introduce the premise (since M' is variable, as declared above, this is the same as $(\forall \mathtt{M})\, \mathtt{M + n = n + M}$):

```
  op n : -> PNat .
  eq M' + n = n + M' .
```

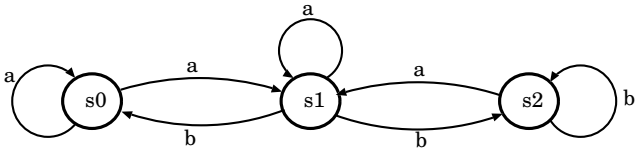
   This is the proof goal (again transformed to a quantifier-free sentence by the 'Generalization Rule'):

```
  red M + (s n) = (s n) + M .
close
```

The actual formal proof of (20) should be completed with proof scores for both sentences of E, following the same method of Prop. 5.5 and Cor. 4.1. We omit this here and leave it as exercise to the reader.

6.3. A POA structural induction proof score

Let us consider the following non-determinstic automata:



We use the Maude language for specifying this automata:

```
mod ND-AUT is
  sorts Letter Word State Config .
  subsort Letter < Word .
  ops a b : -> Letter .
  ops s0 s1 s2 : -> State .
  op nil : -> Word .
  op __ : Word Word -> Word [assoc] .
  op _*_ : State Word -> Config .
  var W : Word .
  eq nil W = W .
  rl s0 * a W => s1 * W .
  rl s0 * a W => s0 * W .
  rl s1 * a W => s1 * W .
  rl s1 * b W => s0 * W .
  rl s1 * b W => s2 * W .
  rl s2 * a W => s1 * W .
  rl s2 * b W => s2 * W .
endm
```

In the Maude notation => corresponds to $\leq$ in our POA notation. Also note in `ND-AUT` the rather mild involvment of the order sorted [22] extension of POA by the subsorting declaration `Letter < Word`. In the order sorted extension of POA the subsorts are interpreted as sub-preorders.

We let Γ denote the set of the axioms of `ND-AUT`, which consists of two universally quantified equations (i.e. the associativity of concatenation and the left identity for `nil`) and seven universally quantified

preorder atoms. The denotation of `ND-AUT` consists of 0_Γ, the initial preordered algebra satisfying Γ. This interprets the main sort, `Config`, as the preordered set of the pairs formed from the states `s0`, `s1`, or `s2` and words over the vocabulary $\{\mathtt{a},\mathtt{b}\}$ and whose preorder is generated by the seven transitions specified in `ND-AUT`.

In order to establish a sub-signature of constructors for Γ (denoted Ω in Thm. 4.1 and (S, F^c) in Cor. 4.2), we need to extend the signature of `ND-AUT` with a couple of derived operations having a pure notational role:

```
ops a._ b._ : Word -> Word .
eq a.W = a W .
eq b.W = b W .
```

Then through Prop. 5.2 we may prove that `s0`, `s1`, `s2`, `nil`, `a._`, and `b._` define a sub-signature of constructors for Γ. We omit this proof here.

Let us consider the following property for `ND-AUT`[3]:

$$(\forall \mathtt{W})\ ((\mathtt{s1} \star (\mathtt{W\ b\ nil}) \leq \mathtt{s2} \star \mathtt{nil}) \wedge (\mathtt{s2} \star (\mathtt{W\ b\ nil}) \leq \mathtt{s2} \star \mathtt{nil})). \tag{21}$$

In order to prove that (21) is satisfied by 0_Γ we first invoke the POA instance of Prop. 5.5 (see Ex. 5.6) with $E = \emptyset$ and then we use the structural induction method for POA given by Cor. 4.2. We have three mappings $Q\colon\ X = \{\mathtt{W}\} \to F^c$ and hence three proof goals as follows. The actual proofs use the Maude `search` command that has the following effect: whenever `search` t `=>*` t' gives `true` it implies that $t \leq t'$.

```
open ND-AUT .
  op w :  -> Word .
  var W : Word
  var S : State
```

1. The case $Q_{\mathtt{W}} = \mathtt{nil}$ (then $Z = Z_{\mathtt{W}} = \emptyset$, and no ψ):

   ```
   search s1 * nil b nil =>* s2 * nil .
   search s2 * nil b nil =>* s2 * nil .
   ```

2. The case $Q_{\mathtt{W}} = \mathtt{a.\_}$ (then $Z = Z_{\mathtt{W}} = \{\mathtt{w}\}$ and only one ψ defined by $\psi(\mathtt{W}) = \mathtt{w}$):
   We introduce the premise compactly specified as follows as Maude conditional transition:

   ```
   ctrans S * w b nil => s2 * nil if (S == s1) or (S == s2) .
   ```

   We prove the goal for this case:

   ```
   search s1 * a w b nil =>* s2 * nil .
   search s2 * a w b nil =>* s2 * nil .
   ```

3. The case $Q_{\mathtt{W}} = \mathtt{b.\_}$ is similar to the previous case, and sharing with it also the same premise, the only difference being in the proof goal:

   ```
   search s1 * b w b nil =>* s2 * nil .
   search s2 * b w b nil =>* s2 * nil .
   close
   ```

[3]This formalizes the fact that from the state resulting from any string of transitions ending with `b` applied to `s1` or `s2` one may reach `s2`.

7. Conclusion

We have developed a generic method for structural induction at the level of abstract institutions that may be instantiated to various actual induction proof methods in various logical systems. The main features of our development are

- an axiomatic approach to substitutions at the level of abstract institutions,
- in actual situations, the possibility of simultaneous induction on several variables,
- although relevant for proving properties of initial models, a proof method applicable in principle to any sets of axioms,
- an abstract generic treatment of constructors.

Our abstract developments have been illustrated with examples from various computing science logics and also with formal verification proof scores written in CafeOBJ and Maude languages.

Future research related to our work may include derivation of other concrete structural induction methodologies with applicability to formal verifications.

Acknowledgement. The author is grateful to the anonymous referee for his very careful reading of the paper and for his many detailed suggestions that have resulted in a better presentation of the material.

References

[1] Edigio Astesiano, Michel Bidoit, Hélène Kirchner, Berndt Krieg-Brückner, Peter Mosses, Don Sannella, and Andrzej Tarlecki. CASL: The common algebraic specification language. *Theoretical Computer Science*, 286(2):153–196, 2002.

[2] Franz Baader and Klaus U. Schultz. On the Combination of Symbolic Constraints, Solution Domains, and Constraint Solvers. In Ugo Montanari and Francesca Rossi, editors, *Principles and Practice of Constraint Programming*, volume 976 of *Lecture Notes in Computer Science*, pages 380–397, 1995.

[3] Franz Baader and Klaus U. Schulz. Combination of constraint solving techniques: An algebraic point of view. In Jieh Hsiang, editor, *Rewriting Techniques and Applications*, volume 914 of *Lecture Notes in Computer Science*, pages 352–366, 1995.

[4] Peter Burmeister. *A Model Theoretic Oriented Approach to Partial Algebras*. Akademie-Verlag, Berlin, 1986.

[5] Rod Burstall. Proving properties of programs by structural induction. *Computer Journal*, 12(1):41–48, 1969.

[6] Rod Burstall and Joseph Goguen. The semantics of Clear, a specification language. In Dines Bjorner, editor, *1979 Copenhagen Winter School on Abstract Software Specification*, volume 86 of *Lecture Notes in Computer Science*, pages 292–332. Springer, 1980.

[7] Manuel Clavel, Francisco Durán, Steven Eker, Patrick Lincoln, Narciso Martí-Oliet, José Meseguer, and Carolyn Talcott. *All About Maude - A High-Performance Logical Framework*, volume 4350 of *Lecture Notes in Computer Science*. Springer, 2007.

[8] Denisa Diaconescu. Model theory for Multiple-valued Logics. Master's thesis, Şcoala Normală Superioară Bucureşti, 2009.

[9] Răzvan Diaconescu. Interpolation for predefined types. *Mathematical Structures in Computer Science*. To appear.

[10] Răzvan Diaconescu. Category-based semantics for equational and constraint logic programming, 1994. DPhil thesis, University of Oxford.

[11] Răzvan Diaconescu. Category-based constraint logic. *Mathematical Structures in Computer Science*, 10(3):373–407, 2000.

[12] Răzvan Diaconescu. Institution-independent ultraproducts. *Fundamenta Informaticæ*, 55(3-4):321–348, 2003.

[13] Răzvan Diaconescu. Herbrand theorems in arbitrary institutions. *Information Processing Letters*, 90:29–37, 2004.

[14] Răzvan Diaconescu. *Institution-independent Model Theory*. Birkhäuser, 2008.

[15] Răzvan Diaconescu. On quasi-varieties of multiple valued logic models. *Mathematical Logic Quarterly*, 57(2):194–203, 2011.

[16] Răzvan Diaconescu and Kokichi Futatsugi. *CafeOBJ Report: The Language, Proof Techniques, and Methodologies for Object-Oriented Algebraic Specification*, volume 6 of *AMAST Series in Computing*. World Scientific, 1998.

[17] Răzvan Diaconescu, Joseph Goguen, and Petros Stefaneas. Logical support for modularisation. In Gerard Huet and Gordon Plotkin, editors, *Logical Environments*, pages 83–130. Cambridge, 1993. Proceedings of a Workshop held in Edinburgh, Scotland, May 1991.

[18] Nikolaos Galatos, Peter Jipsen, Tomasz Kowalski, and Hiroakira Ono. *Residuated Lattices: An Algebraic Glimpse at Substructural Logics*. Elsevier, 2007.

[19] Kurt Gödel. Zum intuitionistischen aussagenkalkül. *Anzeiger Akademie der Wissenschaften Wien*, Mathnaturwiss. Klasse 69:65–66, 1932.

[20] Joseph Goguen. What is unification? A categorical view of substitution, equation and solution. In Maurice Nivat and Hassan Aït-Kaci, editors, *Resolution of Equations in Algebraic Structures, Volume 1: Algebraic Techniques*, pages 217–261. Academic, 1989. Also Report SRI-CSL-88-2R2, SRI International, Computer Science Lab, August 1988.

[21] Joseph Goguen and Rod Burstall. Institutions: Abstract model theory for specification and programming. *Journal of the Association for Computing Machinery*, 39(1):95–146, 1992.

[22] Joseph Goguen and Răzvan Diaconescu. An Oxford survey of order sorted algebra. *Mathematical Structures in Computer Science*, 4(4):363–392, 1994.

[23] Joseph Goguen and José Meseguer. Models and equality for logical programming. In Hartmut Ehrig, Giorgio Levi, Robert Kowalski, and Ugo Montanari, editors, *Proceedings, TAPSOFT 1987*, volume 250 of *Lecture Notes in Computer Science*, pages 1–22. Springer, 1987.

[24] Joseph Goguen and José Meseguer. Order-sorted algebra solves the constructor selector, multiple representation and coercion problems. *Information and Computation*, 103, 1993.

[25] Petr Hájek. *Metamathematics of Fuzzy Logic*. Kluwer, 1998.

[26] Horst Herrlich and George Strecker. *Category Theory*. Allyn and Bacon, 1973.

[27] Saunders Mac Lane. *Categories for the Working Mathematician*. Springer, second edition, 1998.

[28] J. Łukasiewicz. Philosophische bemerkungen zu mehrwertigen systemen des aussagenkalküls. *Comptes Rendus Séances Société des Sciences et Lettres Varsovie*, (cl. III(23):51–77, 1930.

[29] José Meseguer. General logics. In H.-D. Ebbinghaus et al., editors, *Proceedings, Logic Colloquium, 1987*, pages 275–329. North-Holland, 1989.

[30] José Meseguer. Conditional rewriting logic as a unified model of concurrency. *Theoretical Computer Science*, 96(1):73–155, 1992.

[31] B. Mitchell. *Theory of categories*. Academic Press, 1965.

[32] E.L. Post. Introduction to a general theory of elementary propositions. *Amer. J. Math.*, 43:163–185, 1921.

[33] David Rydeheard and Rod Burstall. *Computational Category Theory*. Prentice-Hall, 1988.

[34] Donald Sannella and Andrzej Tarlecki. Specifications in an arbitrary institution. *Information and Control*, 76:165–210, 1988.

[35] Andrzej Tarlecki. On the existence of free models in abstract algebraic institutions. *Theoretical Computer Science*, 37:269–304, 1986.

[36] Andrzej Tarlecki. Quasi-varieties in abstract algebraic institutions. *Journal of Computer and System Sciences*, 33(3):333–360, 1986.

[37] Andrzej Tarlecki, Rod Burstall, and Joseph Goguen. Some fundamental algebraic tools for the semantics of computation, part 3: Indexed categories. *Theoretical Computer Science*, 91:239–264, 1991.

[38] Morgan Ward and Robert Dilworth. Residuated lattices. *Trans. Amer. Math. Soc.*, 45:335–354, 1939.