Quasi-Boolean Encodings and Conditionals in Algebraic Specification

Răzvan Diaconescu

Institute of Mathematics "Simion Stoilow" of the Romanian Academy

Abstract

We develop a general study of the algebraic specification practice, originating from the OBJ tradition, which encodes atomic sentences in logical specification languages as Boolean terms. This practice originally motivated by operational aspects, but also leading to significant increase in expressivity power, has recently become important within the context of some formal verification methodologies mainly because it allows the use of simple equational reasoning for frameworks based on logics that do not have an equational nature. Our development includes a generic rigorous definition of the logics underlying the above mentioned practice, based on the novel concept of 'quasi-Boolean encoding', a general result on existence of initial semantics for these logics, and presents a general method for employing Birkhoff calculus of conditional equations as a sound calculus for these logics. The high level of generality of our study means that the concepts are introduced and the results are obtained at the level of abstract institutions (in the sense of Goguen and Burstall [12]) and are therefore applicable to a multitude of logical systems and environments.

Key words: algebraic specification, institution theory, Boolean algebra, encodings, OBJ, CafeOBJ

1. Introduction

Equational logic, usually in many sorted form, is traditionally the logical basis for classical algebraic specification. The sentences, or the axioms, of algebraic specifications are often considered as *conditional* equations, i.e. first order sentences of the form $(\forall X)H \Rightarrow (t = t')$ where $t = t'$ is an equation, i.e. a formal equality of terms for a corresponding signature, and H is a finite conjunction of equations $(t_1 = t'_1) \wedge \ldots \wedge (t_n = t'_n)$. For example this is assumed to be the case for the famous pioneering language OBJ [17] or for the functional part of many of the modern algebraic specification languages, such as CafeOBJ [6, 8]. Conditional equational logics have a series of properties that make them rather suitable for formal specification. In particular they admit initial semantics, which is the main way to specify data types. They also have good computational properties, that provide a simple and smooth integration between the specification and the formal verification aspects of formal methods based upon conditional equational logic. Moreover, equational logic in conditional form provides the framework for the so-called 'equational logic programming' [14, 15], a rather powerful logic programming paradigm.

In some cases, including OBJ and CafeOBJ, the execution mechanism of conditional equational logic specifications by rewriting requires the following trick: the conditions H of the equations are considered as Boolean terms by encoding the syntactic equality = as an algebraic operation == of the Boolean sort, each equation $t_i = t'_i$ as a Boolean term $t_i == t'_i$, and the syntactic conjunction $\wedge$ as an algebraic operation on the Boolean sort. Moreover, within the multi-logic framework of CafeOBJ such encodings are also used for preordered algebra (the syntactic transition relation => gets encoded as the operation ==>) and hidden algebra for behavioural specification (the syntactic behavioural equivalence gets encoded as the operation =b=); details of these can be found in [6]. Thus, whilst in the OBJ case this practice of conditions as Boolean terms represents an encoding of equational logic into equational logic, in CafeOBJ it also means encodings of other logics into equational logic.

We argue that such encodings, far from being mere operational aspects, in reality lead to other underlying logics than the assumed ones for the respective languages. For example, the common specification practice in OBJ or in

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

Preprint submitted to Journal of Logic and Algebraic Programming *October 4, 2009*

CafeOBJ uses as conditions Boolean expressions that are more complex than finite conjunctions, corresponding to universally quantified first-order sentences that in general may not admit initial semantics. Although the specification power of conditions as Boolean terms is significantly increased by the use of Boolean operators other than conjunction, this poses several problems, such as whether initial semantics is still possible in such an extended context.

This paper develops an analysis of the logic of conditional sentences with conditions as Boolean terms in a generic way at the level of abstract institutions. The theory of institutions of [12] is a category theoretic form of abstract model theory that has gained a foundational status in algebraic specification theory especially in connection with the general developments of concepts and results that are independent of the details of particular logical systems. Such abstract generic developments have proved to be extremely useful for dealing with the population explosion in specification logics that took place over the last two decades. This is the case of our study too. Our general results can be instantiated to a multitude of base logics, captured as institutions, including those of OBJ and of CafeOBJ mentioned above. We illustrate this by developing explicitly applications to a series of concrete logics, including (many sorted) total algebra, predicate logic, preordered algebra, partial algebra, and hidden algebra.

The contents of our work is as follows:

1. We start with a rigorous definition of a generic logic of conditional sentences with conditions as Boolean terms, which is organized as an institution. This is based upon the definition of an encoding of abstract institutions into equational logic. An important aspect of these definitions is that instead of the conventional two-valued Boolean type they use a loose variant of Booleans that in a minimal format can be specified only as a sort with a truth constant. This corrects the current practice of using the standard tight semantics Boolean type for encoding conditions, which may have some serious gaps including inconsistency in the sense of impossibility to have models for the specification.
2. Next we develop a general result about the existence of initial semantics for the institutions of conditional sentences introduced in the previous section of the paper. This is obtained via abstract quasi-varieties of models.
3. The final technical part of this paper develops proof theoretic consequences of the encoding of abstract institutions into equational logic that underlies our work. We show how the standard Birkhoff calculus for conditional equations can be used as a sound calculus for a multitude of institutions of conditional sentences including logics that do not have an equational nature. For example, while this covers the current OBJ and CafeOBJ formal verification practice based upon equational reasoning, in the CafeOBJ case even within the context of non-equational logics, it can also be applied to many other situations, for example to partial algebra specifications.

One of the specific aspects of the encoding of institutions into equational logic studied here is the treatment of the Boolean connectors as algebraic operations. This allows the usage of conditions that correspond to Boolean expressions much beyond simple conjunctions of atoms, with all their specification power benefits, and yet admitting initial semantics and the use of the ordinary conditional equational proof calculus as a sound proof system, situations not enjoyed by the conventional (unencoded) treatment of the Boolean connectors.

The encoding of equations as Boolean terms via the encoding of the syntactic equality = as a Boolean valued operation == plays a central role in the recent so-called OTS/CafeOBJ verification method [10, 24]. The work reported in this paper may provide the necessary foundations for at least some aspects of the above mentioned verification method.

2. Preliminaries

In this section we introduce some institution theory concepts and present a series of examples of institutions that will be used to illustrate instances of the general developments of our paper.

Category theory.. We assume the reader is familiar with basic notions and standard notations from category theory; e.g., see [19] for an introduction to this subject. Here we recall very briefly some of them. By way of notation, $|\mathbb{C}|$ denotes the class of objects of a category $\mathbb{C}$, $\mathbb{C}(A, B)$ the set of arrows 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 $\mathbb{C}at$ is the category of all categories.[1] The opposite of a category $\mathbb{C}$ (obtained by reversing the arrows of $\mathbb{C}$) is denoted $\mathbb{C}^{\mathrm{op}}$. For the purpose of our work let us relax the concept of natural transformation as follows.

Definition 2.1 (Quasi-natural transformations). *Given functors $F, G\colon\ \mathbb{C} \to \mathbb{D}$ a quasi-natural transformation $\gamma\colon\ F \Rightarrow G$ consists of families of arrows $\{\gamma_\Sigma\colon\ F(\Sigma) \to G(\Sigma) \mid \Sigma \in |\mathbb{C}|\}$ and $\{\gamma_\varphi \mid \varphi \in \mathbb{C}\}$ such that for each arrow $\varphi\colon\ \Sigma \to \Sigma'$ in $\mathbb{C}$ we have that $\gamma_\varphi\colon\ F(\varphi);\gamma_{\Sigma'} \to \gamma_\Sigma; G(\varphi)$.*

Quasi-natural transformations are like the well established 2-categorical concept of *lax natural transformation* minus some compositionality conditions on γ_φ. Although all quasi-natural transformations in our examples are in fact lax natural transformations, we prefer to work with the former concept because it is technically enough and in the applications it has the advantage of having to check less conditions.

Institutions. Institutions have been defined by Goguen and Burstall in [2], the journal seminal paper [12] being printed after a delay of many years. Below we recall the concept of institution which formalises the intuitive notion of logical system, including syntax, semantics, and the satisfaction between them.

Definition 2.2 (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 $\mathsf{Sen}^{\mathcal{I}}\colon\ \mathbb{S}ig^{\mathcal{I}} \to \mathbb{S}et$, giving for each signature a set whose elements are called* sentences *over that signature,*
3. *a functor $\mathsf{Mod}^{\mathcal{I}}\colon\ (\mathbb{S}ig^{\mathcal{I}})^{\mathrm{op}} \to \mathbb{CAT}$ 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 $\varphi\colon\ \Sigma \to \Sigma'$ 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 may denote the *reduct* functor $\mathsf{Mod}^{\mathcal{I}}(\varphi)$ by $\_\!\upharpoonright_\varphi$ and the sentence translation $\mathsf{Sen}^{\mathcal{I}}(\varphi)$ by $\varphi(-)$. When $M = M'\!\upharpoonright_\varphi$ 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 $\mathbb{S}ig$. Also, when the signature is clear we may omit it as subscript of the satisfaction relation $\models$.
General assumption: We assume that all our abstract institutions are such that satisfaction is invariant under model isomorphism, i.e. if Σ-models M, M' are isomorphic, then $M \models_\Sigma \rho$ if and only if $M' \models_\Sigma \rho$ for all Σ-sentences ρ. This very basic assumption holds virtually for all concrete institutions of interest, including those discussed in our current paper.

Notation 2.1. *For E and E' sets of Σ-sentences in an arbitrary institution by $E \models_\Sigma E'$ we denote that for all Σ-models M, if $M \models_\Sigma E$ then $M \models_\Sigma E'$.*

There are myriads examples of institutions from logic or computing science (see [4] for some of these). The examples presented below will be used as concrete benchmarks for our general results.

Example 2.1 (Total algebra). This institution is denoted **ALG**. Its signatures are called *algebraic signatures*, which are pairs (S, F) consisting of a set of sort symbols S and of a family $F = \{F_{w\to s} \mid w \in S^*, s \in S\}$ of sets of function symbols indexed by strings of sort symbols, called 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

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

– a function $\varphi^{\mathrm{st}} : S \to S'$,

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

Given a signature (S,F), the models A of (S,F) are called (S,F)-*algebras* and they interpret each sort symbol s as a set A_s and each function symbol $\sigma \in F_{w\to s}$ as a function $A_\sigma : A_w \to A_s$ where by A_w we denote the cartesian product $A_{s_1}\times\ldots\times A_{s_n}$ where $w = s_1 \ldots s_n$. An *algebra homomorphism* $h : A \to A'$ is an indexed family of functions $(h_s : A_s \to A'_s)_{s\in S}$ such that $h_s(A_\sigma(a)) = A'_\sigma(h_w(a))$ for each $\sigma \in F_{w\to s}$ and each $a \in A_w$.[2] For any signature (S,F), the (S,F)-algebra homomorphisms compose component-wise as functions, and this yields the category $\mathsf{Mod}^{\mathbf{ALG}}(S,F)$. For any signature morphism $\varphi : (S,F) \to (S',F')$ and any (S',F')-algebra A', the φ-reduct $A'\restriction_\varphi$ is defined by $(A'\restriction_\varphi)_s = A'_{\varphi^{\mathrm{st}}(s)}$ for each sort symbol $s \in S$, and $(A'\restriction_\varphi)_\sigma = A'_{\varphi^{\mathrm{op}}(\sigma)}$ for each operation symbol σ in F.

(S,F)-*terms* can be defined inductively as follows: for any $\sigma \in F_{s_1\ldots s_n\to s}$, a structure of the form $\sigma(t_1,\ldots,t_n)$ is a term of sort s whenever t_i are terms of sorts s_i, respectively. The set of the (S,F)-terms of sort s is denoted by $(T_{(S,F)})_s$. Each signature morphism $\varphi : (S,F) \to (S',F')$ induces a canonical translation $T_\varphi : T_{(S,F)} \to T_{(S',F')}$ defined by $T_\varphi(\sigma(t_1,\ldots,t_n)) = \varphi^{\mathrm{op}}(T_\varphi(t_1),\ldots,T_\varphi(t_n))$. The *sentences* of the signature (S,F) are the usual first order sentences built from equational atoms of the form $t = t'$, where t and t' are terms of the same sort, by iterative application of Boolean connectives ($\wedge$, $\neg$, $\Rightarrow$, etc.) and quantifiers. For quantifiers this goes formally as follows. For any signature (S,F), a *variable for* (S,F) is a triple $(x,s,(S,F))$ where x is the *name of the variable*, s *its sort*, and (S,F) *its signature*. Any set X of variables for (S,F) such that any two different variables have different names can be added as new constants to (S,F); the extended signature ths obtained is denoted $(S, F\cup X)$ and is formally defined by $(F\cup X)_{w\to s} = F_{w\to s}$ when w is not empty, and $(F\cup X)_{\to s} = F_{\to s}\cup\{(x,s,(S,F)) \mid (x,s,(S,F)) \in X\}$. If ρ is any $(S,F\cup X)$-sentence for a finite set X of variables for (S,F), then $(\forall X)\rho$ and $(\exists X)\rho$ are both (S,F)-sentences. By a *conditional equation* we mean any sentence of the form $(\forall X)H \Rightarrow C$ where H is a finite conjunction of equational atoms and C is a single equational atom. Sentence translations along signature morphisms extend the translations T_φ of terms to sentences; they just rename the sorts and the function symbols according to the respective signature morphisms. The satisfaction of (S,F)-sentences by (S,F)-algebras is the usual Tarskian satisfaction defined inductively on the structure of the sentences. In more detail this means

– $A \models t = t'$ if and only if $A_t = A_{t'}$ where for any term t its *evaluation* in A, denoted by A_t, is defined inductively by the formula $A_{\sigma(t_1,\ldots,t_n)} = A_\sigma(A_{t_1},\ldots,A_{t_n})$.

– $A \models \rho_1 \wedge \rho_2$ if and only if $A \models \rho_1$ and $A \models \rho_2$, $A \models \neg\rho$ if and only if $A \not\models \rho$, etc.

– $A \models (\forall X)\rho$ if and only if $A' \models \rho$ for any $(S, F\cup X)$-expansion A' of A.

Example 2.2 (Predicate logic). This institution is denoted **PDL**. It signatures are triples (S,C,P) where S is a set of sort symbols, $C = (C_s)_{s\in S}$ is a S-indexed family of sets of constant symbols, and $P = (P_w)_{w\in S^*}$ is an S^*-indexed family of predicate or relation symbols. Signature morphisms map the three components of signatures in a compatible way, similar to the signature morphisms of **ALG**. (S,C,P)-models M interpret any sort symbol $s \in S$ as a set M_s, any constant symbol $\sigma \in C_s$ as an element $M_\sigma \in M_s$, and any predicate symbol $\pi \in P_w$ as a relation $M_\pi \subseteq M_w$. (S,C,P)-model homomorphisms $h : M \to N$ are similar to algebra homomorphims, preserving the interpretations of the constants, i.e. $h_s(M_\sigma) = N_\sigma$ for any $\sigma \in C_s$, and of the predicates, i.e. $h_w(M_\pi) \subseteq N_\pi$ for any $\pi \in P_w$. Reducts along signature morphisms are defined like in **ALG**. The sentences, and their translations along signature morphisms are defined also like in **ALG**, with the difference that the atoms of **PDL** consist of expressions of the form $\pi(\sigma_1,\ldots,\sigma_n)$ for $\pi \in P_w$ and $\sigma_1,\ldots,\sigma_n$ string of constants matching w. Then $M \models^{\mathbf{PDL}} \pi(\sigma_1,\ldots,\sigma_n)$ if and only if $(M_{\sigma_1},\ldots,M_{\sigma_n}) \in M_\pi$. This satisfaction relation can be extended to full first order sentences formed from the atoms of **PDL** as in the case of **ALG**.

Example 2.3 (Preordered algebra). This institution, denoted **POA**, represents a dilluted form of rewriting logic [21] in that it considers only *unlabelled* transitions. It is directly realized as a paradigm for specifying transitions by the language CafeOBJ [6].

[2] If $w = s_1 \ldots s_n$ and $a = (a_1,\ldots,a_n)$ then by $h_w(a)$ we mean the tuple $(h_{s_1}(a_1),\ldots,h_{s_n}(a_n))$.

POA has the same signatures as **ALG**, but the models M of a signature (S, F), called *preordered algebras*, interpret any sort symbol $s \in S$ as a preorder relation $(M_s, \leq_s)$ and the operation symbols as monotonic functions with respect to these preorders. Sentences, satisfaction, model reducts and sentence translations along signature morphisms are defined like in **ALG** with the following difference: the atoms are *transitions* of the form $t \leq t'$, with t and t' terms of the same sort, and $M \models^{\mathbf{POA}} t \leq t'$ if and only if $M_t \leq M_{t'}$ for any preordered algebra M of the respective signature.

Example 2.4 (Partial algebra). This institution is denoted **PA**. Here we refer to the partial algebra as used in CASL [22] which represents a slight refinement of the concept of partial algebra as defined in the standard textbook [1].

A *partial algebraic signature* is a tuple (S, TF, PF), where both (S, TF) and (S, PF) are algebraic signatures such that $TF_{w\to s}$ and $PF_{w\to s}$ are always disjoint. TF stands for 'total' function symbols while PF stands for 'partial' function symbols. A *partial algebra* A is just like a total algebra but interpreting the function symbols of PF as partial rather than total functions. This means that for each $\sigma \in PF_{w\to s}$ there is a subset $dom(A_\sigma) \subseteq A_w$ which is the domain of definition of A_σ, i.e. the subset of the arguments for which A_σ is defined. 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 $h_w(A_\sigma(a)) = B_\sigma(h_s(a))$ for each operation $\sigma \in TF_{w\to s} \cup PF_{w\to s}$ and each string of arguments $a \in A_w$ for which $A_\sigma(a)$ is defined. (In particular this also implies that $h_s(a) \in dom(B_\sigma)$.) For any **PA** signature (S, TF, PF), the homomorphisms of partial (S, TF, PF)-algebras compose component-wise as functions, and this yields the category $\mathsf{Mod}^{\mathbf{PA}}(S, TF, PF)$.

The sentences for a signature (S, TF, PF) are built like in the case of the total algebras from existence equality atoms $t \overset{e}{=} t'$ and by restricting the quantification only to sets X of *total variables*, i.e. variables that are added as new constants to TF. An existence equality $t \overset{e}{=} t'$ holds in an algebra A when both terms are defined and are equal. The terms are formed with function symbols from TF and PF, and a term t *is defined in an algebra* A when A_t can be evaluated, which means that by assuming that $t = \sigma(t_1, \ldots, t_n)$ then t is defined in A when each t_i is defined in A and $(A_{t_1}, \ldots, A_{t_n}) \in dom(A_\sigma)$; in this case $A_t = A_\sigma(A_{t_1}, \ldots, A_{t_n})$. The satisfaction of existence equalities by partial algebras is extended to all sentences like in **ALG**; note the role played by the assumption that the quantifications are total.

Example 2.5 (Hidden algebra). Hidden algebra has been introduced in [11, 13] as an algebraic formalism underlying the behavioural specification paradigm and further developed by works such as [7, 18, 25]. In an essential form it can be presented as the following institution, denoted **HA**. The signatures of **HA** are triples (H, V, F) where V and H are sets of *visible* and *hidden* sort symbols, respectively, with $H \cap V = \emptyset$, and $(H \cup V, F)$ is an **ALG** signature. Signature morphisms $\varphi\colon (H, V, F) \to (H', V', F')$ are **ALG** signature morphisms such that $\varphi(H) \subseteq H'$, $\varphi(V) \subseteq V'$ and such that the following encapsulation condition holds: for each operation symbol σ' of F' such that its arity contains a hidden sort of $\varphi(H)$, there exists an operation symbol σ of F such that $\varphi(\sigma) = \sigma'$. The (H, V, F)-models, called (H, V, F)-algebras, are exactly the $(H \cup V, F)$-algebras. A *hidden congruence* on a given (H, V, F)-algebra is a many-sorted congruence which is the equality on the visible sorts. A crucial result in the theory of hidden algebras establishes the existence of the largest hidden congruence on a given algebra A (see [25], for example); this is called the *behavioural equivalence of* A and may be denoted by $\sim_A$. A *homomorphism of hidden algebras* is a homomorphism of ordinary algebras which in addition preserves the behavioural equivalence relations. The sentences in **HA** are defined like those in **ALG** with the difference that the atoms are *behavioural equalities* of the form $t \sim t'$, with t and t' terms of the same sort. An $(H \cup V, F)$-algebra satisfies $t \sim t'$ when $A_t \sim_A A_{t'}$. The encapsulation condition for signature morphisms plays a crucial role for proving the satisfaction condition of **HA**; this connection between its pragmatic object-oriented meaning and its logical significance and has been discovered in [11].

Actual instances of results in this paper often consider institutions having only atoms as sentences.

Notation 2.2 (Atomic sub-institutions). *For each institution $\mathcal{I}$ presented in the examples above, let $A(\mathcal{I})$ denote the 'sub-institution' of $\mathcal{I}$ which has only the atoms of $\mathcal{I}$ as sentences.*

The following property, needed by our work, plays a crucial role for the semantics studies of formal specifications and comes up in very many works in the area, a few early examples being [9, 20, 26, 27]. It is a necessary condition in many model theoretic results using institutions (see [4]), thus being one of the most desirable properties for an institution. It is also not to be confused with a much harder property in conventional first order model theory which refers to amalgamation along elementary extension of models within the same signature.

Definition 2.3 (Model amalgamation). *An institution has* model amalagamation *when for each pushout in the category of it signatures, as in the diagram below,*

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

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 an unique Σ'-model M' such that $M'\restriction_{\theta_1} = M_1$ and $M'\restriction_{\theta_2} = M_2$.

A relaxed variant of this property is obtained by dropping off the uniqueness requirement on M'; this is called weak model amalgamation*.*

Most of the institutions formalizing conventional or non-conventional logics have model amalgamation, including all the examples presented above. An easy proof of model amalgamation in first order logic, which can be easily replicated for the examples presented above in this section, can be found in [4].

3. Quasi-Boolean Encodings

In this section we describe a kind of encodings of abstract institutions to the institution **ALG** of total algebra that capture the phenomenon of conditions as Boolean terms at a general institution-independent level. This concept is illustrated with several examples based upon the actual institutions presented in the previous section. Next we show that on top of such an encoding, in the presence of a rather technical condition easily satisfied by examples, we can define an institution of conditional sentences with conditions in the form of Boolean terms. Actual logical systems underlying the practice of conditions as Boolean terms in OBJ or in CafeOBJ appear as instances of this general construction.

3.1. Quasi-Boolean encodings: definition and examples

Definition 3.1 (Quasi-Boolean encoding). *Let $\mathcal{I} = (\mathbb{S}ig, \mathsf{Sen}, \mathsf{Mod}, \models)$ be any institution. A* quasi-Boolean encoding *of $\mathcal{I}$ consists of the following data:*

1. *a functor $\Phi\colon \mathbb{S}ig \to \mathbb{S}ig^{\mathbf{ALG}}$ such that*
   - *for each signature Σ in $\mathcal{I}$, $\Phi(\Sigma)$ has a distinguished sort $\mathtt{B}_\Sigma$ and a distinguished constant $\mathtt{true}_\Sigma$ of sort $\mathtt{B}_\Sigma$, and*
   - *for each signature morphism φ in $\mathcal{I}$, $\Phi(\varphi)$ preserves $\mathtt{B}$ and $\mathtt{true}$,*
2. *a natural transformation $\alpha\colon \mathsf{Sen} \Rightarrow (T_{\Phi(-)})_{\mathtt{B}}$ (i.e. a mapping of sentences to terms of sorts $\mathtt{B}$), and*
3. *a quasi-natural transformation $\gamma\colon \mathsf{Mod} \Rightarrow \Phi; \mathsf{Mod}^{\mathbf{ALG}}$,*

such that the following Encoding Condition holds:

$$M \models^{\mathcal{I}}_{\Sigma} \rho \quad \textit{if and only if} \quad \gamma_\Sigma(M) \models^{\mathbf{ALG}}_{\Phi(\Sigma)} (\alpha_\Sigma(\rho) = \mathtt{true})$$

for each signature $\Sigma \in |\mathbb{S}ig|$, each Σ-model M, and each Σ-sentence ρ.

In the applications the base institution $\mathcal{I}$ from Def. 3.1 is often an 'atomic institution', i.e. an institution whose sentences are atoms. This is also the case in the following examples. For the following series of examples of quasi-Boolean encodings we give only the construction of the encoding, and invite the reader to check the detailed technical conditions by [her/him]self.

Example 3.1 (Equational logic). This is a quasi-Boolean encoding of $A(\mathbf{ALG})$ which underlies the practice of conditions as Boolean terms in OBJ and the equational logic part of CafeOBJ, the operations $\circleddash_s$ below corresponding to the operations $==_s$ in OBJ/CafeOBJ. A key aspect of this encoding is that the semantic equalities $a \circleddash b$ for which $a \neq b$ are not collapsed to a value representing 'false'.
For each algebraic signature (S, F), $\Phi(S, F) = (S \uplus \{\mathtt{B}\}, F^*)$ where

- $F^*_{w\to s} = F_{w\to s}$ when $s \neq \mathtt{B}$,
- $F^*_{ss\to\mathtt{B}} = \{\circleddash_s\}$, $F^*_{\to\mathtt{B}} = \{\mathtt{true}\}$, and
- $F^*_{w\to\mathtt{B}} = \emptyset$ otherwise.

For each (S,F)-algebra A, $\gamma_{(S,F)}(A)$ expands A as follows:

- $\gamma(A)_{\mathtt{B}} = \{1\} \cup \{(s,a,b) \mid s \in S, a \neq b \in A_s\}$ and
- $\gamma(A)_{\mathtt{true}} = 1$ and $\gamma(A)_{\circleddash_s}(a,b) = \begin{cases} 1 & \text{when } a = b \\ (s,a,b) & \text{when } a \neq b. \end{cases}$

For each (S,F)-homomorphism $h\colon A \to B$, $\gamma_{(S,F)}(h)$ expands h as follows:

- $\gamma(h)_{\mathtt{B}}(x) = \begin{cases} 1 & \text{when } x = 1 \text{ or } x = (s,a,b) \text{ and } h(a) = h(b) \\ (s,h(a),h(b)) & \text{when } x = (s,a,b) \text{ and } h(a) \neq h(b). \end{cases}$

For each morphism of signatures $\varphi\colon (S,F) \to (S',F')$ and each (S',F')-algebra A', the algebra homomorphism $(\gamma_\varphi)_{A'}\colon \gamma_{(S,F)}(A'\upharpoonright_\varphi) \to \gamma_{(S',F')}(A')\upharpoonright_{\Phi(\varphi)}$ is identity on the sorts $s \neq \mathtt{B}$ and maps 1 to 1 and each (s,a,b) to $(\varphi(s),a,b)$.
For all terms t_1 and t_2 of sort s,

- $\alpha_{(S,F)}(t_1 = t_2) = (t_1 \circleddash_s t_2)$.

Example 3.2 (Predicate logic). This quasi-Boolean encoding of A(**PDL**), i.e. the atomic part of **PDL**, has been first defined in [3].
For each **PDL** signature (S,C,P), $\Phi(S,C,P) = (S \uplus \{\mathtt{B}\}, C \oplus P)$ where

- $(C\oplus P)_{\to s} = C_{\to s}$ when $s \neq \mathtt{B}$,
- $(C\oplus P)_{w\to\mathtt{B}} = \{\circled{\pi} \mid \pi \in P_w\}$, when w is non-empty,
- $(C\oplus P)_{\to\mathtt{B}} = \{\mathtt{true}\}$, and
- $(C\oplus P)_{w\to s} = \emptyset$ when w is non-empty and $s \neq \mathtt{B}$.

For each (S,C,P)-model M, $\gamma_{(S,C,P)}(M)$ expands M as follows:

- $\gamma(M)_{\mathtt{B}} = \{1\} \cup \{(\pi,a_1,\ldots,a_n) \mid \pi \in P_{s_1\ldots,s_n}, (a_1,\ldots,a_n) \notin A_{s_1\ldots s_n} \setminus M_\pi\}$, and
- $\gamma(M)_{\mathtt{true}} = 1$ and $\gamma(M)_{\circled{\pi}}(a_1,\ldots,a_n) = \begin{cases} 1 & \text{when } (a_1,\ldots,a_n) \in M_\pi \\ (\pi,a_1,\ldots,a_n) & \text{when } (a_1,\ldots,a_n) \notin M_\pi \end{cases}$

For each (S,C,P)-homomorphism $h\colon M \to N$, $\gamma_{(S,C,P)}(h)$ expands h as follows:

- $\gamma(h)_{\mathtt{B}}(x) = \begin{cases} 1 & \text{when } x = 1 \text{ or} \\ & x = (\pi,a_1,\ldots,a_n) \text{ and } (h(a_1),\ldots,h(a_n)) \in N_\pi \\ (\pi,h(a_1),\ldots,h(a_n)) & \text{when } x = (\pi,a_1,\ldots,a_n) \text{ and } (h(a_1),\ldots,h(a_n)) \notin N_\pi \end{cases}$

For each morphism of signatures $\varphi\colon (S,C,P) \to (S',C',P')$ and each (S',C',P')-model M', the model homomorphism $(\gamma_\varphi)_{M'}\colon \gamma_{(S,C,P)}(M'\upharpoonright_\varphi) \to \gamma_{(S',C',P')}(M')\upharpoonright_{\Phi(\varphi)}$ is identity on the sorts $s \neq \mathtt{B}$ and maps 1 to 1 and each $(\pi,a_1,\ldots,a_n)$ to $(\varphi(\pi),a_1,\ldots,a_n)$.
For any relation symbol π and any terms $t_1,\ldots,t_n$ of appropriate sorts,

- $\alpha_{(S,C,P)}(\pi(t_1,\ldots,t_n)) = \circled{\pi}(t_1,\ldots,t_n)$.

Example 3.3 (Preordered algebra). This quasi-Boolean encoding of $A(\mathbf{POA})$, i.e. the atomic part of **POA**, underlies the implementation of the preordered algebra specification paradigm in CafeOBJ. The CafeOBJ operations ==> correspond to the operations $\ominus$ below.
For each algebraic signature (S, F), $\Phi(S, F) = (S \uplus \{\mathtt{B}\}, F^*)$ where

- $F^*_{w\to s} = F_{w\to s}$ when $s \neq \mathtt{B}$,
- $F^*_{ss\to\mathtt{B}} = \{\ominus_s\}$, $F^*_{\to\mathtt{B}} = \{\mathtt{true}\}$, and
- $F^*_{w\to\mathtt{B}} = \emptyset$ otherwise.

For each (S, F)-algebra A, $\gamma_{(S,F)}(A)$ first forgets the preorder relations of A and then expands the resulting (S, F)-algebra as follows:

- $\gamma(A)_{\mathtt{B}} = \{1\} \cup \{(s, a, b) \mid s \in S, a \not\leq b \in A_s\}$, and
- $\gamma(A)_{\mathtt{true}} = 1$ and $\gamma(A)_{\ominus_s}(a, b) = \begin{cases} 1 & \text{when } a \leq b \\ (s, a, b) & \text{when } a \not\leq b. \end{cases}$

For each (S, F)-homomorphism $h\colon A \to B$, $\gamma_{(S,F)}(h)$ expands h as follows:

- $\gamma(h)_{\mathtt{B}}(x) = \begin{cases} 1 & \text{when } x = 1 \text{ or } x = (s, a, b) \text{ and } h(a) \leq h(b) \\ (s, h(a), h(b)) & \text{when } x = (s, a, b) \text{ and } h(a) \not\leq h(b). \end{cases}$

For each morphism φ of **POA** signatures, γ_φ is defined like in Ex. 3.1.
For all terms t_1 and t_2 of sort s,

- $\alpha_{(S,F)}(t_1 \Rightarrow t_2) = (t_1 \ominus_s t_2)$.

Example 3.4 (Partial algebra). This quasi-Boolean encoding of $A(\mathbf{PA})$, i.e. the atomic part of **PA**, has been defined in [5]. A key aspect of this encoding is that the syntactic existence equality relation $\stackrel{e}{=}$ is represented as an algebraic operation ⓔ, the 'undefined' expressions are not collapsed to one single 'undefined' value, and the semantic equalities aⓔ$\,b$ for which a and b are not equal or one of them is an 'undefined' element are not collapsed to a value representing 'false'.
For each **PA** signature (S, TF, PF), $\Phi(S, TF, PF) = (S \uplus \{\mathtt{B}\}, TF \oplus PF)$ where

- $(TF \oplus PF)_{w\to s} = TF_{w\to s} \cup PF_{w\to s}$ when $s \neq \mathtt{B}$,
- $(TF \oplus PF)_{ss\to\mathtt{B}} = \{ⓔ_s\}$, $(TF \oplus PF)_{\to\mathtt{B}} = \{\mathtt{true}\}$, and
- $(TF \oplus PF)_{w\to\mathtt{B}} = \emptyset$ otherwise.

For each (S, TF, PF)-algebra A, $\gamma_{(S,TF,PF)}(A)$ is defined as follows:

1. Let $(S, TF{+}PF{+}A)$ be the algebraic signature which adds each element of A_s as a new constant of sort s to the signature that puts together the total and the partial operation symbols of (S, TF, PF).
2. Let A^* be the initial $(S, TF{+}PF{+}A)$-algebra satisfying all equations

   $$\sigma(a_1, \ldots, a_n) = A_\sigma(a_1, \ldots, a_n)$$

   for all $(a_1, \ldots, a_n) \in dom(A_\sigma)$.
3. Then $\gamma_{(S,TF,PF)}(A)$ is defined as follows:
   - $\gamma(A)_s = A^*_s$ for each $s \in S$,
   - $\gamma(A)_{\mathtt{B}} = \{1\} \cup \{(s, a, a') \mid a, a' \in A^*_s,\ \ a \neq a' \text{ or } a \notin A_s\}$,
   - $\gamma(A)_\sigma = A^*_\sigma$ for any operation symbol σ from TF or PF,
   - $\gamma(A)_{\mathtt{true}} = 1$, and

– $\gamma(A)_{ⓔ_s}(a, a') = \begin{cases} 1 & \text{when } a = a' \in A_s, \\ (s, a, a') & \text{otherwise.} \end{cases}$

For each (S, TF, PF)-homomorphism $h\colon A \to B$, $\gamma_{(S,TF,PF)}(h)$ is defined as follows:

- The $(S, TF{+}PF{+}B)$-algebra B^* can be regarded as a $(S, TF{+}PF{+}A)$-algebra B^*_h by letting $(B^*_h)_a = h(a)$ for each element a of A. Because h is a homomorphism of partial algebras it is easy to show that B^*_h, satisfies the equations defining A^*. Hence let h^* be the unique homomorphism $A^* \to B^*_h$.
- Then $\gamma(h)$ is defined as follows:
  - $\gamma(h)_s = h^*_s$ for each $s \in S$, and
  - $\gamma(h)_{\mathtt{B}}(x) = \begin{cases} 1 & \text{when } x = 1 \text{ or } x = (s, a, b) \text{ and } a, b \in A_s \text{ and } h(a) = h(b) \\ (s, h^*(a), h^*(b)) & \text{when } x = (s, a, b) \text{ and } h(a) \neq h(b) \text{ or } a \notin A_s. \end{cases}$

For each morphism of signatures $\varphi\colon (S, TF, PF) \to (S', TF', PF')$ and each (S', TF', PF')-algebra A', the homomorphism $(\gamma_\varphi)_{A'}\colon \gamma_{(S,TF,PF)}(A'\upharpoonright_\varphi) \to \gamma_{(S',TF',PF')}(A')\upharpoonright_{\Phi(\varphi)}$ is defined as follows. Let $\varphi^*\colon (S, TF + PF + A\upharpoonright_\varphi) \to (S', TF' + PF' + A')$ be the morphism of **ALG** signatures that is determined canonically by φ. Note that the set of equations defining A'^* contains the translations by φ^* of the equations defining $A\upharpoonright^*_\varphi$. Then $(\gamma_\varphi)_{A'}$ is the expansion of the unique homomorphism $(A'\upharpoonright_\varphi)^* \to A'^*\upharpoonright_{\varphi^*}$ that maps 1 to 1 and each (s, a, a') to $(\varphi(s), a, a')$.
For all $(S, TF{+}PF)$-terms t_1 and t_2 of sort s,

- $\alpha_{(S,TF,PF)}(t_1 \overset{\mathtt{e}}{=} t_2) = (t_1 ⓔ_s t_2)$.

Example 3.5 (Hidden algebra). This quasi-Boolean encoding of $A(\mathbf{HA})$, i.e. the atomic part of **HA**, underlies the implementation of the behavioural specification paradigm in CafeOBJ. The CafeOBJ operations =b= correspond to the operations $\circleddash$ below.
For each **HA** signature (H, V, F), $\Phi(H, V, F) = (H \cup V \uplus \{\mathtt{B}\}, F^*)$ where

- $F^*_{w\to s} = F_{w\to s}$ when $s \neq \mathtt{B}$,
- $F^*_{ss\to\mathtt{B}} = \{\circleddash_s\}$, $F^*_{\to\mathtt{B}} = \{\mathtt{true}\}$, and
- $F^*_{w\to\mathtt{B}} = \emptyset$ otherwise.

For each (H, V, F)-algebra A, $\gamma_{(H,V,F)}(A)$ expands A as follows:

- $\gamma(A)_{\mathtt{B}} = \{1\} \cup \{(s, a, b) \mid s \in S, a \nsim_A b \in A_s\}$, and
- $\gamma(A)_{\mathtt{true}} = 1$ and $\gamma(A)_{\circleddash_s}(a, b) = \begin{cases} 1 & \text{when } a \sim_A b \\ (s, a, b) & \text{when } a \nsim_A b. \end{cases}$

For each homomorphism of (H, V, F)-algebras $h\colon A \to B$, $\gamma_{(H,V,F)}(h)$ expands h as follows:

- $\gamma(h)_{\mathtt{B}}(x) = \begin{cases} 1 & \text{when } x = 1 \text{ or } x = (s, a, b) \text{ and } h(a) \sim_B h(b) \\ (s, h(a), h(b)) & \text{when } x = (s, a, b) \text{ and } h(a) \nsim_B h(b). \end{cases}$

For each morphism φ of **HA** signatures, γ_φ is defined like in Ex. 3.1.
For all terms t_1 and t_2 of sort s,

- $\alpha_{(S,F)}(t_1 \sim t_2) = (t_1 \circleddash_s t_2)$.

Example 3.6 (Adding Boolean logical connectives as algebraic operations). All examples above can be developed further by coding the Boolean logical connectives as operations on the sort B. Let us present in detail this idea only for equational logic (Ex. 3.1), since the other examples presented above in this section can be upgraded similarly.
Our upgrading of Ex. 3.1 starts by including all Boolean operations to F^* as follows:

– $F^{*}_{\to\mathtt{B}} = \{\mathtt{true}, \mathtt{false}\}$, $F^{*}_{\mathtt{B}\to\mathtt{B}} = \{\ominus\}$ and $F^{*}_{\mathtt{BB}\to\mathtt{B}} = \{\owedge, \ovee\}$.

For any (S, F)-algebra A, the algebra $\gamma_{(S,F)}(A)$ gets upgraded to the expansion of A such that $(\gamma(A)_{\mathtt{B}}, \owedge, \ovee, \ominus, \mathtt{true}, \mathtt{false})$ is the Boolean algebra freely generated by the set $\{(s, a, b) \mid s \in S,\ a \neq b \in A_s\}$.
For each (S, F)-homomorphism $h\colon A \to B$, the homomorphism $\gamma_{(S,F)}(h)$ expands h such that $\gamma(h)_{\mathtt{B}}$ is the unique algebra homomorphism $(\gamma(A)_{\mathtt{B}}, \owedge, \ovee, \ominus, \mathtt{true}, \mathtt{false}) \to (\gamma(B)_{\mathtt{B}}, \owedge, \ovee, \ominus, \mathtt{true}, \mathtt{false})$ extending the function that maps (s, a, b) to $\begin{cases} 1 & \text{when } h(a) = h(b) \\ (s, h(a), h(b)) & \text{when } h(a) \neq h(b). \end{cases}$
For each morphism of signatures $\varphi\colon (S, F) \to (S', F')$ and each (S', F')-algebra A', the homomorphism $(\gamma_\varphi)_{A'}$ is identity on the sorts $s \neq \mathtt{B}$ and $((\gamma_\varphi)_{A'})_{\mathtt{B}}$ is the unique Boolean algebra homomorphism that maps each (s, a, b) to $(\varphi(s), a, b)$.

An important aspect of this logical semantics of the Boolean connectives, which constitutes one of the main motivation for our study, is that it is different from the standard one since equivalence relationships such as

$A \models \neg\rho$ if and only if $\gamma(A) \models \ominus\alpha(\rho)$, or

$A \models \rho_1 \vee \rho_2$ if and only if $\gamma(A) \models \alpha(\rho_1) \ovee \alpha(\rho_2)$

do *not* hold in general. This is due to the fact that the Boolean algebras $(\gamma(A)_{\mathtt{B}}, \owedge, \ovee, \ominus, \mathtt{true}, \mathtt{false})$ in general consist of more that two values. If one attempts to fix this by defining $\gamma(A)_{\mathtt{B}}$ to consist of two values only by collapsing all values (s, a, b) for $a \neq b$ to $\mathtt{false}$, then it is not possible anymore to have γ defined on non-injective homomorphisms. In order to see this, it is enough to consider $a \neq b \in A$ and $h\colon A \to B$ such that $h(a) = h(b)$. Thus $A_{\circeq}(a, b) = A_{\mathtt{false}}$. Since h is homomorphism, on the one hand $h(A_{\circeq}(a, b)) = B_{\circeq}(h(a), h(b)) = B_{\mathtt{true}}$, and on the other hand $h(A_{\mathtt{false}}) = B_{\mathtt{false}}$, hence $B_{\mathtt{true}} = B_{\mathtt{false}}$. However, as we will see below in the paper, the fact that $\gamma_{(S,F)}$ as a functor is defined also on *all* homomorphisms plays a crucial role for initial semantics. In fact it is exactly the difference between this semantics of the Boolean connectives as algebraic operations and the standard semantics the Boolean logical connectives that is responsible for the possibility of initial semantics for sentences conditioned by any Boolean expression formed over atoms.

3.2. Institutions with quasi-Boolean conditioned sentences

On top of a quasi-Boolean encoding of an institution $\mathcal{I}$ we define an extension of $\mathcal{I}$ in which the sentences are conditioned by quasi-Boolean terms (i.e. terms of sort $\mathtt{B}$) and are universally quantified. We define this extension at the fully general level of abstract quasi-Boolean encodings. Particular concrete examples include the institutions underlying the practice of conditions as Boolean terms in OBJ and in the various specification logics of CafeOBJ; also this construction can be applied to specification frameworks based upon other logics, such as partial algebra.

The universally quantified conditioned sentences are introduced in two steps. First we introduce the conditioned sentences without quantifiers and then we apply a general universal quantification construction to the result of the first step.

Definition 3.2 (Truth injective). *A quasi-Boolean encoding* (Φ, α, γ) *is* truth injective *when for each signature morphism* $\varphi\colon \Sigma \to \Sigma'$ *and each* Σ'*-model* M' *we have that* $(\gamma_\varphi)^{-1}_{M'}((\gamma_{\Sigma'}(M')\restriction_{\Phi(\varphi)})_{\mathtt{true}}) = \{(\gamma_\Sigma(M'\restriction_\varphi))_{\mathtt{true}}\}$.

Truth injectivity just says that the homomorphisms $(\gamma_\varphi)_{M'}$ do not map to $\mathtt{true}$ any value that is not $\mathtt{true}$. This is a merely technical property that is satisfied by all quasi-Boolean encodings presented in Examples 3.1-3.6; we invite the reader to check this fact by [her/him]self.

Theorem 3.1. *For any truth injective quasi-Boolean encoding* (Φ, α, γ) *of an institution* $\mathcal{I}$ *the following defines an institution, denoted* $C(\Phi, \alpha, \gamma)$ *or just* C *when there is no danger of confusion:*

– $\mathbb{S}ig^{C} = \mathbb{S}ig^{\mathcal{I}}$ *and* $\mathsf{Mod}^{C} = \mathsf{Mod}^{\mathcal{I}}$,

– $\mathsf{Sen}^{C}(\Sigma) = \{H \Rightarrow C \mid H \in (T_{\Phi(\Sigma)})_{\mathtt{B}}, C \in \mathsf{Sen}^{\mathcal{I}}(\Sigma)\}$ *for each signature* Σ*, and*

– $\mathsf{Sen}^{C}(\varphi)(H \Rightarrow C) = ((T_{\Phi(\varphi)})_{\mathtt{B}}(H) \Rightarrow \mathsf{Sen}^{\mathcal{I}}(\varphi)(C))$ *for each signature morphism* $\varphi\colon \Sigma \to \Sigma'$*, and*

– $M \models^{C}_{\Sigma} (H \Rightarrow C)$ *if and only if* $M \models^{\mathcal{I}}_{\Sigma} C$ *when* $\gamma_{\Sigma}(M) \models^{\mathbf{ALG}}_{\Phi(\Sigma)} (H = \mathtt{true})$.

Proof. We check only the Satisfaction Condition for $C(\Phi, \alpha, \gamma)$ since in this case the other institution axioms hold rather trivially. Let $\varphi\colon \Sigma \to \Sigma'$ be any signature morphism, M' be a Σ'-model, and let $(H \Rightarrow C) \in \mathsf{Sen}^{C}(\Sigma)$. We have to show that

$$M' \models^{C}_{\Sigma'} ((T_{\Phi(\varphi)})_{\mathrm{B}}(H) \Rightarrow \mathsf{Sen}^{\mathcal{I}}(\varphi)(C)) \ \text{ if and only if } \ M' \upharpoonright_{\varphi} \models^{C}_{\Sigma} (H \Rightarrow C).$$

By the Satisfaction Condition for $\mathcal{I}$ we have that $M' \models^{\mathcal{I}}_{\Sigma'} \mathsf{Sen}^{\mathcal{I}}(\varphi)(C)$ if and only if $M' \upharpoonright_{\varphi} \models^{\mathcal{I}}_{\Sigma} C$. This means that it is enough to show that

$$\gamma_{\Sigma'}(M') \models^{\mathbf{ALG}}_{\Phi(\Sigma')} ((T_{\Phi(\varphi)})_{\mathrm{B}}(H) = \mathtt{true}) \ \text{ if and only if } \ \gamma_{\Sigma}(M' \upharpoonright_{\varphi}) \models^{\mathbf{ALG}}_{\Phi(\Sigma)} (H = \mathtt{true}). \tag{1}$$

We need the following lemma whose proof, omitted here, consists of a simple induction process on the structure of the term t.

Lemma 3.1. *For any* **ALG** *signature* Σ*, any homomorphism of* Σ*-models* $h\colon A \to B$*, and any* Σ*-term* t*, we have that* $h(A_t) = B_t$*.*

By induction on the structure of the term H and by Lemma 3.1 applied for the model homomorphism $\gamma_{\varphi}(M')$ we have that

$$\gamma_{\Sigma'}(M')_{(T_{\Phi(\varphi)})_{\mathrm{B}}(H)} = (\gamma_{\Sigma'}(M') \upharpoonright_{\Phi(\varphi)})_H = (\gamma_{\varphi})_{M'}((\gamma_{\Sigma}(M' \upharpoonright_{\varphi}))_H). \tag{2}$$

By Lemma 3.1 applied for the model homomorphism $\gamma_{\varphi}(M')$ we also have that

$$\gamma_{\Sigma'}(M')_{\mathtt{true}} \ (= (\gamma_{\Sigma'}(M') \upharpoonright_{\Phi(\varphi)})_{\mathtt{true}}) \ = (\gamma_{\varphi})_{M'}((\gamma_{\Sigma}(M' \upharpoonright_{\varphi}))_{\mathtt{true}}). \tag{3}$$

The implication from the right to the left in (1) follows immediately from (2) and (3). The implication from the left to the right in (1) follows from (2) by the truth injectivity hypothesis. □

For the rest of this paper we assume that all abstract quasi-Boolean encodings are truth injective.

The main idea of the following treatment of quantifiers at the level of abstract institutions originates probably from [28] (see also [4]).

Proposition 3.1. *Let* $\mathcal{I}$ *be any institution with pushout of signatures and weak model amalgamation and let* $\mathcal{D}$ *be a class of its signature morphisms such that for any signature morphisms* $(\chi\colon \Sigma \to \Sigma') \in \mathcal{D}$ *and* $\varphi\colon \Sigma \to \Sigma_1$ *there is a designated pushout*

$$\begin{array}{ccc}
\Sigma & \xrightarrow{\varphi} & \Sigma_1 \\
{\scriptstyle\chi}\downarrow & & \downarrow{\scriptstyle\chi(\varphi)} \\
\Sigma' & \xrightarrow[\varphi[\chi]]{} & \Sigma'_1
\end{array}$$

with $\chi(\varphi) \in \mathcal{D}$ *and such that the 'horizontal' composition of such designated pushouts is a designated pushout too, i.e. for the pushouts of the following diagram*

$$\begin{array}{ccccc}
\Sigma & \xrightarrow{\varphi} & \Sigma_1 & \xrightarrow{\theta} & \Sigma_2 \\
{\scriptstyle\chi}\downarrow & & \downarrow{\scriptstyle\chi(\varphi)} & & \downarrow{\scriptstyle\chi(\varphi)(\theta)} \\
\Sigma' & \xrightarrow[\varphi[\chi]]{} & \Sigma'_1 & \xrightarrow[\theta[\chi(\varphi)]]{} & \Sigma'_2
\end{array}$$

we have that $\varphi[\chi]; \theta[\chi(\varphi)] = (\varphi; \theta)[\chi]$ *and* $\chi(\varphi)(\theta) = \chi(\varphi; \theta)$*, and such that*
$\chi(1_{\Sigma}) = \chi$ *and* $1_{\Sigma}[\chi] = 1_{\Sigma}$*.*

Then the following data defines an institution, called the institution of universally $\mathcal{D}$-quantified sentences over $\mathcal{I}$ *and denoted* $\forall_{\mathcal{D}}\mathcal{I}$*, or just* $\forall\mathcal{I}$ *when* $\mathcal{D}$ *is clearly fixed from the context:*

– $\mathbb{S}ig^{\forall\mathcal{I}} = \mathbb{S}ig^{\mathcal{I}}$ *and* $\mathsf{Mod}^{\forall\mathcal{I}} = \mathsf{Mod}^{\mathcal{I}}$*,*

– $\mathsf{Sen}^{\forall\mathcal{I}}(\Sigma) = \{(\forall\chi)\rho' \mid (\chi\colon \Sigma \to \Sigma') \in \mathcal{D}$ *and* $\rho' \in \mathsf{Sen}^{\mathcal{I}}(\Sigma')\}$ *for each signature* Σ,

– $\mathsf{Sen}^{\forall\mathcal{I}}(\varphi)((\forall\chi)\rho') = (\forall\chi(\varphi))\mathsf{Sen}^{\mathcal{I}}(\varphi[\chi])(\rho')$ *for each signature morphism* $\varphi\colon \Sigma \to \Sigma_1$, *and*

– $M \models^{\forall\mathcal{I}}_{\Sigma} (\forall\chi)\rho'$ *if and only if* $M' \models^{\mathcal{I}}_{\Sigma} \rho'$ *for all* χ*-expansions* M' *of* M.

Proof. We have to prove only the functoriality property for $\mathsf{Sen}^{\forall\mathcal{I}}$ and the Satisfaction Condition for $\models^{\forall\mathcal{I}}$. The former follows immediately from the compositionality hypotheses on $\mathcal{D}$. For showing the latter we consider a signature morphism $\varphi\colon \Sigma \to \Sigma_1$, any Σ_1-model M_1, and any Σ-sentence $(\forall\chi)\rho'$ where $(\chi\colon \Sigma \to \Sigma') \in \mathcal{D}$. We have to prove that

$$M_1 \models^{\forall\mathcal{I}}_{\Sigma_1} (\forall\chi(\varphi))\mathsf{Sen}^{\mathcal{I}}(\varphi[\chi])(\rho') \text{ if and only if } M_1\restriction_{\varphi} \models^{\forall\mathcal{I}}_{\Sigma} (\forall\chi)\rho'$$

For the implication from the right to the left, for any $\chi(\varphi)$-expansion M'_1 of M_1 we have that $M'_1\restriction_{\varphi[\chi]}$ is a χ-expansion of $M_1\restriction_{\varphi}$ and thus by the hypothesis $M'_1\restriction_{\varphi[\chi]} \models^{\mathcal{I}}_{\Sigma'} \rho'$. By the Satisfaction Condition for $\mathcal{I}$ this implies $M'_1 \models^{\mathcal{I}}_{\Sigma'_1} \mathsf{Sen}^{\mathcal{I}}(\varphi[\chi])(\rho')$.

For the implication from the left to the right we consider any χ-expansion M' of $M_1\restriction_{\varphi}$. By the weak model amalgamation hypothesis for $\mathcal{I}$, there exists a Σ'_1-model M'_1 such that $M'_1\restriction_{\chi(\varphi)} = M_1$ and $M'_1\restriction_{\varphi[\chi]} = M'$. Because $M_1 \models^{\forall\mathcal{I}}_{\Sigma_1} (\forall\chi(\varphi))\mathsf{Sen}^{\mathcal{I}}(\varphi[\chi])(\rho')$ it follows that $M'_1 \models^{\mathcal{I}}_{\Sigma'_1} \mathsf{Sen}^{\mathcal{I}}(\varphi[\chi])(\rho')$ and by the Satisfaction Condition for $\mathcal{I}$ that $M'_1\restriction_{\varphi[\chi]} \models^{\mathcal{I}}_{\Sigma'} \rho'$. □

Example 3.7. For the institutions presented in Sect. 2, the sentences of the form $(\forall X)\rho'$, where ρ' is a quantifier-free sentence, are special cases of universally $\mathcal{D}$-quantified sentences in the sense of Prop. 3.1 when we consider $\mathcal{D}$ to consist of all signature extensions with a finite set of variables in the case of **ALG**, **PDL**, **POA**, **HA**, and with a finite set of *total* variables in the case of **PA**. For these examples the designated pushouts from Prop. 3.1 are defined as follows. If χ is an extension of a signature Σ with a finite set X of variables and $\varphi\colon \Sigma \to \Sigma_1$ is a signature morphism then $\chi(\varphi)$ is the extension of Σ_1 with the set X^{φ} of variables where $X^{\varphi} = \{(x, \varphi(s), \Sigma_1) \mid (x, s, \Sigma) \in X\}$.

Example 3.8. The institutions of universally quantified sentences conditioned by (quasi-)Boolean terms from OBJ and CafeOBJ can be obtained by applying the construction of Prop. 3.1 through Ex. 3.7 to the result of the construction of Thm. 3.1 applied to the corresponding examples of quasi-Boolean encodings presented in Sect. 3.1. Thus the respective institutions underlying the equational specification paradigm in OBJ and CafeOBJ, the preordered algebra specification and behavioural specification in CafeOBJ arise as $\forall_{\mathcal{D}}C(\Phi, \alpha, \gamma)$ for $\mathcal{D}$ respective classes of signature extensions with finite sets of variables and where (Φ, α, γ) are the quasi-Boolean encodings of the variants of Ex. 3.6 corresponding to A(**ALG**), A(**POA**), and A(**HA**) respectively.

4. Initial semantics

In this section we establish a set of general and widely applicable conditions for the existence of initial semantics for institutions of universally quantified sentences conditioned by quasi-Boolean terms, i.e. institutions of the form $\forall_{\mathcal{D}}C(\Phi, \alpha, \gamma)$ as defined by Thm. 3.1 and 3.1. Since we are aiming here for a general result in line with the developments in Sect. 3.1, we use the method of abstract quasi-varieties. This means that we have to show that universally quantified sentences conditioned by quasi-Boolean terms are preserved by direct products and sub-models (i.e. their models form a quasi-variety) and then use a general result on existence of initial models for quasi-varieties. In the literature there are several general approaches on quasi-varieties at the level of abstract categories with only slight technical differences among them. Since the concept of direct product is a standard categorical concept, all above mentioned approaches are essentially abstract definitions for notions of 'sub-models'. Here we use the approach of [4] that handles abstractly the concept of 'sub-model' via the so-called inclusion systems of [9].

Let us recall the definition of inclusion systems from [4], which is a slightly simplified variant of the original definition given in [9].

Definition 4.1 (Inclusion system). $\langle\mathcal{I}, \mathcal{E}\rangle$ *is a* inclusion system *for a category* $\mathbb{C}$ *if* $\mathcal{I}$ *and* $\mathcal{E}$ *are two sub-categories with* $|\mathcal{I}| = |\mathcal{E}| = |\mathbb{C}|$ *such that*

1. $\mathcal{I}$ *is a partial order (with the ordering relation denoted by* $\subseteq$ *or by* $\hookrightarrow$*), and*
2. *every arrow* f *in* $\mathbb{C}$ *can be factored uniquely as* $f = e_f; i_f$ *with* $e_f \in \mathcal{E}$ *and* $i_f \in \mathcal{I}$.

The arrows of $\mathcal{I}$ *are called* abstract inclusions*, and the arrows of* $\mathcal{E}$ *are called* abstract surjections*. The domain of the inclusion* i_f *in the factorization of* f *is called the* image of f *and is denoted as* $\mathrm{Im}(f)$ *or* $f(A)$ *when* A *is a domain of* f*. When* $f\colon A \to B$ *is an abstract inclusion,* A *is called a* sub-model *of* B.

From the multitude of examples of inclusion systems used in specification theory and in model theory (many of them can be found in [4]) we present below only the examples that are going to be used in our current work.

Example 4.1 (Inclusion systems for models in ALG, PDL, POA, and PA). According to the terminology of [4] a homomorphism $h\colon M \to N$ of models for a signature is *closed* when

- $h^{-1}(N_\pi) = M_\pi$ for each relation symbol π of the signature, in the case of **PDL**,
- $m_1 \leq m_2$ if $h(m_1) \leq h(m_2)$ for any elements m_1, m_2 of M of the same sort, in the case of **POA**, and
- $m \in dom(M_\sigma)$ if $h(m) \in dom(N_\sigma)$ for any $m \in M_w$ and any partial operation symbol σ with arity w, in the case of **PA**.

Then a model homomorphism

- is an abstract inclusion when it is a set theoretic inclusion on each of its components and in the case of **PDL**, **POA**, and **PA** is also closed, and
- is an abstract surjection when it is surjective on each of its components in the case of **ALG**, **PDL**, and **POA**, and when it is epimorphism in the case of **PA**.

The institutions **PDL**, **POA**, and **PA** admit also other non-trivial inclusion systems (see [4]) but those do not fit our aims here. The institution **HA** does not admit non-trivial inclusion systems for its categories of models; however this does not pose any problem here since we do not aim to establish initial semantics for the **HA** related institution of universally quantified quasi-Boolean terms conditioned sentences because behavioural specification is a loose or a final semantics specification paradigm.

Definition 4.2 (Sub-model). *In any institution that is equipped with inclusion systems for its categories of models, we say that* M *is a* sub-model *of* N *when* M *is a sub-object of* N *with respect to the inclusion system of the category of models of the respective signature.*

Let us recall from [4] the abstract concept of quasi-variety.

Definition 4.3 (Categorical quasi-variety). *A class* Q *of objects of a category with direct products of models and with a designated inclusion system is a* quasi-variety *when is closed under direct products and sub-objects.*

Within our abstract framework, let us also recall the following model theoretic terminology about 'preservation of sentences'.

Definition 4.4 (Preservation by sub-models). *In any institution that is equipped with inclusion systems for its categories of models, a* Σ*-sentence* ρ *is preserved by sub-models when for any two* Σ*-models* M *and* N*, if* $N \models \rho$ *and* M *is a sub-model of* N *then* $M \models \rho$ *too.*

Example 4.2. Each sentence of A(**ALG**), A(**PDL**), A(**POA**), and A(**PA**) is preserved by sub-models with respect to the corresponding inclusion system of Ex. 4.1. This fact is rather easy to check, therefore we omit this check here.

Definition 4.5 (Preservation by direct products). *In any institution such that any of its categories of models has small (direct) products (denoted* $\prod$*), a* Σ*-sentence* ρ *is preserved by direct products when for any family* $(M_i)_{i\in I}$ *of* Σ*-models, if* $M_i \models \rho$ *for each* $i \in I$*, then* $\prod_{i\in I} M_i \models \rho$ *too.*

Example 4.3. The institutions **ALG**, **PDL**, **POA**, **PA**, and **HA** admit direct products of models and their atoms, i.e. the sentences of $A(\mathbf{ALG})$, $A(\mathbf{PDL})$, $A(\mathbf{POA})$, $A(\mathbf{PA})$, and $A(\mathbf{HA})$ respectively, are preserved by the direct products of models. This fact is also rather easy to check and therefore we also omit this check here.

Now we are ready to proceed with the development of the general result on existence of initial semantics for the institutions of the form $\forall C(\Phi, \alpha, \gamma)$. This is done in two steps corresponding to the quantifier-free and to the quantified layers of these institutions. For the rest of this section we assume a fixed quasi-Boolean encoding (Φ, α, γ) of an institution $\mathcal{I} = (\mathbb{S}ig, \mathsf{Sen}, \mathsf{Mod}, \models)$ that has direct products of models and that comes equiped with inclusion systems for its categories of models. By C we denote the institution $C(\Phi, \alpha, \gamma)$ of Thm. 3.1.

Proposition 4.1. *If each sentence of $\mathcal{I}$ is preserved by direct products then each sentence of C is preserved by direct products too.*

Proof. Let $H \Rightarrow C$ be a Σ-sentence of C. Let $(A_i)_{i\in I}$ be a family of Σ-models such that $A_i \models H \Rightarrow C$ for each $i \in I$ and let $\prod_{i\in I} A_i$ be the product of this family with $p_i\colon \prod_{i\in I} A_i \to A_i$ being the corresponding projections. Let us assume that $\gamma_\Sigma(\prod_{i\in I} A_i)_H = \gamma_\Sigma(\prod_{i\in I} A_i)_{\mathtt{true}}$. By considering the homomorphism $\gamma_\Sigma(p_i)$, it follows that $\gamma_\Sigma(A_i)_H = \gamma_\Sigma(A_i)_{\mathtt{true}}$ for each $i \in I$. Hence $A_i \models C$ for each $i \in I$ and by the preservation hypothesis we obtain that $\prod_{i\in I} A_i \models C$. □

Proposition 4.2. *If each sentence of $\mathcal{I}$ is preserved by sub-models then each sentence of C is preserved by sub-models too.*

Proof. Let $H \Rightarrow C$ be a Σ-sentence of C and let $A \hookrightarrow B$ be an inclusion of Σ-models such that $B \models H \Rightarrow C$. Assume that $\gamma_\Sigma(A)_H = \gamma_\Sigma(A)_{\mathtt{true}}$. We need to show that $A \models C$. By Lemma 3.1 applied to $\gamma_\Sigma(A \hookrightarrow B)$ it follows that $\gamma_\Sigma(B)_H = \gamma_\Sigma(B)_{\mathtt{true}}$, hence $B \models C$. Since C is preserved by sub-models we obtain $A \models C$. □

Definition 4.6 (Lifting direct products). *A signature morphism $\chi\colon \Sigma \to \Sigma'$* lifts direct products *when each χ-expansion A' of a product $\prod_{i\in I} A_i$ of Σ-models is a product $\prod_{i\in I} A'_i$ of χ-expansions of A'_i of A_i for each $i \in I$.*

Example 4.4. In the institutions **ALG**, **PDL**, **POA**, and **HA** each signature extension with constants lifts direct products and in **PA** each signature extension with total constants lifts direct products. Let us show this for **ALG**, the other cases being rather similar. Let $(\prod_{i\in I} A_i \xrightarrow{p_i} A_i)_{i\in I}$ be a product of (S, F)-algebras, (S, F') an extension of (S, F) with constants, and A' an expansion of $\prod_{i\in I} A_i$ to (S, F'). Then $A' = \prod_{i\in I} A'_i$ where for each new constant σ and each $i \in I$, $(A'_i)_\sigma = p_i(A'_\sigma)$.

Proposition 4.3. *In any institution let ρ' be a Σ'-sentence that is preserved by direct products and $\chi\colon \Sigma \to \Sigma'$ be a signature morphism that lifts direct products. Then the Σ-sentence $(\forall\chi)\rho'$ is also preserved by direct products.*

Proof. Let $(A_i)_{i\in I}$ be a family of Σ-models such that $A_i \models_\Sigma (\forall\chi)\rho'$ for each $i \in I$. Consider the direct product $\prod_{i\in I} A_i$ and let A' be any χ-expansion of $\prod_{i\in I} A_i$. By the lifting hypothesis on χ we have that $A' = \prod_{i\in I} A'_i$ where A'_i are χ-expansions of A_i, respectively. Then $A'_i \models_{\Sigma'} \rho'$ for each $i \in I$. Since ρ' is preserved by products it follows that $A' \models_{\Sigma'} \rho'$. □

Definition 4.7 (Lifting inclusions). *A signature morphism $\chi\colon \Sigma \to \Sigma'$* lifts inclusions *when for each inclusion homomorphism $A \hookrightarrow B$ and each χ-expansion A' of a A there exists a χ-expansion of $A \hookrightarrow B$ to an inclusion homomorphism $A' \hookrightarrow B'$.*

Example 4.5. In the institutions **ALG**, **PDL**, **POA**, and **HA** each signature extension with constants lifts inclusions and in **PA** each signature extension with total constants lifts the inclusions of the inclusion systems of Ex. 4.1. In all these cases, given an inclusion homomorphism $A \hookrightarrow B$ of Σ-models, a signature extension with constants $\chi\colon \Sigma \to \Sigma'$, and A' a χ-expansion of A, then we define B' to be the χ-expansion of B such that $B'_\sigma = A'_\sigma$ for each new constant σ.

Proposition 4.4. *In any institution let ρ' be any Σ'-sentence that is preserved by sub-models and $\chi\colon \Sigma \to \Sigma'$ be a signature morphism that lifts inclusions. Then the Σ-sentence $(\forall\chi)\rho'$ is also preserved by sub-models.*

Proof. Let $A \hookrightarrow B$ be an inclusion of Σ-models such that $B \models (\forall\chi)\rho'$. Let A' be any χ-expansion of A. Since χ lifts inclusions, there exists an inclusion $A' \hookrightarrow B'$ of Σ'-models such that $B'\restriction_\chi = B$. Hence $B' \models \rho'$. Since ρ' is preserved by sub-models we have that $A' \models \rho'$. □

The results of Propositions 4.1, 4.2, 4.4, and 4.3 are collected by the following consequence.

Corollary 4.1. *Let $\mathcal{D}$ be a class of signature morphism in $\mathcal{I}$ that satisfies the conditions of Prop. 3.1 and such that each signature morphism in $\mathcal{D}$ lifts direct products and inclusions. If each sentence of $\mathcal{I}$ is preserved by direct products and sub-models then each sentence of $\forall_{\mathcal{D}}C(\Phi,\alpha,\gamma)$ is also preserved by direct products and sub-models.*

The conclusion of Cor. 4.1 says that the class of models of any set of sentences in $\forall_{\mathcal{D}}C(\Phi,\alpha,\gamma)$ is a quasi-variety. At this moment we may apply a general result on existence of initial models in quasi-varieties. Before recalling this result in the variant presented in [4] we need also to recall a couple of technical concepts.

Definition 4.8 (Epic inclusion system). *An inclusion system is* epic *when each abstract surjection is epimorphism.*

Example 4.6. Since homomorphisms with surjective carrier functions are epimorphisms, it follows that each of the inclusion systems of Ex. 4.1 is epic.

Definition 4.9 (Co-well powered inclusion system). *Given an inclusion system, a* quotient representation *of any object A is an abstract surjection $A \to B$. A* quotient *of A is an isomorphism class of quotient representations. An inclusion system is* co-well powered *when each of its objects has a* set *of quotients.*

Example 4.7. In general all non-trivial inclusion systems in 'concrete categories', i.e. for which there exists a faithful functor to $\mathbb{S}et$, the quasi-category of sets and functions, enjoy the property of co-well-powered. This is the case of all inclusion systems of Ex. 4.1. For example, the quotients of a total algebra are in a bijective correspondence to the congruences on that algebra, which are less than the binary relations on the algebra, which obviously form a set.

The following result is well known in the general categorical approaches to quasi-varieties. The variant presented below comes from [4].

Proposition 4.5. *In any category with direct products and with a designated epic and co-well powered inclusion system, each quasi-variety has an initial object.*

Corollary 4.2. *In addition to the conditions of Cor. 4.1 we suppose that the inclusion systems of the categories of models are epic and co-well-powered. Then each set of sentences in $\forall_{\mathcal{D}}C(\Phi,\alpha,\gamma)$ admits an initial model. In particular, this property of initial semantics holds in all institutions of Ex. 3.8 apart of those derived from* **HA**.

5. Proof theoretic aspects

In this section we show how Birkhoff calculus for conditional equations can be used as a sound calculus for institutions with sentences conditioned by Boolean terms, i.e. institutions of the form $\forall_{\mathcal{D}}C(\Phi,\alpha,\gamma)$. We need the following rather technical concept that relates abstract quantifiers to the concrete quantifiers in **ALG**.

Definition 5.1 (Quantifier translatability). *Let (Φ,α,γ) be a quasi-Boolean encoding of an institution $\mathcal{I}$. A signature morphism $\chi\colon \Sigma \to \Sigma'$ in $\mathcal{I}$ is* translatable by (Φ,α,γ) *when $\Phi(\chi)$ is an extension of $\Phi(\Sigma)$ with a finite sets of variables and there exists a finite set $\Delta(\chi)$ of equations on the sort* B *of $\Phi(\Sigma')$ such that:*

- *$\gamma_{\Sigma'}(M') \models^{\mathbf{ALG}}_{\Phi(\Sigma')} \Delta(\chi)$ for each Σ'-model M', and*
- *for each Σ-model M and for each $\Phi(\chi)$-expansion A' of $\gamma_\Sigma(M)$ if $A' \models^{\mathbf{ALG}}_{\Phi(\Sigma')} \Delta(\chi)$ then there exists a χ-expansion M' of M such that $A' = \gamma_{\Sigma'}(M')$.*

Example 5.1. Let the signature Σ' extends Σ with a finite set of variables in **ALG**, **PDL**, **POA**, or in **HA**. Then the signature extension $\chi\colon\ \Sigma \to \Sigma'$ is translatable by the corresponding quasi-Boolean encodings of Examples 3.1, 3.2, 3.3, 3.5 and 3.6 by letting $\Delta(\chi)$ be empty. In the case of a signature extension with a finite set X of *total* variables in **PA**, this is translatable by the corresponding quasi-Boolean encodings of Examples 3.4 and 3.6 by letting $\Delta(\chi) = \{x \circledcirc x = \mathtt{true} \mid x \in X\}$.

Proposition 5.1. *For any quasi-Boolean encoding* (Φ, α, γ) *of an institution* $\mathcal{I}$ *and for any signature morphism* $\chi\colon\ \Sigma \to \Sigma'$ *such that*

- χ *is translatable by* (Φ, α, γ)*, and*
- $\gamma_\chi\colon\ \mathsf{Mod}(\chi); \gamma_\Sigma \Rightarrow \gamma_{\Sigma'}; \mathsf{Mod}(\Phi(\chi))$ *is identity.*

For each Σ*-model* M *and each* Σ*-sentence* $H \Rightarrow C$ *in* $C(\Phi, \alpha, \gamma)$ *we have that*

$$M \models_\Sigma (\forall\chi)H \Rightarrow C \ \text{ if and only if } \ \gamma_\Sigma(M) \models^{\mathbf{ALG}}_{\Phi(\Sigma)} (\forall X)(\bigwedge \Delta(\chi)) \wedge (H = \mathtt{true}) \Rightarrow (\alpha_{\Sigma'}(C) = \mathtt{true})$$

where X *is the finite set of variables extending* $\Phi(\Sigma)$ *to* $\Phi(\Sigma')$*.*

Proof. For showing the implication from the left to the right we consider A' any $\Phi(\chi)$-expansion of $\gamma_\Sigma(M)$ such that $A' \models^{\mathbf{ALG}}_{\Phi(\Sigma')} \bigwedge(\Delta(\chi)) \wedge (H = \mathtt{true})$. Because χ is translatable there exists a χ-expansion M' of M such that $A' = \gamma_{\Sigma'}(M')$. Hence $\gamma_{\Sigma'}(M') \models^{\mathbf{ALG}}_{\Phi(\Sigma')} (H = \mathtt{true})$. From the hypothesis it follows that $M' \models_{\Sigma'} C$ and by the the Encoding Condition for (Φ, α, γ) that $\gamma_{\Sigma'}(M') \models^{\mathbf{ALG}}_{\Phi(\Sigma')} (\alpha_{\Sigma'}(C) = \mathtt{true})$. This means $A' \models^{\mathbf{ALG}}_{\Phi(\Sigma')} (\alpha_{\Sigma'}(C) = \mathtt{true})$.

For showing the implication from the right to the left let M' be any χ-expansion of M. Let us assume that $\gamma_{\Sigma'}(M') \models^{\mathbf{ALG}}_{\Phi(\Sigma')} (H = \mathtt{true})$. Because χ is translatable we have that $\gamma_{\Sigma'}(M') \models^{\mathbf{ALG}}_{\Phi(\Sigma')} \Delta(\chi)$. Because γ_χ is identity we have that $\gamma_{\Sigma'}(M')\restriction_{\Phi(\chi)} = \gamma_\Sigma(M'\restriction_\chi) = \gamma_\Sigma(M)$. By applying the hypothesis it follows that $\gamma_{\Sigma'}(M') \models^{\mathbf{ALG}}_{\Phi(\Sigma')} \alpha_{\Sigma'}(C) = \mathtt{true}$ which by the Encoding Condition for (Φ, α, β) means $M' \models_{\Sigma'} C$. □

Note that when χ is a signature extension with constants in **ALG**, **PDL**, **POA**, or **HA**, and with total constants in **PA**, for all quasi-Boolean encodings of Ex. 3.1-3.6 we have that γ_χ is identity.

Notation 5.1. *Within the framework of Prop. 5.1 by* $\alpha^*_{\Sigma'}((\forall\chi)H \Rightarrow C)$ *let us denote the conditional* $\Phi(\Sigma)$*-equation* $(\forall X)(\bigwedge \Delta(\chi)) \wedge (H = \mathtt{true}) \Rightarrow (\alpha_{\Sigma'}(C) = \mathtt{true})$.

The following is an immediate consequence of Prop. 5.1.

Corollary 5.1. *Let* (Φ, α, γ) *be a truth injective quasi-Boolean encoding of an institution* $\mathcal{I}$ *such that:*

- $\mathcal{I}$ *has pushouts of signatures and weak model amalgamation, and*
- $\mathcal{I}$ *has a designated class* $\mathcal{D}$ *of signature morphisms satisfying the conditions of Prop. 3.1 such that*
- *each morphism* χ *in* $\mathcal{D}$ *is translatable by* (Φ, α, γ) *and* γ_χ *is identity.*

Then for any sets E *and* E' *of* Σ*-sentences in* $\forall_{\mathcal{D}}C(\Phi, \alpha, \gamma)$ *and for each set* Γ *of conditional* $\Phi(\Sigma)$*-equations that is satisfied by* $\gamma_\Sigma(M)$ *for any* Σ*-model* M*, we have that*

$$\alpha^*_\Sigma(E) \cup \Gamma \models^{\mathbf{ALG}}_{\Phi(\Sigma)} \alpha^*_\Sigma(E') \ \text{ implies } \ E \models^{\forall_{\mathcal{D}}C(\Phi,\alpha,\gamma)}_\Sigma E'.$$

Since $\alpha^*_\Sigma(E)$, $\alpha^*_\Sigma(E')$, and Γ are sets of conditional equations, by the soundness of Birkhoff calculus for conditional equations we may replace the semantic consequence relation $\models^{\mathbf{ALG}}_{\Phi(\Sigma)}$ in Cor. 5.1 by the entailment relation $\vdash^{\mathrm{eq}}_{\Phi(\Sigma)}$ determined by Birkhoff calculus.[3] In this way, Birkhoff calculus for conditional equations, with all its rather developed execution techniques, such as rewriting, may serve as a sound calculus for the institutions $\forall_{\mathcal{D}}C(\Phi, \alpha, \gamma)$. The provability power of this import of equational calculus depends on choosing Γ as complete as possible. This idea is illustrated by the following couple of examples.

[3] Thus $\vdash^{\mathrm{eq}}$ are the least entailment relations that contain the well known rules of Reflexivity, Symmetry, Transitivity, Congruence and Substitutivity and are closed under Modus Ponens and Generalization.

Example 5.2. Let us show how deduction in **PA** can be performed by ordinary Birkhoff calculus for conditional equations by using the quasi-Boolean encoding of Ex. 3.4. For each **PA** signature (S, TF, PF) let $\Gamma_{(S,TF,PF)}$ be the following set of conditional $\Phi(S, TF, PF)$-equations:

1. $(\forall X)(X \circledcirc X = \texttt{true}) \Rightarrow (\sigma(X) \circledcirc \sigma(X) = \texttt{true})$ for any operation symbol $\sigma \in TF$.[4]
2. $(\forall X, Y)(X \circledcirc Y = \texttt{true}) \Rightarrow (X \circledcirc X = \texttt{true})$.
3. $(\forall X, Y)(X \circledcirc Y = \texttt{true}) \Rightarrow (X = Y)$.
4. $(\forall X)(\sigma(X) \circledcirc \sigma(X) = \texttt{true}) \Rightarrow (X \circledcirc X = \texttt{true})$ for any operation symbol σ in TF or in PF.

One may easily check that $\gamma_{(S,TF,PF)}(A) \models^{\mathbf{ALG}}_{\Phi(S,TF,PF)} \Gamma_{(S,TF,PF)}$ for each partial (S, TF, PF)-algebra A; we omit this check here.

Now let us consider a concrete **PA** signature with only one sort s, three unary partial operation symbols $\tau, \sigma_1, \sigma_2 : s \to s$, and one partial constant symbol $a : \to s$ and let us prove the following deduction

$$\{\tau(a) \overset{e}{=} \tau(a),\ (\forall x)\sigma_1(x) \overset{e}{=} \sigma_2(x)\} \models^{\mathbf{PA}} \{\sigma_1(a) \overset{e}{=} \sigma_2(a)\} \tag{4}$$

by using ordinary Birkhoff calculus for conditional equations. Since the equations involved in deduction (4) are unconditional this can be considered a deduction in $\forall_{\mathcal{D}}(\Phi, \alpha, \gamma)$ where (Φ, α, γ) is the quasi-Boolean encoding of Ex. 3.4 and $\mathcal{D}$ is the class of **PA** signature extensions with finite sets of total variables. By Cor. 5.1 and by the soundness of ordinary Birkhoff calculus for conditional equations it is sufficient to prove that

$$\{\tau(a) \circledcirc \tau(a) = \texttt{true},\ (\forall x)(x \circledcirc x = \texttt{true}) \Rightarrow (\sigma_1(x) \circledcirc \sigma_2(x) = \texttt{true})\} \cup \Gamma_{(S,TF,PF)} \vdash^{\mathrm{eq}}$$

$$\sigma_1(a) \circledcirc \sigma_2(a) = \texttt{true} \tag{5}$$

From the 4th axiom scheme in $\Gamma_{(S,TF,PF)}$ applied for τ, by the rule of Substitutivity applied for the substitution $x \mapsto a$ we have

$$\Gamma_{(S,TF,PF)} \vdash^{\mathrm{eq}} (\tau(a) \circledcirc \tau(a) = \texttt{true}) \Rightarrow (a \circledcirc a = \texttt{true}) \tag{6}$$

From (6) by Modus Ponens we obtain

$$\Gamma_{(S,TF,PF)} \cup \{\tau(a) \circledcirc \tau(a) = \texttt{true}\} \vdash^{\mathrm{eq}} a \circledcirc a = \texttt{true} \tag{7}$$

From (5) by Substitutivity applied for $x \mapsto a$ we have

$$(\forall x)(x \circledcirc x = \texttt{true}) \Rightarrow (\sigma_1(x) \circledcirc \sigma_2(x) = \texttt{true}) \vdash^{\mathrm{eq}} (a \circledcirc a = \texttt{true}) \Rightarrow (\sigma_1(a) \circledcirc \sigma_2(a) = \texttt{true}) \tag{8}$$

Then (5) is obtained by Modus Ponens from (7) and (8).

For those readers that are familiar with the institution theoretic concept of 'persistently liberal simple theoroidal comorphism of institutions', we may note that within the context of Ex. 5.2 the mapping of (S, TF, PF) to $(\Phi(S, TF, PF), \Gamma_{(S,TF,PF)})$ is part of a 'persistently liberal' simple theoroidal comorphism **PA** $\to$ **ALG** with $\gamma_{(S,TF,PF)}$ being the left adjoints to the model translation functors of this comorphism. More details about this can be found in [5]. More on the concept of persistently liberal theoroidal comorphisms can be found in [4, 16, 22, 23]. In fact all quasi-Boolean encodings of Examples 3.1, 3.2, 3.3, 3.4, and 3.5 (but not those of Ex. 3.6) can be presented as persistently liberal simple theoroidal comorphisms, and this gives a general method for chosing Γ applicable to many concrete situations.

The following simple example shows how ordinary Birkhoff calculus for conditional equations can be used to perform deductions with sentences conditioned by Boolean terms, that in the conventional unencoded framework are not Horn sentences.

---

[4] If $X = \{x_1, \ldots, x_n\}$ then $X \circledcirc X$ denotes the finite conjunction $(x_1 \circledcirc x_1) \wedge \ldots \wedge (x_n \circledcirc x_n)$.

Example 5.3. Within the framework of Ex. 3.6 applied to **PDL** (i.e. Ex. 3.2) let us consider a **PDL** signature Σ that has two predicate symbols p and q with arity zero. Let Γ_Σ be the set of Boolean algebra equations on the sort $\mathtt{B}$. Cf. Ex. 3.6 it is clear that for any Σ-model M we have that $\gamma_\Sigma(M) \models^{\mathbf{ALG}} \Gamma_\Sigma$. Let us show that

$$\{p,\ \textcircled{p} \textcircled{\vee} \textcircled{q} \ \Rightarrow q\} \models q.$$

By Cor. 5.1 and by the soundness of ordinary Birkhoff calculus for conditional equations it is sufficient to show that

$$\Gamma_\Sigma \cup \{\textcircled{p} = \mathtt{true},\ (\textcircled{p} \textcircled{\vee} \textcircled{q} = \mathtt{true}) \Rightarrow (\textcircled{q} = \mathtt{true})\} \vdash^{\mathrm{eq}} \textcircled{q} = \mathtt{true}. \tag{9}$$

By the rules of Reflexivity and Congruence we have that

$$\textcircled{p} \ = \mathtt{true} \vdash^{\mathrm{eq}} \textcircled{p} \textcircled{\vee} \textcircled{q} \ = \mathtt{true} \textcircled{\vee} \textcircled{q}\ . \tag{10}$$

We also have

$$\Gamma_\Sigma \vdash^{\mathrm{eq}} \mathtt{true} \textcircled{\vee} \textcircled{q} \ = \mathtt{true}. \tag{11}$$

From (10) and (11) by the rule of Transitivity we obtain that

$$\Gamma_\Sigma \cup \{\textcircled{p} \ = \mathtt{true}\} \vdash^{\mathrm{eq}} \textcircled{p} \textcircled{\vee} \textcircled{q} \ = \mathtt{true}. \tag{12}$$

Finally, (9) is obtained from (12) by Modus Ponens.

Acknowledgement. The author is grateful to the anonymous referees for their suggestions that have led to several improvements in the original draft of this work. A technical remark of one of them implied a series of significant technical adjustments.

6. Conclusions and Future Research

Based upon the novel concept of quasi-Boolean encoding that we have introduced in this paper, we have provided rigorous logical foundations for the formal specification and verification practice of using sentences conditioned by Boolean-valued terms. The generality of our institution theoretic approach leads to wide applicability of our results to various logic based specification environments. The main results of our paper include a general theorem on existence of initial semantics and a general result allowing the import of ordinary Birkhoff calculus for conditional equations as a sound proof calculus for institutions with universally quantified sentences conditioned by Boolean terms.

The work developed in this paper leads to a series of open problems as follows:

1. Find a general set of sufficient conditions with good applicability for importing ordinary Birkhoff calculus for conditional equations as a complete proof calculus for the institutions $\forall C(\Phi, \alpha, \gamma)$.
2. Study of important model theoretic properties of the institutions $\forall C(\Phi, \alpha, \gamma)$ that are most relevant for specification theory, such as interpolation and definability.
3. Use the theoretical framework introduced here to provide clear full foundations for the OTS/CafeOBJverification method, and extract a series of methodological guidelines supporting and correcting the current practice involved in the OTS/CafeOBJverification method.

References

[1] Peter Burmeister. *A Model Theoretic Oriented Approach to Partial Algebras*. Akademie-Verlag, Berlin, 1986.
[2] 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.
[3] Răzvan Diaconescu. The logic of Horn clauses is equational. Technical Report PRG-TR-3-93, Programming Research Group, University of Oxford, 1990.
[4] Răzvan Diaconescu. *Institution-independent Model Theory*. Birkhäuser, 2008.
[5] Răzvan Diaconescu. An encoding of partial algebras as total algebras. *Information Processing Letters*, 2009. DOI:10.1016/j.ipl.2009.09.008. To appear in print.
[6] 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.

[7] Răzvan Diaconescu and Kokichi Futatsugi. Behavioural coherence in object-oriented algebraic specification. *Universal Computer Science*, 6(1):74–96, 2000. First version appeared as JAIST Technical Report IS-RR-98-0017F, June 1998.
[8] Răzvan Diaconescu and Kokichi Futatsugi. Logical foundations of CafeOBJ. *Theoretical Computer Science*, 285:289–318, 2002.
[9] 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.
[10] Kokichi Futatsugi, Joseph Goguen, and Kazuhiro Ogata. Verifying design with proof scores. In Bertrand Meyer and Jim Woodcock, editors, *Verified Software: Theories, Tools, Experiments*, volume 4171 of *Lecture Notes in Computer Science*, pages 227–290. Springer, 2008.
[11] Joseph Goguen. Types as theories. In George Michael Reed, Andrew William Roscoe, and Ralph F. Wachter, editors, *Topology and Category Theory in Computer Science*, pages 357–390. Oxford, 1991. Proceedings of a Conference held at Oxford, June 1989.
[12] 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.
[13] Joseph Goguen and Răzvan Diaconescu. Towards an algebraic semantics for the object paradigm. In Harmut Ehrig and Fernando Orejas, editors, *Recent Trends in Data Type Specification*, volume 785 of *Lecture Notes in Computer Science*, pages 1–34. Springer, 1994.
[14] Joseph Goguen and José Meseguer. Eqlog: Equality, types, and generic modules for logic programming. In Douglas DeGroot and Gary Lindstrom, editors, *Logic Programming: Functions, Relations and Equations*, pages 295–363. Prentice-Hall, 1986.
[15] 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.
[16] Joseph Goguen and Grigore Roşu. Institution morphisms. *Formal Aspects of Computing*, 13:274–307, 2002.
[17] Joseph Goguen, Timothy Winkler, José Meseguer, Kokichi Futatsugi, and Jean-Pierre Jouannaud. Introducing OBJ. In Joseph Goguen and Grant Malcolm, editors, *Software Engineering with OBJ: algebraic specification in action*. Kluwer, 2000.
[18] Rolf Hennicker and Michel Bidoit. Observational logic. In A. M. Haeberer, editor, *Algebraic Methodology and Software Technology*, number 1584 in LNCS, pages 263–277. Springer, 1999. Proc. AMAST'99.
[19] Saunders Mac Lane. *Categories for the Working Mathematician*. Springer, second edition, 1998.
[20] José Meseguer. General logics. In H.-D. Ebbinghaus et al., editors, *Proceedings, Logic Colloquium, 1987*, pages 275–329. North-Holland, 1989.
[21] José Meseguer. Rewriting as a unified model of concurrency. In *Proceedings, Concur'90 Conference*, Lecture Notes in Computer Science, Volume 458, pages 384–400, Amsterdam, August 1990. Springer.
[22] Till Mossakowski. Relating CASL with other specification languages: the institution level. *Theoretical Computer Science*, 286:367–475, 2002.
[23] Till Mossakowski. Heterogeneous specification and the heterogeneous tool set. Habilitation thesis, University of Bremen, 2005.
[24] Kazuhiro Ogata and Kokichi Futatsugi. Simulation-based verification for invariant properties in the OTS/CafeOBJ method. *Electronic Notes in Theoretical Computer Science*, 201:127–154, 2008.
[25] Grigore Roşu. *Hidden Logic*. PhD thesis, University of California at San Diego, 2000.
[26] Donald Sannella and Andrzej Tarlecki. Specifications in an arbitrary institution. *Information and Control*, 76:165–210, 1988.
[27] Andrzej Tarlecki. On the existence of free models in abstract algebraic institutions. *Theoretical Computer Science*, 37:269–304, 1986.
[28] Andrzej Tarlecki. Quasi-varieties in abstract algebraic institutions. *Journal of Computer and System Sciences*, 33(3):333–360, 1986.