*Math. Struct. in Comp. Science* (2014), vol. 24, iss. 2, e240205, 37 pages. 
doi:10.1017/S0960129513000212

# Property-oriented semantics of structured specifications†

DONALD SANNELLA‡ and ANDRZEJ TARLECKI§

‡*Laboratory for Foundations of Computer Science, University of Edinburgh, Edinburgh, United Kingdom*
*Website:* homepages.inf.ed.ac.uk/dts/
§*Institute of Informatics, University of Warsaw, Warsaw, Poland*
*Website:* www.mimuw.edu.pl/∼tarlecki/

*Received 27 July 2012; revised 6 April 2013*

We consider structured specifications built from flat specifications using union, translation and hiding with their standard model-class semantics in the context of an arbitrary institution. We examine the alternative of sound property-oriented semantics for such specifications, and study their relationship to model-class semantics. An exact correspondence between the two (completeness) is not achievable in general. We show through general results on property-oriented semantics that the semantics arising from the standard proof system is the strongest sound and compositional property-oriented semantics in a wide class of such semantics. We also sharpen one of the conditions that does guarantee completeness and show that it is a necessary condition.

## 1. Introduction

Specification formalisms offer *specification-building operations* that can be used to build complex structured specifications by combining and extending simpler ones (Burstall and Goguen 1977). Then, an understanding of a large specification is achieved through an understanding of its components. The meaning of a specification formalism needs to be completely and precisely defined, and this raises the question of what specifications should denote. The ultimate role of any specification is to describe the class of behaviours that satisfy the specification – its *models*, in logical terminology – and hence are to be regarded as correct for the task at hand. In algebraic specification, we represent behaviours as *algebras*, abstracting away from the details of code and algorithms used to implement behaviours. It is then natural to take the class of algebras that represent correct behaviours – its *model class* – as the semantics of a specification. This view carries over to the framework of an arbitrary logical system formalised as an *institution* (Goguen and Burstall 1992), where algebras may be replaced by other semantic structures appropriate for modelling the behaviours of the programs at hand.

However, while model-class semantics remains fundamental, it is vital to be able to determine whether or not a given property is a consequence of a given specification,

† The work reported in this paper was partially supported by the Polish Ministry of Science and Higher Education, grant N206 493138 (AT).

that is, holds in all of its models. This is the purpose of proof systems for consequences of structured specifications, as given, for instance, in Sannella and Tarlecki (1988). The essential property of such a system is its *soundness*, which ensures that the consequences derived from a specification do indeed hold in all of its models. Another key property is that the proof system is *compositional*, so that the consequences of a structured specification are derived from the consequences of its immediate constituents. This allows consequences of structured specifications to be deduced in stages, with the structure of the specification as a guide to the 'shape' of the proof. *Completeness* holds when every property that holds in all of the models of a specification is always derivable; this is highly desirable, but rarely achievable in the practical context of specification formalisms, which often provide the means for defining the standard model of the natural numbers and other datatypes (MacQueen and Sannella 1985).

A proof system for consequences of structured specifications determines an alternative 'property-oriented' semantics for specifications that maps them to sets of properties (or theories), as in the original semantics of the Clear specification language – see Burstall and Goguen (1980); see also Diaconescu *et al.* (1993). Requiring the proof system to be sound amounts to requiring that the properties given by this semantics hold in all of the models given by the model-class semantics. The compositionality requirement amounts to requiring the meaning of a structured specification in the property-oriented semantics to depend functionally on the meanings of its immediate constituents. Completeness, together with soundness, means that the two forms of semantics essentially coincide.

Sound and compositional property-oriented semantics are the subject of study in this paper, which we conduct in the context of an arbitrary institution (Goguen and Burstall 1992). We will begin by recalling the standard property-oriented semantics for structured specifications built from flat specifications using union, translation and hiding, originating from Clear (Burstall and Goguen 1980), with the model-class semantics given in Sannella and Tarlecki (1988). We then recall some existing results concerning the completeness of this semantics and its corresponding proof system, and in the process sharpen one of the conditions that guarantee completeness and show that it is, moreover, a necessary condition. The semantics is only complete when the logic in use admits interpolation, so, for instance, there is a 'gap' between the model class semantics and the property-oriented semantics for many-sorted equational specifications (unless we impose strong restrictions on the algebras and morphisms involved).

A new result is that the standard property-oriented semantics (and its corresponding proof system) cannot be improved: no sound and compositional semantics can be better. This is a consequence of a more general result we prove concerning property-oriented semantics for structured specifications built using any collection of specification-building operations. Surprisingly, this result requires a mild but unexpected technical assumption: that the semantics considered must not 'forget' any of the axioms present in flat specifications. We first show this under the assumption that an oracle (that is, a complete proof system) for semantic consequences of any set of axioms in the underlying logic is given and used in the semantics to close the sets of properties assigned to specifications under consequence. Then we show that this assumption may be dropped when a stronger form of compositionality is assumed. Finally, we show how these results carry over to the

https://doi.org/10.1017/S0960129513000212 Published online by Cambridge University Press

context of a sound but not necessarily complete proof system for the underlying logic, given as an entailment system.

## 2. Institutional preliminaries

Following Goguen and Burstall (1992) and Sannella and Tarlecki (1988), we abstract away from any particular logical system and study specifications built in an arbitrary logical system formalised as an institution.

An *institution* (Goguen and Burstall 1992) **INS** consists of:

— a category

$$\mathbf{Sign}_{\mathbf{INS}}$$

of *signatures*;

— a functor

$$\mathbf{Sen}_{\mathbf{INS}} : \mathbf{Sign}_{\mathbf{INS}} \to \mathbf{Set},$$

giving a set $\mathbf{Sen}_{\mathbf{INS}}(\Sigma)$ of $\Sigma$-*sentences* for each signature $\Sigma \in |\mathbf{Sign}_{\mathbf{INS}}|$;

— a functor

$$\mathbf{Mod}_{\mathbf{INS}} : \mathbf{Sign}_{\mathbf{INS}}^{op} \to \mathbf{Cat},$$

giving a category $\mathbf{Mod}_{\mathbf{INS}}(\Sigma)$ of $\Sigma$-*models* for each signature $\Sigma \in |\mathbf{Sign}_{\mathbf{INS}}|$; and

— a family

$$\langle \models_{\mathbf{INS},\Sigma} \subseteq |\mathbf{Mod}_{\mathbf{INS}}(\Sigma)| \times \mathbf{Sen}_{\mathbf{INS}}(\Sigma) \rangle_{\Sigma \in |\mathbf{Sign}_{\mathbf{INS}}|}$$

of *satisfaction relations*

such that for any signature morphism $\sigma : \Sigma \to \Sigma'$ the translations $\mathbf{Mod}_{\mathbf{INS}}(\sigma)$ of models and $\mathbf{Sen}_{\mathbf{INS}}(\sigma)$ of sentences preserve the satisfaction relation, that is, for any $\varphi \in \mathbf{Sen}_{\mathbf{INS}}(\Sigma)$ and $M' \in |\mathbf{Mod}_{\mathbf{INS}}(\Sigma')|$ the following *satisfaction condition* holds:

$$M' \models_{\mathbf{INS},\Sigma'} \mathbf{Sen}_{\mathbf{INS}}(\sigma)(\varphi) \quad \text{iff} \quad \mathbf{Mod}_{\mathbf{INS}}(\sigma)(M') \models_{\mathbf{INS},\Sigma} \varphi.$$

Examples of institutions abound. The institution **EQ** of equational logic has many-sorted algebraic signatures as signatures, many-sorted algebras as models and (explicitly quantified) equations as sentences. The institution **FOPEQ** of first-order predicate logic with equality has signatures that add predicate names to many-sorted algebraic signatures, models that extend algebras by interpreting predicate names as relations and sentences that are all closed (no free variables) formulae of first-order logic with equality. The institution $\mathbf{L}_{\omega_1\omega}$ then extends **FOPEQ** by permitting infinitely countable disjunction and conjunction in formulae. See Sannella and Tarlecki (2012) for detailed definitions of these and many other institutions. We will also consider single-sorted versions of these institutions ($\mathbf{EQ}^{ss}$, and so on), as well as versions where models are required to have non-empty carriers of all sorts ($\mathbf{EQ}^{ne}$, and so on).

We will make free use of standard terminology, and say that a $\Sigma$-model $M$ *satisfies* a $\Sigma$-sentence $\varphi$, or that $\varphi$ *holds* in $M$, whenever $M \models_{\mathbf{INS},\Sigma} \varphi$. We will omit the subscript **INS**, writing

$$\mathbf{INS} = \langle \mathbf{Sign}, \mathbf{Sen}, \mathbf{Mod}, \langle \models_\Sigma \rangle_{\Sigma \in |\mathbf{Sign}|} \rangle.$$

https://doi.org/10.1017/S0960129513000212 Published online by Cambridge University Press

Similarly, the subscript $\Sigma$ on the satisfaction relations will often be omitted. For any signature morphism $\sigma : \Sigma \to \Sigma'$, the translation function

$$\mathbf{Sen}(\sigma) : \mathbf{Sen}(\Sigma) \to \mathbf{Sen}(\Sigma')$$

will be denoted by

$$\sigma : \mathbf{Sen}(\Sigma) \to \mathbf{Sen}(\Sigma'),$$

the coimage function with respect to $\mathbf{Sen}(\sigma)$ by

$$\sigma^{-1} : \mathcal{P}(\mathbf{Sen}(\Sigma')) \to \mathcal{P}(\mathbf{Sen}(\Sigma)),$$

and the reduct functor

$$\mathbf{Mod}(\sigma) : \mathbf{Mod}(\Sigma') \to \mathbf{Mod}(\Sigma)$$

by

$$\_|_\sigma : \mathbf{Mod}(\Sigma') \to \mathbf{Mod}(\Sigma).$$

Thus, the satisfaction condition can be re-stated in the form

$$M' \models \sigma(\varphi) \quad \text{iff} \quad M'|_\sigma \models \varphi.$$

From now on, we will work with an arbitrary but fixed institution

$$\mathbf{INS} = \langle \mathbf{Sign}, \mathbf{Sen}, \mathbf{Mod}, \langle \models_\Sigma \rangle_{\Sigma \in |\mathbf{Sign}|} \rangle.$$

For any signature $\Sigma$, the satisfaction relation extends naturally to sets of $\Sigma$-sentences and classes of $\Sigma$-models. In other words, for any set $\Phi \subseteq \mathbf{Sen}(\Sigma)$ of $\Sigma$-sentences and model $M \in |\mathbf{Mod}(\Sigma)|$, we write $M \models \Phi$ to mean $M \models \varphi$ for all $\varphi \in \Phi$. Then, for any $\Sigma$-sentence $\varphi \in \mathbf{Sen}(\Sigma)$ and class $\mathcal{M} \subseteq |\mathbf{Mod}(\Sigma)|$ of $\Sigma$-models, $\mathcal{M} \models \varphi$ means $M \models \varphi$ for all $M \in \mathcal{M}$. Finally, we will also write $\mathcal{M} \models \Phi$ with the obvious meaning.

Given a class of $\Sigma$-models $\mathcal{M} \subseteq |\mathbf{Mod}(\Sigma)|$, its *theory* is given by

$$Th(\mathcal{M}) = \{\varphi \in \mathbf{Sen}(\Sigma) \mid \mathcal{M} \models \varphi\}.$$

Given a set of $\Sigma$-sentences $\Phi \subseteq \mathbf{Sen}(\Sigma)$, the class of its *models* is given by

$$Mod(\Phi) = \{M \in |\mathbf{Mod}(\Sigma)| \mid M \models \Phi\}.$$

For any signature $\Sigma$, a $\Sigma$-sentence $\varphi \in \mathbf{Sen}(\Sigma)$ is a *semantic consequence* of a set of $\Sigma$-sentences $\Phi \subseteq \mathbf{Sen}(\Sigma)$ (written $\Phi \models_\Sigma \varphi$ or simply $\Phi \models \varphi$) if for all $\Sigma$-models $M \in |\mathbf{Mod}(\Sigma)|$, we have $M \models \varphi$ whenever $M \models \Phi$.

It is trivial to check that for any class of $\Sigma$-models $\mathcal{M} \subseteq |\mathbf{Mod}(\Sigma)|$, its theory $Th(\mathcal{M})$ is closed under semantic consequence, and that if a set $\Phi \subseteq \mathbf{Sen}(\Sigma)$ is closed under semantic consequence ($\Phi \models \varphi$ implies $\varphi \in \Phi$), then it is a theory of its model class. We write $Cl_\Sigma(\Phi)$ for the *closure* of $\Phi$ under semantic consequence: that is,

$$Cl_\Sigma(\Phi) = Th(Mod(\Phi)).$$

Translation under signature morphisms preserves semantic consequence: that is, for any $\sigma : \Sigma \to \Sigma'$, $\varphi \in \mathbf{Sen}(\Sigma)$ and $\Phi \subseteq \mathbf{Sen}(\Sigma)$, we have

$$\text{if } \Phi \models \varphi, \text{ then } \sigma(\Phi) \models \sigma(\varphi).$$

The opposite implication may fail in general. However, it does hold, for instance, if the reduct

$$\_|_\sigma : \mathbf{Mod}(\Sigma') \to \mathbf{Mod}(\Sigma)$$

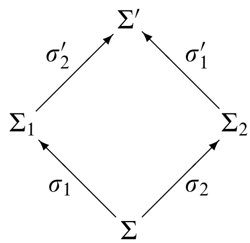
is surjective on models, so in that case we do have

$$\Phi \models \varphi \quad \text{iff} \quad \sigma(\Phi) \models \sigma(\varphi).$$

Consequently, semantic consequence is (preserved and) reflected by translation under all signature morphisms that are injective in $\mathbf{EQ}^{ne}$, $\mathbf{FOPEQ}^{ne}$ and $\mathbf{L}^{ne}_{\omega_1\omega}$ (since in these institutions, injective morphisms induce surjective reduct functors).

The institutional structure is rich enough to enable a number of key features of logical systems to be expressed. For instance, amalgamation and interpolation properties may be captured as follows.

Consider the following commuting diagram in **Sign**:




This diagram *admits amalgamation* if for any two models $M_1 \in |\mathbf{Mod}(\Sigma_1)|$ and $M_2 \in |\mathbf{Mod}(\Sigma_2)|$ such that

$$M_1|_{\sigma_1} = M_2|_{\sigma_2},$$

there exists a unique model $M' \in |\mathbf{Mod}(\Sigma')|$ such that

$$M'|_{\sigma'_2} = M_1$$
$$M'|_{\sigma'_1} = M_2$$

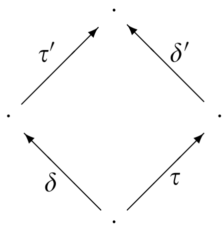
(we call such an $M'$ the *amalgamation* of $M_1$ and $M_2$), and similarly for model morphisms.

An institution is *semi-exact* if pushouts of signature morphisms always exist and admit amalgamation (or, equivalently, $\mathbf{Mod} : \mathbf{Sign}^{op} \to \mathbf{Cat}$ translates them to pullbacks in **Cat**). In fact, the developments below do not depend on the amalgamation properties for model morphisms, so semi-exactness is a bit too strong for our needs. However, we will use this standard notion nonetheless since we are not aware of any example of an institution of practical importance where pushout diagrams admit amalgamation of models but not of morphisms.

Another way to weaken the requirement of exactness is to drop the uniqueness of the amalgamation; in fact, the results below still hold if we require institutions to be *weakly semi-exact*, that is, map the pushouts considered in the category of signatures to *weak* pullbacks in **Cat**. Again, we will not adopt this possible generalisation since amalgamation, rather than weak amalgamation, is a crucial property of 'useful' logical systems that enables the systematic combination of models (that represent programs) in architectural designs (Sannella and Tarlecki 2012).

It is well known that **EQ**, **FOPEQ** and $\mathbf{L}_{\omega_1\omega}$ (and their single-sorted versions) are semi-exact. But this fails for some other institutions of interest, where it is useful to rely on a slightly more subtle notion, which is parameterised by additional classes of signature morphisms.

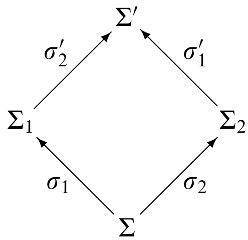
Consider two classes $\mathcal{H}, \mathcal{W} \subseteq |\mathbf{Sign}|$ of signature morphisms[†]. Then **INS** is $\langle \mathcal{H}, \mathcal{W} \rangle$-*exact* if for any signature morphisms $\delta \in \mathcal{H}$ and $\tau \in \mathcal{W}$ with a common source, there are $\delta' \in \mathcal{H}$ and $\tau' \in \mathcal{W}$ forming a pushout




that admits amalgamation, and then any such pushout admits amalgamation as well.

In the following we will always assume that $\mathcal{H}$ and $\mathcal{W}$ form wide subcategories of **Sign** (that is, are closed under composition and contain all identities in **Sign**), and that $\mathcal{H} \subseteq \mathcal{W}$.

Consider again the following commuting diagram in **Sign**:




This diagram *admits parameterised* (or *Craig–Robinson*) *interpolation* if for any $\Phi_1 \subseteq \mathbf{Sen}(\Sigma_1)$, $\Phi_2 \subseteq \mathbf{Sen}(\Sigma_2)$ and $\varphi_2 \in \mathbf{Sen}(\Sigma_2)$, whenever

$$\sigma_2'(\Phi_1) \cup \sigma_1'(\Phi_2) \models \sigma_1'(\varphi_2),$$

we have for some $\Phi \subseteq \mathbf{Sen}(\Sigma)$ such that $\Phi_1 \models \sigma_1(\Phi)$ that

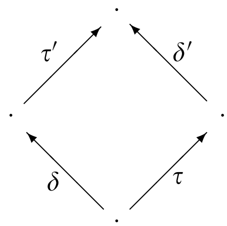
$$\Phi_2 \cup \sigma_2(\Phi) \models \varphi_2.$$

Such a set $\Phi$ will be called a *set of interpolants* for $\Phi_1$ and $\varphi_2$ with respect to $\Phi_2$. The diagram *admits Craig interpolation* if it admits parameterised interpolation with 'parameter set' $\Phi_2 = \varnothing$.

[†] In the context of structured specifications (see Section 3), morphisms in $\mathcal{H}$ will be used for **hide** (hiding), while those in $\mathcal{W}$ will be used for **with** (translation).

https://doi.org/10.1017/S0960129513000212 Published online by Cambridge University Press

Given classes $\mathcal{H}, \mathcal{W} \subseteq \mathbf{Sign}$ of signature morphisms, we say that **INS** *admits parameterised* (respectively, *Craig*) *$\langle \mathcal{H}, \mathcal{W} \rangle$-interpolation* if for any signature morphisms $\delta \in \mathcal{H}$ and $\tau \in \mathcal{W}$ with a common source there are $\delta' \in \mathcal{H}$ and $\tau' \in \mathcal{W}$ forming a pushout

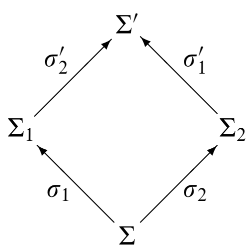



that admits parameterised (respectively, Craig) interpolation. Then any such pushout admits parameterised (respectively, Craig) interpolation as well.

Tarlecki (1986) provided the basis for the reformulation of classical (first-order) Craig interpolation (Chang and Keisler 1990) given above. We relax the requirement that the interpolant be given by a single formula, since a set of interpolants is more natural, for instance, for equational logic, as argued in Rodenburg (1991) and Diaconescu *et al.* (1993). It is well known that single-sorted first-order predicate logic with equality, $\mathbf{FOPEQ}^{ss}$, admits Craig as well as parameterised interpolation, but in the many-sorted case, interpolation requires additional assumptions on the signature morphisms involved: **FOPEQ** admits Craig and parameterised $\langle \mathcal{H}, \mathcal{W} \rangle$-interpolation when all morphisms in $\mathcal{H}$ are injective on sorts – see Borzyszkowski (2005).

Interpolation properties for equational logic are a bit more delicate. $\mathbf{EQ}^{ne}$ admits Craig $\langle \mathcal{H}, \mathcal{W} \rangle$-interpolation for classes $\mathcal{H}$ and $\mathcal{W}$ where all morphisms are injective, but the restriction to non-empty carriers cannot be dropped (Roşu and Goguen 2000; Tarlecki 2011). Parameterised $\langle \mathcal{H}, \mathcal{W} \rangle$-interpolation for $\mathbf{EQ}^{ne}$ fails (a counterexample may be extracted from Example 4.3 below, *cf.* Proposition 4.5) unless injectivity and strong 'encapsulation' properties are imposed on the morphisms in $\mathcal{H}$ (Diaconescu 2008).

In the framework of first-order predicate logic, it is easy to derive the (stronger) parameterised interpolation property from Craig interpolation. This relies on compactness and closure of the set of first-order sentences under conjunction and implication as follows. Consider the commuting diagram




in the category of first-order signatures, and assume that it admits Craig interpolation. Let $\Phi_1 \subseteq \mathbf{Sen}(\Sigma_1)$, $\Phi_2 \subseteq \mathbf{Sen}(\Sigma_2)$ and $\varphi \in \mathbf{Sen}(\Sigma_2)$ be such that

$$\sigma_2'(\Phi_1) \cup \sigma_1'(\Phi_2) \models \sigma_1'(\varphi).$$

https://doi.org/10.1017/S0960129513000212 Published online by Cambridge University Press

By compactness, there are finite $\Psi_1 \subseteq \Phi_1$ and $\Psi_2 \subseteq \Phi_2$ such that

$$\sigma_2'(\Psi_1) \cup \sigma_1'(\Psi_2) \models \sigma_1'(\varphi).$$

Then

$$\sigma_2'(\Psi_1) \models \sigma_1'\left(\bigwedge \Psi_2 \Rightarrow \varphi\right),$$

so, by the simple Craig interpolation property, we have a set $\Psi$ of $\Sigma$-sentences such that

$$\Psi_1 \models \sigma_1(\Psi)$$
$$\sigma_2(\Psi) \models \left(\bigwedge \Psi_2 \Rightarrow \varphi\right).$$

Hence, we also have

$$\Phi_1 \models \sigma_1(\Psi)$$
$$\Phi_2 \cup \sigma_2(\Psi) \models \varphi,$$

so $\Psi$ is an interpolant set for $\Phi_1$ and $\varphi$ with respect to $\Phi_2$. Although this argument may be generalised to any institution where implication and 'sufficiently large' conjunction are expressible, in general, parameterised interpolation is properly stronger than Craig interpolation.

Just as in classical first-order logic, where interpolation is sometimes derived from the Robinson consistency theorem, or from various conservativity properties, similar relationships hold between analogous notions in the institutional framework. For instance, we say that $\sigma_1 : \Sigma \to \Sigma_1$ is *conservative* for $\Phi_1 \subseteq \mathbf{Sen}(\Sigma_1)$[†] if

$$Mod(\sigma_1^{-1}(Cl_{\Sigma_1}(\Phi_1))) = Mod(\Phi_1)|_{\sigma_1},$$

that is, every model of the $\sigma_1$-coimage of the theory generated by $\Phi_1$ has a $\sigma_1$-expansion that satisfies $\Phi_1$. Given a pushout as above, and $\Phi_1 \subseteq \mathbf{Sen}(\Sigma_1)$, $\Phi_2 \subseteq \mathbf{Sen}(\Sigma_2)$ and $\varphi_2 \in \mathbf{Sen}(\Sigma_2)$ such that

$$\sigma_2'(\Phi_1) \cup \sigma_1'(\Phi_2) \models \sigma_1'(\varphi_2),$$

it is easy to check that $\sigma_1^{-1}(Cl_{\Sigma_1}(\Phi_1))$ is a set of interpolants for $\Phi_1$ and $\varphi_2$ with respect to $\Phi_2$ whenever $\sigma_1 : \Sigma \to \Sigma_1$ is conservative for $\Phi_1$. So, conservativity in this sense is a stronger property than parameterised interpolation; in fact, easy examples show that it is strictly stronger.

## 3. Structured specifications

As we said in Section 2, we will work with an arbitrary institution

$$\mathbf{INS} = \langle \mathbf{Sign}, \mathbf{Sen}, \mathbf{Mod}, \langle \models_\Sigma \rangle_{\Sigma \in |\mathbf{Sign}|} \rangle$$

equipped with two classes $\mathcal{H} \subseteq \mathcal{W} \subseteq \mathbf{Sign}$ of signature morphisms that contain all identities and are closed under composition.

[†] That is, if we stretch the terminology of Goguen and Roşu (2004) somewhat, the module $\langle \sigma_1 : \Sigma \to \Sigma_1, \Phi_1 \rangle$ is conservative

https://doi.org/10.1017/S0960129513000212 Published online by Cambridge University Press

We study specifications built in **INS**. Whatever the specifications are exactly, and whatever their exact written form, each specification has to determine a class of programs that correctly realise it. If the models of the institution capture (the semantics of) the programs we want to deal with, and they have signatures capturing their (static) interfaces, then the most basic semantics of a specification is given in terms of its signature and its class of models. Consequently, we will consider a class *Spec* of *specifications* in **INS** with a semantics that for each specification $SP \in Spec$ defines its signature $Sig[SP]$ and its class of models

$$Mod[SP] \subseteq |\mathbf{Mod}(Sig[SP])|.$$

We will refer to specifications $SP$ with $Sig[SP] = \Sigma$ as $\Sigma$-*specifications* and write $Spec(\Sigma)$ for the class of all $\Sigma$-specifications. When we want to stress that we are dealing with specifications built in the particular institution **INS**, we will write $Spec_{\mathbf{INS}}$ and $Spec_{\mathbf{INS}}(\Sigma)$ rather than just $Spec$ and $Spec(\Sigma)$.

The semantics determines an obvious notion of specification equivalence: specifications $SP_1$ and $SP_2$ are equivalent, written $SP_1 \equiv SP_2$, if their semantics coincide: that is,

$$Sig[SP_1] = Sig[SP_2]$$
$$Mod[SP_1] = Mod[SP_2].$$

The simplest specifications are *presentations* that simply give a set of axioms asserting the required properties. We write such *flat specifications* as $\langle\Sigma, \Phi\rangle$ for any $\Sigma \in |\mathbf{Sign}|$ and $\Phi \subseteq \mathbf{Sen}(\Sigma)$ and define their semantics in the obvious way:

$$\begin{array}{rcl} Sig[\langle\Sigma, \Phi\rangle] & = & \Sigma \\ Mod[\langle\Sigma, \Phi\rangle] & = & \{M \in |\mathbf{Mod}(\Sigma)| \mid M \models \Phi\}. \end{array}$$

Following the ideas of Burstall and Goguen (1977) and Burstall and Goguen (1981), more complex specifications can now be systematically formed using a collection of *specification-building operations*. This stratified way of designing specification formalisms, with a clear separation of basic blocks given as flat specifications from specification structuring mechanisms that are largely independent of the underlying logic, has become standard and can be usefully exploited to ensure clarity and reusability in the context of different logical systems formalised as institutions. One example of a specification formalism that is structured in this way is Casl (Bidoit and Mosses 2004; Mosses 2004).

We will assume that specification-building operations are 'strongly typed' by specification signatures, and write

$$\mathbf{sbo} : Spec(\Sigma_1) \times \ldots \times Spec(\Sigma_n) \rightarrow Spec(\Sigma)$$

to indicate that a specification-building operation **sbo** takes any specifications

$$SP_1 \in Spec(\Sigma_1), \ldots, SP_n \in Spec(\Sigma_n)$$

and yields a specification

$$\mathbf{sbo}(SP_1, \ldots, SP_n) \in Spec(\Sigma).$$

https://doi.org/10.1017/S0960129513000212 Published online by Cambridge University Press

The meaning of such a specification-building operation is then given as a function on classes of models:

$$[\![\mathbf{sbo}]\!] : \mathcal{P}(|\mathbf{Mod}(\Sigma_1)|) \times \ldots \times \mathcal{P}(|\mathbf{Mod}(\Sigma_n)|) \to \mathcal{P}(|\mathbf{Mod}(\Sigma)|).$$

The semantics of specifications is then given compositionally, by defining the model class of $\mathbf{sbo}(SP_1, \ldots, SP_n)$ in terms of the model classes of $SP_1, \ldots, SP_n$ using $[\![\mathbf{sbo}]\!]$.

Flat specifications may be viewed as constant (nullary) specification-building operations. In addition to flat specifications, we will concentrate here on three (families of) kernel specification-building operations, originating from ASL (Sannella and Wirsing 1983), and then re-introduced in Sannella and Tarlecki (1988) with institution-independent model-class semantics. These operations are also at the core of CASL, and we will use here a notation that is closer to the syntax of CASL; see also Sannella and Tarlecki (2012).

**Union:** For any signature $\Sigma$, we have

$$\_ \cup \_ : Spec(\Sigma) \times Spec(\Sigma) \to Spec(\Sigma)$$

with

$$[\![\_ \cup \_]\!] = (\_ \cap \_).$$

That is, given $\Sigma$-specifications $SP_1$ and $SP_2$, we obtain a specification $SP_1 \cup SP_2$ with the following semantics:

$$\begin{aligned} Sig[SP_1 \cup SP_2] &= \Sigma \\ Mod[SP_1 \cup SP_2] &= Mod[SP_1] \cap Mod[SP_2]. \end{aligned}$$

$SP_1 \cup SP_2$ combines the constraints imposed by $SP_1$ and $SP_2$.

**Translation:** For any signature morphism $\sigma : \Sigma \to \Sigma'$ in $\mathcal{W}$, we have

$$\_ \ \mathbf{with}\ \sigma : Spec(\Sigma) \to Spec(\Sigma')$$

with

$$[\![\_ \ \mathbf{with}\ \sigma]\!] = (\_|_\sigma^{-1})$$

where $\_|_\sigma^{-1}$ is the coimage function with respect to the $\sigma$-reduct of models. That is, given any $\Sigma$-specification $SP$, we obtain a specification $SP$ **with** $\sigma$ with the following semantics:

$$\begin{aligned} Sig[SP\ \mathbf{with}\ \sigma] &= \Sigma' \\ Mod[SP\ \mathbf{with}\ \sigma] &= \{M' \in |\mathbf{Mod}(\Sigma')| \mid M'|_\sigma \in Mod[SP]\}. \end{aligned}$$

$SP$ **with** $\sigma$ changes the names in $SP$ according to $\sigma$, and adds new components.

**Hiding:** For any signature morphism $\sigma : \Sigma' \to \Sigma$ in $\mathcal{H}$, we have

$$\_ \ \mathbf{hide\ via}\ \sigma : Spec(\Sigma) \to Spec(\Sigma')$$

with

$$[\![\_ \ \mathbf{hide\ via}\ \sigma]\!] = (\_|_\sigma)$$

where $\_|_\sigma$ is the image function with respect to the $\sigma$-reduct of models. That is, given any $\Sigma$-specification $SP$, we obtain a specification $SP$ **hide via** $\sigma$ with the following

https://doi.org/10.1017/S0960129513000212 Published online by Cambridge University Press

semantics:

$$\begin{aligned} Sig[SP\ \textbf{hide via}\ \sigma] &= \Sigma' \\ Mod[SP\ \textbf{hide via}\ \sigma] &= \{M|_\sigma \mid M \in Mod[SP]\}. \end{aligned}$$

$SP$ **hide via** $\sigma$ views $SP$ as a $\Sigma'$-specification, hiding auxiliary components.

We will write $Spec^{UTH}$ for the class of specifications built from flat specifications using union of specifications over common signatures, translation along morphisms in $\mathcal{W}$ and hiding with respect to morphisms in $\mathcal{H}$. Note that the definitions of the syntax and of the signature for specifications in $Spec^{UTH}$ do not depend on the models and satisfaction relations of the institution involved (although, of course, they are used to determine the model-class semantics of specifications) but only on the category of signatures **Sign**, with indicated classes $\mathcal{H}, \mathcal{W} \subseteq$ **Sign**, and sentences given by the functor **Sen** : **Sign** $\to$ **Set**. When we want to make this dependency and the independence from the other components of the institution more explicit, we write $Spec^{UTH}_{\mathbf{Sen}}$ for $Spec^{UTH}$. However, in general, there may be specification-building operations that involve the model part (or satisfaction relations) of the underlying institution even in the formulation of the 'syntax' – see, for instance, the *singleton* operation used in Sannella *et al.* (1992).

A specification is *finitary* if all the flat specifications it involves have a finite set of axioms.

The following normal form theorem provides an important technical tool.

**Theorem 3.1.** If **INS** is $\langle \mathcal{H}, \mathcal{W} \rangle$-exact, then any specification $SP \in Spec^{UTH}$ has an equivalent *normal form* $\mathsf{nf}(SP)$ given as

$$\langle \Sigma', \Phi' \rangle\ \textbf{hide via}\ \sigma,$$

for some $\Sigma' \in |\mathbf{Sign}|$, $\sigma : Sig[SP] \to \Sigma'$ in $\mathcal{H}$ and $\Phi' \subseteq \mathbf{Sen}(\Sigma')$. Moreover, $\Phi'$ is finite if $SP$ is finitary.

We will omit the explicit inductive definition of $\mathsf{nf}(SP)$ and the proof of equivalence – such normal form results have been well known since Bergstra *et al.* (1990), with a predecessor in Ehrig *et al.* (1983) and the current general version in Borzyszkowski (2002). Note that (weak) $\langle \mathcal{H}, \mathcal{W} \rangle$-exactness of the institution considered is crucial here.

In Goguen and Roşu (2004), specifications of the form $\langle \Sigma, \Phi \rangle$ **hide via** $\sigma$ are taken as the basic meanings of specification expressions. The above theorem shows that this causes no loss with respect to the model-class semantics, at least for specifications built using the operations introduced above.

## 4. Property-oriented semantics for structured specifications

While we view the semantics of specifications given in terms of their model classes as the most basic, their logical consequences are obviously of prime importance.

A $\Sigma$-sentence $\varphi \in \mathbf{Sen}(\Sigma)$ is a *semantic consequence* of a $\Sigma$-specification $SP \in Spec(\Sigma)$ if $Mod[SP] \models \varphi$; we write this $SP \models \varphi$. The set of all semantic consequences of $SP$, called the *theory* of $SP$, is denoted by $Th(SP)$, so

$$Th(SP) = Th(Mod[SP]),$$

https://doi.org/10.1017/S0960129513000212 Published online by Cambridge University Press

and, in particular,

$$Th(\langle \Sigma, \Phi \rangle) = Th(Mod[\langle \Sigma, \Phi \rangle]) = Cl_\Sigma(\Phi).$$

Some authors go as far as to take the theory assigned to a specification as its meaning. This goes back to Clear (Burstall and Goguen 1980), and is the stance taken in Diaconescu *et al.* (1993). In this section and the next we will discuss this option and its relationship with the model-class semantics defined above.

We define a *property-oriented semantics* for specifications to be any function $\mathcal{T}$ that assigns to each specification $SP \in Spec$ a set

$$\mathcal{T}(SP) \subseteq \mathbf{Sen}(Sig[SP])$$

of $Sig[SP]$-sentences.

The assignment $Th$ that maps each specification $SP$ to its theory $Th(SP)$ is one such semantics. In fact, this is the 'best' such semantics in the sense that it captures all and only the properties that hold in all models of the given specification. We will use it as a yardstick to measure the 'strength' and 'soundness' of other such semantics.

We will now introduce some vocabulary to talk about properties of such semantics. Let $\mathcal{T}$ be a property-oriented semantics for specifications. Then:

— $\mathcal{T}$ is *sound* if

$$\mathcal{T}(SP) \subseteq Th(SP)$$

for every specification $SP \in Spec$.

— A sound $\mathcal{T}$ is *complete* if

$$\mathcal{T}(SP) = Th(SP)$$

for every specification $SP \in Spec$.

— $\mathcal{T}$ is *monotone* for a specification-building operation

$$\mathbf{sbo} : Spec(\Sigma_1) \times \ldots \times Spec(\Sigma_n) \to Spec(\Sigma)$$

if

$$\mathcal{T}(\mathbf{sbo}(SP_1, \ldots, SP_n)) \subseteq \mathcal{T}(\mathbf{sbo}(SP'_1, \ldots, SP'_n))$$

for all specifications $SP_1, SP'_1 \in Spec(\Sigma_1), \ldots, SP_n, SP'_n \in Spec(\Sigma_n)$ such that

$$\mathcal{T}(SP_i) \subseteq \mathcal{T}(SP'_i), \quad \text{for } i = 1, \ldots, n.$$

— $\mathcal{T}$ is *compositional* for a specification-building operation

$$\mathbf{sbo} : Spec(\Sigma_1) \times \ldots \times Spec(\Sigma_n) \to Spec(\Sigma)$$

if

$$\mathcal{T}(\mathbf{sbo}(SP_1, \ldots, SP_n)) = \mathcal{T}(\mathbf{sbo}(SP'_1, \ldots, SP'_n))$$

for all specifications $SP_1, SP'_1 \in Spec(\Sigma_1), \ldots, SP_n, SP'_n \in Spec(\Sigma_n)$ such that

$$\mathcal{T}(SP_i) = \mathcal{T}(SP'_i), \quad \text{for } i = 1, \ldots, n.$$

— A sound $\mathcal{T}$ is *closed-complete* for a specification-building operation

$$\mathbf{sbo} : Spec(\Sigma_1) \times \ldots \times Spec(\Sigma_n) \to Spec(\Sigma)$$

if

$$\mathcal{T}(\mathbf{sbo}(SP_1,\dots,SP_n)) = Th(\mathbf{sbo}(SP_1,\dots,SP_n))$$

for all $SP_1 \in Spec(\Sigma_1),\dots,SP_n \in Spec(\Sigma_n)$ such that

$$Mod_{Sig[SP_i]}(\mathcal{T}(SP_i)) = Mod[SP_i] \quad \text{for } i = 1,\dots,n.$$

— $\mathcal{T}$ is *flat-complete* if

$$\mathcal{T}(\langle\Sigma,\Phi\rangle) = Cl_\Sigma(\Phi)$$

for every signature $\Sigma$ and set $\Phi$ of $\Sigma$-sentences.

— $\mathcal{T}$ is *extensive* if

$$\Phi \subseteq \mathcal{T}(\langle\Sigma,\Phi\rangle)$$

for every signature $\Sigma$ and set $\Phi$ of $\Sigma$-sentences.

— $\mathcal{T}$ is *theory-oriented* if for all specifications $SP \in Spec$, we have $\mathcal{T}(SP)$ is a theory (that is, a set of sentences that is closed under semantic consequence).

$\mathcal{T}$ is *monotone* (respectively, *compositional*, *closed-complete*) if it is monotone (respectively, compositional, closed-complete) for all the specification-building operations in use.

Soundness is the property we must insist on for any property-oriented semantics. Completeness is the goal we should aim to soundly approximate as accurately as possible. Compositionality (implied by monotonicity) is a crucial property, which is needed to deal with large structured specifications in a modular way. Closed-completeness is a technical notion to capture how accurate the semantics is for a given specification-building operation: we want the semantics for a specification built using an operation to be complete at least under the assumption that it exactly captures the model classes of the argument specifications (which for theory-oriented semantics is properly stronger than completeness of the semantics for the argument specifications). Flat-completeness is closed-completeness for flat specifications as nullary specification-building operations. Extensiveness requires the semantics of a flat specification to include all of its axioms. Surprisingly, this simple technical condition turns out to play a key role in the results below. Clearly, any flat-complete semantics is extensive. We usually expect semantics to be theory-oriented: we could in principle always close the set of properties given in one way or another under semantic consequences, but this would make our analysis of the issues of dealing with structured specifications more restrictive, and potentially dependent on the completeness of entailment used for the underlying logical system. Any (flat-complete and) closed-complete semantics is theory-oriented. Any extensive and theory-oriented semantics is flat-complete.

The semantics $Th$ above, which is defined using the model-class semantics for specifications, is sound, complete and theory-oriented. It is compositional for hiding: for any signature morphism $\sigma : \Sigma' \to \Sigma$ and $\Sigma$-specification $SP$, we have

$$Th(SP \textbf{ hide via } \sigma) = \sigma^{-1}(Th(SP))$$

(Sannella and Tarlecki 1988); note that, by the satisfaction condition, $\sigma^{-1}(\Phi)$ is a theory if $\Phi$ is a theory.

https://doi.org/10.1017/S0960129513000212 Published online by Cambridge University Press

A key drawback of *Th* is that it is not compositional for union and translation, as we will illustrate in the following counterexamples, which will be presented by constructing an artificial institution in which the point is clear – the reader is encouraged to look for analogous situations in more standard logical systems.

**Example 4.1.** Consider an institution **INS** with exactly two signatures $\Sigma$ and $\Sigma'$, and $\sigma : \Sigma \to \Sigma'$ as the only non-identity signature morphism. Let

$$\mathbf{Sen}(\Sigma) = \{\varphi, \varphi'\}$$
$$\mathbf{Sen}(\Sigma') = \{\varphi, \varphi', \psi_1, \psi_2\},$$

with $\sigma$-translation preserving $\varphi$ and $\varphi'$, and let

$$|\mathbf{Mod}(\Sigma)| = |\mathbf{Mod}(\Sigma')| = \{M_1, M_2\},$$

with the identity $\sigma$-reduct. Define

$$\begin{array}{lll} M_1 \models_\Sigma \varphi & M_1 \models_{\Sigma'} \varphi & M_1 \models_{\Sigma'} \psi_1 \\ M_2 \models_\Sigma \varphi & M_2 \models_{\Sigma'} \varphi & M_2 \not\models_{\Sigma'} \psi_1 \\ M_1 \not\models_\Sigma \varphi' & M_1 \not\models_{\Sigma'} \varphi' & M_1 \not\models_{\Sigma'} \psi_2 \\ M_2 \not\models_\Sigma \varphi' & M_2 \not\models_{\Sigma'} \varphi' & M_2 \models_{\Sigma'} \psi_2. \end{array}$$

In $\Sigma'$, we have

$$Mod(\{\psi_1\}) = \{M_1\}$$
$$Mod(\{\psi_2\}) = \{M_2\}.$$

Let $SP_1$ be

$$\langle \Sigma', \{\psi_1\} \rangle \ \mathbf{hide\ via}\ \sigma$$

and $SP_2$ be

$$\langle \Sigma', \{\psi_2\} \rangle \ \mathbf{hide\ via}\ \sigma.$$

Then

$$Mod[SP_1] = \{M_1\}$$
$$Mod[SP_2] = \{M_2\},$$

yielding

$$Th(SP_1) = \{\varphi\} = Th(SP_2).$$

Now:

— $Mod[SP_1 \cup SP_2] = \emptyset$, so

$$Th(SP_1 \cup SP_2) = \{\varphi, \varphi'\},$$

which is distinct from

$$Th(SP_2 \cup SP_2) = Th(SP_2) = \{\varphi\}.$$

— $Th(SP_1\ \mathbf{with}\ \sigma) = \{\varphi, \psi_1\}$, which is distinct from $Th(SP_2\ \mathbf{with}\ \sigma) = \{\varphi, \psi_2\}$.

This shows that *Th* is compositional for neither union nor translation.

The lack of compositionality of *Th* for union and translation, as well as a natural consideration of proof-theoretic issues (see Section 7 below), led to the following standard compositional property-oriented semantics $\mathcal{T}_{\mathbf{INS}}$ for specifications in $Spec^{UTH}_{\mathbf{INS}}$. This semantics has its origins in the proof rules in Sannella and Tarlecki (1988), was given in Bergstra *et al.* (1990) and was used in Diaconescu *et al.* (1993). Here is the inductive definition:

$$\mathcal{T}_{\mathbf{INS}}(\langle\Sigma,\Phi\rangle) = Cl_{\Sigma}(\Phi)$$
$$\mathcal{T}_{\mathbf{INS}}(SP \cup SP') = Cl_{Sig[SP]}(\mathcal{T}_{\mathbf{INS}}(SP) \cup \mathcal{T}_{\mathbf{INS}}(SP'))$$
$$\mathcal{T}_{\mathbf{INS}}(SP \textbf{ with } \sigma : Sig[SP] \to \Sigma) = Cl_{\Sigma}(\sigma(\mathcal{T}_{\mathbf{INS}}(SP)))$$
$$\mathcal{T}_{\mathbf{INS}}(SP \textbf{ hide via } \sigma : \Sigma \to Sig[SP]) = \sigma^{-1}(\mathcal{T}_{\mathbf{INS}}(SP)).$$

**Proposition 4.2.** $\mathcal{T}_{\mathbf{INS}}$ is a sound theory-oriented semantics for specifications built from flat specifications using union, translation and hiding. It is monotone, compositional, extensive, flat-complete and closed-complete for union, translation and hiding.

*Proof.* Monotonicity and compositionality follow from the definitions, while soundness requires a simple inductive proof. Closed-completeness for hiding follows from the definitions and the satisfaction condition. (Soundness, monotonicity and closed-completeness were shown in Sannella and Tarlecki (1988).) □

The missing property here is completeness – and, indeed, $\mathcal{T}_{\mathbf{INS}}$ is not complete, as the following counterexample shows.

**Example 4.3.** Consider the following specifications built in the institution **EQ** of equational logic, where we use what is a hopefully self-explanatory notation based on the syntax of CASL[†]:

$$\begin{array}{l}\textbf{spec } SP_0 = \textbf{sorts } s\\ \qquad\qquad \textbf{opns } a,b,c : s,\\ \qquad\qquad\qquad f,g : s \to s\\ \qquad\qquad \bullet\ f(a) = b\\ \qquad\qquad \bullet\ g(a) = c\\ \textbf{spec } SP_1 = SP_0 \textbf{ hide ops } a : s\\ \textbf{spec } SP\ = SP_1 \textbf{ then } \forall x{:}s \bullet f(x) = g(x)\end{array}$$

This example relies on the fact that the class of models of any set of equations is closed under subalgebras. Note that using conditional equations would not help, as this property also holds for them.

Now, $Mod[SP_1]$ consists of all $Sig[SP_1]$-algebras with an element on which $f$ yields $b$ and $g$ yields $c$. Consequently, given the axiom added in $SP$, we have $SP \models b = c$. However,

[†] In particular, the missing signature morphism in the definition of $SP_1$ is the inclusion from $Sig[SP_0] \setminus \{a : s\}$ to $Sig[SP_0]$, and the definition of $SP$ abbreviates

$$\textbf{spec } SP = SP_1 \cup \langle Sig[SP_1], \{\forall x{:}s \bullet f(x) = g(x)\}\rangle.$$

https://doi.org/10.1017/S0960129513000212 Published online by Cambridge University Press

since any $Sig[SP_1]$-algebra is a subalgebra of an algebra in $Mod[SP_1]$, the equational theory $Th(SP_1)$ is trivial (that is, generated by the empty set). Hence $\mathcal{T}_{\mathbf{INS}}(SP_1)$ also only consists of equational tautologies, and

$$\mathcal{T}_{\mathbf{INS}}(SP) = Cl_{Sig[SP]}(\{\forall x{:}s \bullet f(x) = g(x)\})$$

does not contain $b = c$.

However, completeness does hold if we impose additional requirements on the underlying institution.

**Theorem 4.4.** Suppose **INS** is $\langle \mathcal{H}, \mathcal{W} \rangle$-exact and that it admits parameterised $\langle \mathcal{H}, \mathcal{W} \rangle$-interpolation. Then $\mathcal{T}_{\mathbf{INS}}$ is complete for specifications built from flat specifications using union, translation and hiding.

*Proof.* We assume that $SP \models \varphi$ for a specification $SP$ built from flat specifications using union, translation and hiding, with $Sig[SP] = \Sigma$ and $\varphi \in \mathbf{Sen}(\Sigma)$. We will show that $\varphi \in \mathcal{T}_{\mathbf{INS}}(SP)$ by induction on the structure of $SP$, since this will show completeness as $\mathcal{T}_{\mathbf{INS}}$ is sound by Proposition 4.2:

— Let $SP$ be $\langle \Sigma, \Phi \rangle$ for $\Phi \subseteq \mathbf{Sen}(\Sigma)$.
Then $\Phi \models_\Sigma \varphi$, so

$$\varphi \in Cl_\Sigma(\Phi) = \mathcal{T}_{\mathbf{INS}}(\langle \Sigma, \Phi \rangle).$$

— Let $SP$ be $SP'$ **hide via** $\sigma$ for some specification $SP'$ with $Sig[SP'] = \Sigma'$ and $\sigma : \Sigma \to \Sigma'$ in $\mathcal{H}$.
Then $SP' \models \sigma(\varphi)$. By the induction hypothesis, $\sigma(\varphi) \in \mathcal{T}_{\mathbf{INS}}(SP')$, so

$$\varphi \in \sigma^{-1}(\mathcal{T}_{\mathbf{INS}}(SP')) = \mathcal{T}_{\mathbf{INS}}(SP).$$

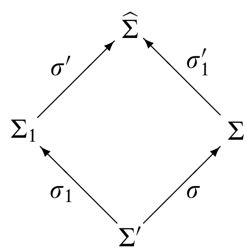
— Let $SP$ be $SP'$ **with** $\sigma$ for some specification $SP'$ with $Sig[SP'] = \Sigma'$ and $\sigma : \Sigma' \to \Sigma$ in $\mathcal{W}$.
By Theorem 3.1,

$$SP' \equiv \langle \Sigma_1, \Phi_1 \rangle \textbf{ hide via } \sigma_1$$

for some $\Sigma_1 \in |\mathbf{Sign}|, \Phi_1 \subseteq \mathbf{Sen}(\Sigma_1)$ and $\sigma_1 : \Sigma' \to \Sigma_1$ in $\mathcal{H}$. Then, as in the (omitted, but well-known) proof of Theorem 3.1, we have

$$SP \equiv \langle \widehat{\Sigma}, \sigma'(\Phi_1) \rangle \textbf{ hide via } \sigma_1',$$

where the following is a pushout in **Sign**:




with $\sigma' \in \mathcal{W}$, $\sigma_1' \in \mathcal{H}$. Then $SP \models \varphi$ implies $\sigma'(\Phi_1) \models_{\widehat{\Sigma}} \sigma_1'(\varphi)$. Hence, by Craig $\langle \mathcal{H}, \mathcal{W} \rangle$-interpolation (the stronger, parameterised version is not needed in this case), there is

https://doi.org/10.1017/S0960129513000212 Published online by Cambridge University Press

an interpolant set $\Psi \subseteq \mathbf{Sen}(\Sigma')$ such that

$$\Phi_1 \models_{\Sigma_1} \sigma_1(\Psi)$$

and

$$\sigma(\Psi) \models_{\Sigma} \varphi.$$

The former yields $SP' \models \Psi$, so, by the induction hypothesis, $\Psi \subseteq \mathcal{T}_{\mathbf{INS}}(SP')$; and by the latter

$$\varphi \in Cl_{\Sigma}(\sigma(\Psi)) \subseteq Cl_{\Sigma}(\sigma(\mathcal{T}_{\mathbf{INS}}(SP'))) = \mathcal{T}_{\mathbf{INS}}(SP).$$

— Let $SP$ be $SP_1 \cup SP_2$ for specifications $SP_1$ and $SP_2$ with $Sig[SP_1] = Sig[SP_2] = \Sigma$. By Theorem 3.1,

$$SP_1 \equiv \langle \Sigma_1, \Phi_1 \rangle \ \textbf{hide via} \ \sigma_1$$

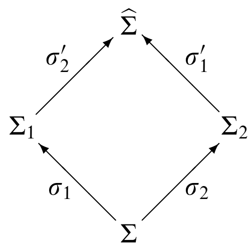
for some $\Sigma_1 \in |\mathbf{Sign}|, \Phi_1 \subseteq \mathbf{Sen}(\Sigma_1)$ and $\sigma_1 : \Sigma \to \Sigma_1$ in $\mathcal{H}$, and

$$SP_2 \equiv \langle \Sigma_2, \Phi_2 \rangle \ \textbf{hide via} \ \sigma_2$$

for some $\Sigma_2 \in |\mathbf{Sign}|, \Phi_2 \subseteq \mathbf{Sen}(\Sigma_2)$ and $\sigma_2 : \Sigma \to \Sigma_2$ in $\mathcal{H} \subseteq \mathcal{W}$. Then

$$SP \equiv \langle \widehat{\Sigma}, \{\sigma_2'(\Phi_1), \sigma_1'(\Phi_2)\} \rangle \ \textbf{hide via} \ \sigma_2;\sigma_1',$$

as in the (omitted) proof of Theorem 3.1, where the following is a pushout in **Sign**:




with $\sigma_1' \in \mathcal{H}$, $\sigma_2' \in \mathcal{W}$. $SP \models \varphi$ implies

$$\sigma_2'(\Phi_1), \sigma_1'(\Phi_2) \models_{\widehat{\Sigma}} \sigma_1'(\sigma_2(\varphi)).$$

Then, by the parameterised $\langle \mathcal{H}, \mathcal{W} \rangle$-interpolation property, there is an interpolant set $\Psi \subseteq \mathbf{Sen}(\Sigma)$ such that

$$\Phi_1 \models_{\Sigma_1} \sigma_1(\Psi)$$

and

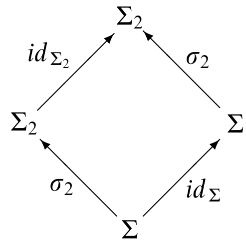
$$\Phi_2 \cup \sigma_2(\Psi) \models_{\Sigma} \sigma_2(\varphi).$$

The former yields $SP_1 \models \Psi$, so by the induction hypothesis,

$$\Psi \subseteq \mathcal{T}_{\mathbf{INS}}(SP_1) \subseteq \mathcal{T}_{\mathbf{INS}}(SP);$$

https://doi.org/10.1017/S0960129513000212 Published online by Cambridge University Press

and, by the latter, considering the pushout




the parameterised $\langle \mathcal{H}, \mathcal{W} \rangle$-interpolation property gives us an interpolant set $\Psi' \subseteq$ **Sen**$(\Sigma)$ (for $\Phi_2$ and $\varphi$ with respect to $\Psi$) such that

$$\Phi_2 \models \sigma_2(\Psi')$$

and

$$\Psi' \cup \Psi \models \varphi.$$

Since the former now yields $SP_2 \models \Psi'$, by the induction hypothesis, we have

$$\Psi' \subseteq \mathcal{T}_{\mathbf{INS}}(SP_2) \subseteq \mathcal{T}_{\mathbf{INS}}(SP),$$

so we also get

$$\varphi \in Cl_{\Sigma}(\mathcal{T}_{\mathbf{INS}}(SP)) = \mathcal{T}_{\mathbf{INS}}(SP),$$

which completes the proof. □

In this proof we have mimicked the proof in Borzyszkowski (2002), which in turn largely followed Bergstra *et al.* (1990), where, perhaps, the role of first-order interpolation for such results was first explored. However, we use a stronger (parameterised) interpolation property instead of Craig interpolation, together with an assumption that the underlying institution is compact and has conjunction and implication. The latter idea was used, for instance, in Diaconescu (2008) to show the completeness of a (stronger) calculus for proving entailment in a context of structured specifications, where the case of specifications built using union was simpler.

Another result of this kind, but for a somewhat different collection of specification-building operations, was given in Goguen and Roşu (2004). They showed soundness and completeness of $\mathcal{T}_{\mathbf{INS}}$ with respect to a semantics that, in essence, calculates a normal form of specification expressions (see Theorem 3.1), but they relied on conservativity of modules (see the footnote at the end of Section 2) rather than on the strictly weaker requirement of parameterised interpolation[†].

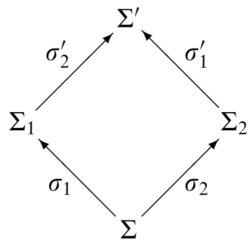
It is easy to see that the parameterised interpolation property is necessary for the above completeness result.

[†] Contrary to a claim in the abstract of Goguen and Roşu (2004), conservativity is not a necessary condition for their results. In fact, they give no technical statement that repeats this claim, and merely show that completeness fails in certain non-conservative examples.

https://doi.org/10.1017/S0960129513000212 Published online by Cambridge University Press

**Proposition 4.5.** Suppose **INS** is an $\langle \mathcal{H}, \mathcal{W} \rangle$-exact institution such that $\mathcal{T}_{\mathbf{INS}}$ is complete for specifications built from flat specifications using union, translation and hiding in **INS**. Then **INS** admits parameterised $\langle \mathcal{H}, \mathcal{W} \rangle$-interpolation.

*Proof.* Consider any pushout diagram in **Sign**




with $\sigma_1, \sigma_1' \in \mathcal{H}$, $\sigma_2, \sigma_2' \in \mathcal{W}$, and

$$\Phi_1 \subseteq \mathbf{Sen}(\Sigma_1)$$
$$\Phi_2 \subseteq \mathbf{Sen}(\Sigma_2)$$
$$\varphi_2 \in \mathbf{Sen}(\Sigma_2)$$

such that

$$\sigma_2'(\Phi_1) \cup \sigma_1'(\Phi_2) \models \sigma_1'(\varphi_2).$$

Let $SP \in Spec^{UTH}(\Sigma_2)$ be the specification

$$((\langle \Sigma_1, \Phi_1 \rangle \textbf{ hide via } \sigma_1) \textbf{ with } \sigma_2) \cup \langle \Sigma_2, \Phi_2 \rangle.$$

Then

$$\begin{aligned} \mathcal{T}_{\mathbf{INS}}(SP) &= Cl_{\Sigma_2}(Cl_{\Sigma_2}(\sigma_2(\mathcal{T}_{\mathbf{INS}}(\langle \Sigma_1, \Phi_1 \rangle \textbf{ hide via } \sigma_1))) \cup \Phi_2) \\ &= Cl_{\Sigma_2}(\sigma_2(\mathcal{T}_{\mathbf{INS}}(\langle \Sigma_1, \Phi_1 \rangle \textbf{ hide via } \sigma_1)) \cup \Phi_2). \end{aligned}$$

If $\mathcal{T}_{\mathbf{INS}}$ is complete, we have

$$\varphi_2 \in Th(SP) = \mathcal{T}_{\mathbf{INS}}(SP),$$

and we can take

$$\Phi = \mathcal{T}_{\mathbf{INS}}(\langle \Sigma_1, \Phi_1 \rangle \textbf{ hide via } \sigma_1)$$

to be a set of interpolants for $\Phi_1$ and $\varphi_2$ with respect to $\Phi_2$. □

## 5. Comparing property-oriented semantics

As can be seen from Theorem 4.4 and Proposition 4.5, $\mathcal{T}_{\mathbf{INS}}$ is only complete under rather strong assumptions concerning the underlying logical system. Even though these hold for **FOPEQ**, the institution of first-order logic (with classes $\mathcal{W}$ and $\mathcal{H}$ chosen, say, to be all injective signature morphisms), this is a rather rare situation, and $\mathcal{T}_{\mathbf{INS}}$ is incomplete in many typical institutions of practical importance, including **EQ** and $\mathbf{EQ}^{ne}$ (see Example 4.3). There have been attempts to preserve compositionality and still ensure completeness (Mossakowski *et al.* 2006). However, we show below that to improve on $\mathcal{T}_{\mathbf{INS}}$, at least some aspects of compositionality must be sacrificed.

https://doi.org/10.1017/S0960129513000212 Published online by Cambridge University Press

**Theorem 5.1.** Consider two property-oriented semantics $\mathcal{T}$ and $\mathcal{T}'$ for specifications constructed using a set of specification-building operations, including all flat specifications. Let $\mathcal{T}$ be sound, monotone and closed-complete. Let $\mathcal{T}'$ be theory-oriented, sound, compositional and extensive. Then $\mathcal{T}$ is at least as strong as $\mathcal{T}'$: that is, for every $SP$,

$$\mathcal{T}'(SP) \subseteq \mathcal{T}(SP).$$

*Proof.* We use induction on the structure of $SP$. For flat specifications,

$$\mathcal{T}'(\langle \Sigma, \Phi \rangle) \subseteq Cl_\Sigma(\Phi) = \mathcal{T}(\langle \Sigma, \Phi \rangle)$$

by soundness of $\mathcal{T}'$ and flat-completeness of $\mathcal{T}$ (which is the same as closed-completeness for flat specifications).

More generally, consider any well-formed specification $\mathbf{sbo}(SP_1, \ldots, SP_n)$ with $\Sigma_i = Sig[SP_i]$, where $i = 1, \ldots, n$ here and below, and suppose

$$\mathcal{T}'(SP_i) \subseteq \mathcal{T}(SP_i).$$

Since $\mathcal{T}'$ is theory-oriented and extensive,

$$\mathcal{T}'(\langle \Sigma_i, \mathcal{T}'(SP_i) \rangle) = \mathcal{T}'(SP_i).$$

We also have

$$\mathcal{T}(\langle \Sigma_i, \mathcal{T}'(SP_i) \rangle) = \mathcal{T}'(SP_i).$$

Then, using compositionality of $\mathcal{T}'$, soundness of $\mathcal{T}'$, closed-completeness of $\mathcal{T}$ for **sbo**, and monotonicity of $\mathcal{T}$ (and the induction hypothesis) in turn, we have

$$\begin{aligned}
\mathcal{T}'(\mathbf{sbo}(SP_1, \ldots, SP_n)) &= \mathcal{T}'(\mathbf{sbo}(\langle \Sigma_1, \mathcal{T}'(SP_1) \rangle, \ldots, \langle \Sigma_n, \mathcal{T}'(SP_n) \rangle)) \\
&\subseteq Th(\mathbf{sbo}(\langle \Sigma_1, \mathcal{T}'(SP_1) \rangle, \ldots, \langle \Sigma_n, \mathcal{T}'(SP_n) \rangle)) \\
&= \mathcal{T}(\mathbf{sbo}(\langle \Sigma_1, \mathcal{T}'(SP_1) \rangle, \ldots, \langle \Sigma_n, \mathcal{T}'(SP_n) \rangle)) \\
&\subseteq \mathcal{T}(\mathbf{sbo}(SP_1, \ldots, SP_n)).
\end{aligned}$$

This completes the induction step and thus the proof of the theorem. □

**Corollary 5.2.** $\mathcal{T}_{\mathbf{INS}}$ is at least as strong as any sound, compositional and extensive theory-oriented semantics for structured specifications built from flat specifications using union, translation and hiding.

*Proof.* The statement follows directly from Proposition 4.2 and Theorem 5.1. □

The requirement that the theory-oriented semantics under consideration be extensive is perhaps the most surprising one here. Informally, if we forget about some axioms, we should not be able to soundly get more consequences. However, this requirement cannot be dropped in general, as the following counterexample shows.

**Example 5.3.** Consider an institution $\mathbf{INS}_0$ with signatures $\Sigma$ and $\Sigma'$, and a signature morphism $\sigma : \Sigma \to \Sigma'$. Let

$$\mathbf{Sen}_0(\Sigma) = \{\varphi\}$$
$$\mathbf{Sen}_0(\Sigma') = \{\varphi, \psi\},$$

with $\sigma$-translation preserving $\varphi$, and let

$$|\mathbf{Mod}_0(\Sigma)| = |\mathbf{Mod}_0(\Sigma')| = \{M_0, M_1, M_2\},$$

with the identity $\sigma$-reduct. Suppose

$$\begin{array}{ll} M_0 \not\models \varphi & M_0 \not\models \psi \\ M_1 \models \varphi & M_1 \not\models \psi \\ M_2 \models \varphi & M_2 \models \psi \end{array}$$

(over appropriate signatures) and that we have a $\Sigma$-specification $SP_{bad}$ with

$$Mod[SP_{bad}] = \{M_2\}.$$

Then let $\mathcal{T}'$ be such that

$$\mathcal{T}'(SP_{bad}) = \{\varphi\}$$
$$\mathcal{T}'(SP_{bad} \textbf{ with } \sigma) = \{\varphi, \psi\},$$

and $\mathcal{T}'$ forgets the axiom $\varphi$ in all flat specifications. We can then ensure that for all $\Sigma$-specifications $SP$, if $\varphi \in \mathcal{T}'(SP)$, then $M_1 \notin Mod[SP]$, since, very informally, if $\varphi$ cannot be put into the theory of a specification as an axiom, the only way it can be there is as a consequence of $\psi$. So $\mathcal{T}'$ is sound and compositional, but for the $\Sigma'$-specification $SP_{bad}$ **with** $\sigma$, it is stronger than the expected sound, monotone and closed-complete theory-oriented semantics $\mathcal{T}_{\mathbf{INS}_0}$ that yields

$$\mathcal{T}_{\mathbf{INS}_0}(SP_{bad}) = \{\varphi\}$$
$$\mathcal{T}_{\mathbf{INS}_0}(SP_{bad} \textbf{ with } \sigma) = \{\varphi\}.$$

To make this fully specific, we suppose there are no other signatures and non-identity signature morphisms, and no other sentences and models, and we define:

$$\mathcal{T}'(\langle\Sigma, \Phi\rangle) = \varnothing$$
$$\mathcal{T}'(\langle\Sigma', \Phi'\rangle) = Cl_{\Sigma'}(\Phi' \setminus \{\varphi\})$$
$$\mathcal{T}'(SP_1 \cup SP_2) = Cl_{Sig[SP_1]}(\mathcal{T}'(SP_1) \cup \mathcal{T}'(SP_2))$$
$$\mathcal{T}'(SP' \textbf{ hide via } \sigma) = \sigma^{-1}(\mathcal{T}'(SP))$$
$$\mathcal{T}'(SP \textbf{ with } \sigma) = \begin{cases} \varnothing & \text{if } \mathcal{T}'(SP) = \varnothing \\ \{\varphi, \psi\} & \text{if } \varphi \in \mathcal{T}'(SP). \end{cases}$$

We now put

$$SP_{bad} = \langle\Sigma', \{\psi\}\rangle \textbf{ hide via } \sigma$$

and check that it has the properties required above. Indeed, $\mathcal{T}'$ is a sound compositional theory-oriented semantics, but

$$\mathcal{T}'(SP_{bad} \textbf{ with } \sigma) = \{\varphi, \psi\}$$

is a strictly larger theory than

$$\mathcal{T}_{\mathbf{INS}_0}(SP_{bad} \textbf{ with } \sigma) = \{\varphi\}.$$

https://doi.org/10.1017/S0960129513000212 Published online by Cambridge University Press

The following counterexample shows that we cannot drop the requirement that the semantics under consideration be theory-oriented either.

**Example 5.4.** Building on Example 5.3, consider an institution $\mathbf{INS}_1$ with exactly two signatures $\Sigma$ and $\Sigma'$, and $\sigma : \Sigma \to \Sigma'$ as the only non-identity signature morphism. Let

$$\mathbf{Sen}_1(\Sigma) = \{\varphi, \varphi'\}$$
$$\mathbf{Sen}_1(\Sigma') = \{\varphi, \varphi', \psi\},$$

with $\sigma$-translation preserving $\varphi$ and $\varphi'$, and let

$$|\mathbf{Mod}_1(\Sigma)| = |\mathbf{Mod}_1(\Sigma')| = \{M_0, M_1, M_2, M_3\},$$

with the identity $\sigma$-reduct. Define

$$\begin{array}{ccc} M_0 \not\models \varphi' & M_0 \not\models \varphi & M_0 \not\models \psi \\ M_1 \models \varphi' & M_1 \not\models \varphi & M_1 \not\models \psi \\ M_2 \models \varphi' & M_2 \models \varphi & M_2 \not\models \psi \\ M_3 \models \varphi' & M_3 \models \varphi & M_3 \models \psi \end{array}$$

(over appropriate signatures).

| $\models$ | $\varphi'$ | $\varphi$ | $\psi$ |
|---|---|---|---|
| $M_0$ | $-$ | $-$ | $-$ |
| $M_1$ | $+$ | $-$ | $-$ |
| $M_2$ | $+$ | $+$ | $-$ |
| $M_3$ | $+$ | $+$ | $+$ |

So, over the appropriate signatures, we have $\varphi \models \varphi'$ and $\psi \models \varphi$.

We now define a property-oriented semantics $\mathcal{T}''$ using inductive clauses that essentially copy those for $\mathcal{T}_{\mathbf{INS}_1}$, except that for the flat $\Sigma'$-specification with $\psi$ as the only axiom, we omit exactly one of its consequences, $\varphi'$, and then for translation along $\sigma$ when the properties of the argument specification given by the semantics include $\varphi$ but not $\varphi'$, we add $\psi$ as a property of the translated specification. The latter happens only if the specification to be translated along $\sigma$ results from hiding with respect to $\sigma$ of a specification with $\psi$ as the only axiom. We have

$$\mathcal{T}''(\langle \Sigma, \Phi \rangle) = Cl_\Sigma(\Phi)$$
$$\mathcal{T}''(\langle \Sigma', \Phi' \rangle) = \begin{cases} \{\psi, \varphi\} & \text{if } \Phi' = \{\psi\} \\ Cl_{\Sigma'}(\Phi') & \text{otherwise} \end{cases}$$
$$\mathcal{T}''(SP_1 \cup SP_2) = Cl_{Sig[SP_1]}(\mathcal{T}''(SP_1) \cup \mathcal{T}''(SP_2))$$
$$\mathcal{T}''(SP' \ \textbf{hide via} \ \sigma) = \sigma^{-1}(\mathcal{T}''(SP'))$$
$$\mathcal{T}''(SP \ \textbf{with} \ \sigma) = \begin{cases} Cl_{\Sigma'}(\sigma(\mathcal{T}''(SP))) & \text{if } \varphi' \in \mathcal{T}''(SP) \text{ or } \varphi \notin \mathcal{T}''(SP) \\ Cl_{\Sigma'}(\sigma(\mathcal{T}''(SP)) \cup \{\psi\}) & \text{if } \varphi' \notin \mathcal{T}''(SP) \text{ and } \varphi \in \mathcal{T}''(SP). \end{cases}$$

https://doi.org/10.1017/S0960129513000212 Published online by Cambridge University Press

$\mathcal{T}''$ is a sound, compositional, extensive property-oriented semantics. However, it is not theory-oriented since, in particular,

$$\mathcal{T}''(\langle \Sigma', \{\psi\}\rangle) = \{\psi, \varphi\}$$

is not closed under consequence (it does not contain $\varphi'$). Since the semantics $\mathcal{T}''$ yields a theory for all other $\Sigma'$-specifications and for flat $\Sigma$-specifications, this is exploited to 'enlarge our knowledge' about $\Sigma$-specifications $SP$ with $\mathcal{T}''(SP)$ containing $\varphi$ but not $\varphi'$. Specifically, as in Example 5.3, putting

$$SP_{bad} = \langle \Sigma', \{\psi\}\rangle \textbf{ hide via } \sigma,$$

we get

$$\mathcal{T}''(SP_{bad}) = \{\varphi\},$$

(while $\mathcal{T}_{\mathbf{INS}_1}(SP_{bad}) = \{\varphi, \varphi'\}$), so

$$\mathcal{T}''(SP_{bad} \textbf{ with } \sigma) = \{\psi, \varphi, \varphi'\},$$

which is a strictly larger theory than

$$\mathcal{T}_{\mathbf{INS}_1}(SP_{bad} \textbf{ with } \sigma) = \{\varphi, \varphi'\}.$$

The counterexample property-oriented semantics given in Example 5.4 is compositional but not monotone. This is necessarily so, since for the semantics considered in Theorem 5.1 and Corollary 5.2, if we assume that it is monotone, we can drop the requirement that it be theory-oriented.

**Theorem 5.5.** Consider two property-oriented semantics $\mathcal{T}$ and $\mathcal{T}'$ for specifications constructed using a set of specification-building operations, including all flat specifications. Let $\mathcal{T}$ be sound, monotone and closed-complete. Let $\mathcal{T}'$ be sound, monotone and extensive. Then $\mathcal{T}$ is at least as strong as $\mathcal{T}'$: that is, for every $SP$,

$$\mathcal{T}'(SP) \subseteq \mathcal{T}(SP).$$

*Proof.* We use induction on the structure of $SP$. For flat specifications,

$$\mathcal{T}'(\langle \Sigma, \Phi\rangle) \subseteq Cl_{\Sigma}(\Phi) = \mathcal{T}(\langle \Sigma, \Phi\rangle)$$

by soundness of $\mathcal{T}'$ and flat-completeness of $\mathcal{T}$ (which is the same as closed-completeness for flat specifications).

More generally, consider any well-formed specification $\mathbf{sbo}(SP_1, \ldots, SP_n)$ with $\Sigma_i = Sig[SP_i]$, where $i = 1, \ldots, n$ here and below, and suppose

$$\mathcal{T}'(SP_i) \subseteq \mathcal{T}(SP_i)$$

(as induction hypothesis). Since $\mathcal{T}'$ is extensive, we have

$$\mathcal{T}'(SP_i) \subseteq \mathcal{T}'(\langle \Sigma_i, \mathcal{T}'(SP_i)\rangle).$$

We also have

$$\mathcal{T}(\langle \Sigma_i, \mathcal{T}'(SP_i)\rangle) = Cl_{\Sigma_i}(\mathcal{T}'(SP_i)) \subseteq Cl_{\Sigma_i}(\mathcal{T}(SP_i)) = \mathcal{T}(SP_i)$$

by applying flat-completeness of $\mathcal{T}$, the induction hypothesis and closure of $\mathcal{T}(SP)$ under consequence ($\mathcal{T}$ is closed-complete, and thus theory-oriented) in turn. Then, using monotonicity of $\mathcal{T}'$, soundness of $\mathcal{T}'$, closed-completeness of $\mathcal{T}$ for **sbo**, and monotonicity of $\mathcal{T}$ in turn, we have

$$\begin{aligned}\mathcal{T}'(\mathbf{sbo}(SP_1,\ldots,SP_n)) &\subseteq \mathcal{T}'(\mathbf{sbo}(\langle\Sigma_1,\mathcal{T}'(SP_1)\rangle,\ldots,\langle\Sigma_n,\mathcal{T}'(SP_n)\rangle))\\ &\subseteq Th(\mathbf{sbo}(\langle\Sigma_1,\mathcal{T}'(SP_1)\rangle,\ldots,\langle\Sigma_n,\mathcal{T}'(SP_n)\rangle))\\ &= \mathcal{T}(\mathbf{sbo}(\langle\Sigma_1,\mathcal{T}'(SP_1)\rangle,\ldots,\langle\Sigma_n,\mathcal{T}'(SP_n)\rangle))\\ &\subseteq \mathcal{T}(\mathbf{sbo}(SP_1,\ldots,SP_n)).\end{aligned}$$

This completes the induction step and thus the proof of the theorem. □

**Corollary 5.6.** $\mathcal{T}_{\mathbf{INS}}$ is at least as strong as any sound, monotone and extensive property-oriented semantics for structured specifications built from flat specifications using union, translation and hiding.

*Proof.* The statement follows directly from Proposition 4.2 and Theorem 5.5. □

Example 5.3 shows that the requirement that the semantics considered in the above corollary be extensive cannot be dropped: the counterexample semantics $\mathcal{T}'$ given there is monotone.

The above analysis of the relative power of property-oriented semantics for structured specifications was based on an implicit assumption that there are no signatures, sentences and specifications other than those we are dealing with. In a way, this is a version of the famous *closed world assumption*. In particular, Examples 5.3 and 5.4 relied on this to justify soundness of the counterexample semantics constructed there, which for some specifications (built from flat specifications using union, translation and hiding) yield a theory that is properly richer than the theory produced by the standard compositional semantics. Consequently, the counterexamples do not apply if we consider the semantics for specifications in some potential extensions of the specification framework.

As before, we consider a class *Spec* of specifications built using a family of specification-building operations. For any class $\mathcal{SP}$ of new specification constants, with model-theoretic semantics given as usual[†], let $Spec(\mathcal{SP})$ be the class of specifications that contains *Spec* and $\mathcal{SP}$ and is closed under the specification-building operations under consideration, with a semantics that extends the semantics for specifications in *Spec* and $\mathcal{SP}$ using the meaning of the specification-building operations as explained in Section 3.

We say that a property-oriented semantics $\mathcal{T}$ for specifications in *Spec* is *persistently sound and compositional* if for any class of new specification constants $\mathcal{SP}_{new}$ with model-

[†] That is, for each $SP \in \mathcal{SP}$, we have

$$Sig[SP] \in |\mathbf{Sign}|$$

and

$$Mod[SP] \subseteq |\mathbf{Mod}(Sig[SP])|,$$

hence we also have

$$Th(SP) = Th(Mod[SP]).$$

https://doi.org/10.1017/S0960129513000212 Published online by Cambridge University Press

class semantics and for any sound property-oriented meaning for specifications in $\mathcal{SP}_{new}$ given by

$$\mathcal{T}_{new}(SP) \subseteq Th(SP)$$

for all $SP \in \mathcal{SP}_{new}$, there is a sound and compositional property-oriented semantics $\widehat{\mathcal{T}}$ for $Spec(\mathcal{SP}_{new})$ that extends $\mathcal{T}$ and $\mathcal{T}_{new}$, that is, such that

$$\widehat{\mathcal{T}}(SP) = \mathcal{T}(SP)$$

for $SP \in Spec$ and

$$\widehat{\mathcal{T}}(SP) = \mathcal{T}_{new}(SP)$$

for $SP \in \mathcal{SP}_{new}$. Clearly, any persistently sound and compositional property-oriented semantics is sound and compositional, but the opposite implication fails in general.

It is easy to check that the standard compositional property-oriented semantics $\mathcal{T}_{\mathbf{INS}}$ for specifications built from flat specifications using union, translation and hiding is persistently sound and compositional. Moreover, it is the strongest such property-oriented semantics. In contrast to the previous results, this does not require the semantics under consideration to be extensive.

**Theorem 5.7.** Consider two property-oriented semantics $\mathcal{T}$ and $\mathcal{T}'$ for specifications constructed using a set of specification-building operations, including all flat specifications. Let $\mathcal{T}$ be sound, monotone and closed-complete, and let $\mathcal{T}'$ be persistently sound and compositional. Then $\mathcal{T}$ is at least as strong as $\mathcal{T}'$: that is, for every $SP$,

$$\mathcal{T}'(SP) \subseteq \mathcal{T}(SP).$$

*Proof.* We proceed by induction on the structure of specifications. Consider a specification $\mathbf{sbo}(SP_1, \dots, SP_n)$ for $n \geqslant 0$, where

$$\begin{gathered} \mathcal{T}'(SP_1) \subseteq \mathcal{T}(SP_1) \\ \vdots \\ \mathcal{T}'(SP_n) \subseteq \mathcal{T}(SP_n). \end{gathered}$$

We need to show

$$\mathcal{T}'(\mathbf{sbo}(SP_1, \dots, SP_n)) \subseteq \mathcal{T}(\mathbf{sbo}(SP_1, \dots, SP_n)).$$

Let $SP'_1, \dots, SP'_n$ be new specification constants with model-class semantics given by

$$\begin{aligned} Sig[SP'_1] &= Sig[SP_1] & Mod[SP'_1] &= Mod[\langle Sig[SP_1], \mathcal{T}'(SP_1)\rangle] \\ &\vdots & &\vdots \\ Sig[SP'_n] &= Sig[SP_n] & Mod[SP'_n] &= Mod[\langle Sig[SP_n], \mathcal{T}'(SP_n)\rangle], \end{aligned}$$

https://doi.org/10.1017/S0960129513000212 Published online by Cambridge University Press

and with property-oriented meaning given by

$$\mathcal{T}_{new}(SP'_1) = \mathcal{T}'(SP_1)$$
$$\vdots$$
$$\mathcal{T}_{new}(SP'_n) = \mathcal{T}'(SP_n).$$

Since $\mathcal{T}'$ is persistently sound and compositional, there is a sound and compositional property-oriented semantics $\widehat{\mathcal{T}'}$ that extends $\mathcal{T}'$ and $\mathcal{T}_{new}$ to $Spec(\{SP'_1,\ldots,SP'_n\})$. Then, using compositionality of $\widehat{\mathcal{T}'}$, its soundness, the definition of the model-class semantics of the new constants, closed-completeness of $\mathcal{T}$ and extensiveness (which follows from flat-completeness, implied by closed-completeness) and monotonicity of $\mathcal{T}$ in turn, we get

$$\begin{aligned}\mathcal{T}'(\mathbf{sbo}(SP_1,\ldots,SP_n)) &= \widehat{\mathcal{T}'}(\mathbf{sbo}(SP'_1,\ldots,SP'_n))\\ &\subseteq Th(\mathbf{sbo}(SP'_1,\ldots,SP'_n))\\ &= Th(\mathbf{sbo}(\langle Sig[SP_1],\mathcal{T}'(SP_1)\rangle,\ldots,\langle Sig[SP_n],\mathcal{T}'(SP_n)\rangle))\\ &= \mathcal{T}(\mathbf{sbo}(\langle Sig[SP_1],\mathcal{T}'(SP_1)\rangle,\ldots,\langle Sig[SP_n],\mathcal{T}'(SP_n)\rangle))\\ &\subseteq \mathcal{T}(\mathbf{sbo}(SP_1,\ldots,SP_n)).\end{aligned}$$

This completes the induction step and thus the proof of the theorem. □

**Corollary 5.8.** $\mathcal{T}_{\mathbf{INS}}$ is at least as strong as any persistently sound and compositional property-oriented semantics for structured specifications built from flat specifications using union, translation and hiding.

*Proof.* The statement follows directly from Proposition 4.2 and Theorem 5.7. □

## 6. Entailment systems

The notion of an institution as recalled in Section 2 captures model-theoretic aspects of logical systems. An institution is typically augmented by an entailment system that approximates the semantic consequence relation, and in this section we consider the consequences of the results given above in this setting. Entailment systems are normally defined with reference to a set of proof rules, but the presentation here will abstract away from this level of detail.

An *entailment relation* on a set $\mathbb{S}$ of sentences is a binary relation $\vdash \subseteq \mathcal{P}(\mathbb{S})\times\mathbb{S}$ satisfying the following properties:

*reflexivity*: $\{\varphi\} \vdash \varphi$;
*weakening*: if $\Phi \vdash \varphi$, then $\Phi \cup \Psi \vdash \varphi$; and
*transitivity*: if $\Phi \vdash \psi$ and $\Psi_\varphi \vdash \varphi$ for each $\varphi \in \Phi$, then $\bigcup_{\varphi\in\Phi} \Psi_\varphi \vdash \psi$

for all sentences $\varphi, \psi \in \mathbb{S}$ and sets of sentences $\Phi, \Psi \subseteq \mathbb{S}$ and $\Psi_\varphi \subseteq \mathbb{S}$ for $\varphi \in \Phi$.

Clearly, the semantic consequence relation defined in Section 2 is an entailment relation in the above sense.

Let **Sen** : **Sign** → **Set** be a functor. An *entailment system* for **Sen** is a family of entailment relations

$$\mathcal{E} = \langle \vdash_\Sigma \subseteq \mathcal{P}(\mathbf{Sen}(\Sigma)) \times \mathbf{Sen}(\Sigma)\rangle_{\Sigma \in |\mathbf{Sign}|}$$

such that for each morphism $\sigma : \Sigma \to \Sigma'$ in **Sign**, each sentence $\varphi \in \mathbf{Sen}(\Sigma)$ and each set $\Phi \subseteq \mathbf{Sen}(\Sigma)$,

$$\text{if } \Phi \vdash_\Sigma \varphi, \text{then } \mathbf{Sen}(\sigma)(\Phi) \vdash_{\Sigma'} \mathbf{Sen}(\sigma)(\varphi),$$

where $\mathbf{Sen}(\sigma)(\Phi)$ denotes the image of $\Phi$ under $\mathbf{Sen}(\sigma)$.

Given an institution

$$\mathbf{INS} = \langle \mathbf{Sign}, \mathbf{Sen}, \mathbf{Mod}, \langle \models_\Sigma \rangle_{\Sigma \in |\mathbf{Sign}|}\rangle,$$

an *entailment system for* **INS** (Meseguer 1989; Harper *et al.* 1994) is an entailment system

$$\mathcal{E} = \langle \vdash_\Sigma \rangle_{\Sigma \in |\mathbf{Sign}|}$$

for **Sen** that is sound with respect to semantic consequence, that is, for each signature $\Sigma$, each $\Sigma$-sentence $\varphi \in \mathbf{Sen}(\Sigma)$ and each set $\Phi \subseteq \mathbf{Sen}(\Sigma)$, if $\Phi \vdash_\Sigma \varphi$, then $\Phi \models_\Sigma \varphi$. Such an entailment system $\mathcal{E}$ is *complete for* **INS** if the opposite implication holds. Clearly, for any institution **INS**, the semantic consequence relations form an entailment system

$$\mathcal{E}_{\mathbf{INS}} = \langle \models_\Sigma \rangle_{\Sigma \in |\mathbf{Sign}|},$$

which is sound and complete for **INS**.

A *general logic* (Meseguer 1989) is an institution **INS** equipped with an entailment system $\mathcal{E}$ for **INS**.

For the rest of this section, we let $\mathcal{E} = \langle \vdash_\Sigma \rangle_{\Sigma \in |\mathbf{Sign}|}$ be an arbitrary entailment system for **Sen** : **Sign** → **Set**.

For any signature $\Sigma \in |\mathbf{Sign}|$, a set of sentences $\Phi \subseteq \mathbf{Sen}(\Sigma)$ is an *$\mathcal{E}$-theory* if it is closed under $\vdash_\Sigma$: that is, if $\Phi \vdash_\Sigma \varphi$, then $\varphi \in \Phi$ for all $\varphi \in \mathbf{Sen}(\Sigma)$. For any set $\Phi \subseteq \mathbf{Sen}(\Sigma)$, the least *$\mathcal{E}$-theory* that contains $\Phi$ will be denoted by $Cl^{\mathcal{E}}_\Sigma(\Phi)$. Clearly, for any institution **INS** and its semantic entailment system $\mathcal{E}_{\mathbf{INS}}$, we have $Cl^{\mathcal{E}_{\mathbf{INS}}}_\Sigma(\_)$ coincides with $Cl_\Sigma(\_)$, and $\mathcal{E}_{\mathbf{INS}}$-theories are exactly the theories in **INS** as defined in Section 2.

An entailment system

$$\mathcal{E} = \langle \vdash_\Sigma \rangle_{\Sigma \in |\mathbf{Sign}|}$$

for **Sen** : **Sign** → **Set** is *trivial* if for each signature $\Sigma \in |\mathbf{Sign}|$, we have

$$\vdash_\Sigma = \mathcal{P}(\mathbf{Sen}(\Sigma)) \times \mathbf{Sen}(\Sigma)$$

(each set entails all sentences).

**Proposition 6.1.** Given an entailment system $\mathcal{E}$, if $\mathcal{E}$ is non-trivial, there is an institution $\mathbf{INS}_0$ such that $\mathcal{E}$ is a (sound) entailment system for $\mathbf{INS}_0$, but $\mathcal{E}$ is not complete for $\mathbf{INS}_0$. If $\mathcal{E}$ is trivial, it is complete for any institution for which it is sound.

*Proof.* A non-trivial entailment system is incomplete for an institution $\mathbf{INS}_0$ in which all categories of models are empty; more interesting institutions $\mathbf{INS}_0$ can be constructed as well. The other part is trivial. □

https://doi.org/10.1017/S0960129513000212 Published online by Cambridge University Press

**Proposition 6.2.** For any entailment system $\mathcal{E}$ there is an institution $\mathbf{INS}_{\mathcal{E}}$ such that $\mathcal{E}$ is (sound and) complete for $\mathbf{INS}_{\mathcal{E}}$.

*Proof.* Consider an entailment system

$$\mathcal{E} = \langle \vdash_\Sigma \rangle_{\Sigma \in |\mathbf{Sign}|}$$

for $\mathbf{Sen}\colon \mathbf{Sign} \to \mathbf{Set}$. For any signature $\Sigma \in |\mathbf{Sign}|$, we define $\Sigma$-models to be $\mathcal{E}$-theories and satisfaction to be membership: that is,

$$\mathbf{Mod}(\Sigma) = \{\Phi \subseteq \mathbf{Sen}(\Sigma) \mid Cl^{\mathcal{E}}_{\Sigma}(\Phi) = \Phi\}$$

(considered as a discrete category), and then for $M \in \mathbf{Mod}(\Sigma)$ and $\varphi \in \mathbf{Sen}(\Sigma)$, we define $M \models_\Sigma \varphi$ to hold if and only if $\varphi \in M$. Furthermore, for any signature morphism $\sigma : \Sigma \to \Sigma'$, we define the reduct to be the coimage with respect to translation of sentences: that is, for $M' \in \mathbf{Mod}(\Sigma')$,

$$M'|_\sigma = \sigma^{-1}(M').$$

By preservation of entailment in $\mathcal{E}$ along signature morphisms, it follows that indeed $M'|_\sigma \in \mathbf{Mod}(\Sigma)$, and the satisfaction condition holds trivially. This defines an institution

$$\mathbf{INS}_{\mathcal{E}} = \langle \mathbf{Sign}, \mathbf{Sen}, \mathbf{Mod}, \langle \models_\Sigma \rangle_{\Sigma \in |\mathbf{Sign}|} \rangle.$$

Now, given any set of $\Sigma$-sentences $\Phi \subseteq \mathbf{Sen}(\Sigma)$ and $\Sigma$-sentence $\varphi \in \mathbf{Sen}(\Sigma)$, we have $\Phi \models \varphi$ in $\mathbf{INS}_{\mathcal{E}}$ means that for all $M \in \mathbf{Mod}(\Sigma)$, if $\Phi \subseteq M$, then $\varphi \in M$, which is equivalent to $\varphi \in M_\Phi$, where $M_\Phi = Cl^{\mathcal{E}}_{\Sigma}(\Phi)$ is the least model in $\mathbf{Mod}(\Sigma)$ that contains $\Phi$. Hence, $\Phi \models \varphi$ in $\mathbf{INS}_{\mathcal{E}}$ if and only if $\Phi \vdash_\Sigma \varphi$. □

As noted in Section 3, the syntax of flat specifications and union, translation and hiding introduced there for an arbitrary institution **INS** depends only on the category of signatures with distinguished class $\mathcal{H}$ and $\mathcal{W}$ of signature morphisms satisfying the requirements imposed in Section 3, and the sentence functor $\mathbf{Sen}\colon \mathbf{Sign} \to \mathbf{Set}$. Consequently, we can consider such specifications whenever we are just given an entailment system $\mathcal{E}$ for **Sen** together with appropriate $\mathcal{H}$ and $\mathcal{W}$, rather than an entire institution. In particular, the signature $Sig[SP]$ of any specification $SP \in Spec^{UTH}$ is then well defined.

The concept of a property-oriented semantics carries over directly to this framework: as in Section 4, a property-oriented semantics is a function $\mathcal{T}$ that maps any specification $SP$ to a set of $\Sigma$-sentences

$$\mathcal{T}(SP) \subseteq \mathbf{Sen}(Sig[SP]).$$

Given such a property-oriented semantics, the definitions of its monotonicity, compositionality and extensiveness carry over in a similarly straightforward way. We say that $\mathcal{T}$ is *$\mathcal{E}$-theory-oriented* if $\mathcal{T}(SP)$ is an $\mathcal{E}$-theory for all specifications $SP$.

However, concepts related to the model-theoretic part of the institution require more care.

https://doi.org/10.1017/S0960129513000212 Published online by Cambridge University Press

A property-oriented semantics $\mathcal{T}$ is $\mathcal{E}$*-sound* if it is sound in any institution **INS** (with the same signature category and sentence functor as for $\mathcal{E}$) for which $\mathcal{E}$ is sound (or, equivalently, in any general logic with $\mathcal{E}$ as the entailment system).

A sound property-oriented semantics $\mathcal{T}$ is $\mathcal{E}$*-complete* in a class of institutions $\mathcal{INS}$ if it is complete in any institution $\mathbf{INS} \in \mathcal{INS}$ (with the same signature category and sentence functor as for $\mathcal{E}$) for which $\mathcal{E}$ is sound and complete. $\mathcal{E}$*-closed-completeness* and $\mathcal{E}$*-flat-completeness* may be defined analogously, but we will not use these concepts here.

$\mathcal{E}$-completeness is perhaps a weaker notion than we might expect: we might have required completeness of the semantics in any institution $\mathbf{INS} \in \mathcal{INS}$ for which $\mathcal{E}$ is sound but not necessarily complete, and thus in any general logic with $\mathcal{E}$ as the entailment system. However, it can be shown from Proposition 6.1 that such a stronger property would not be achievable at all unless the entailment system were trivial (or a very narrow class $\mathcal{INS}$ is considered).

The definition of the standard compositional theory-oriented semantics for specifications in $Spec^{UTH}$ only requires an obvious tiny adjustment:

$$\begin{aligned}
\mathcal{T}_{\mathcal{E}}(\langle \Sigma, \Phi \rangle) &= Cl^{\mathcal{E}}_{\Sigma}(\Phi) \\
\mathcal{T}_{\mathcal{E}}(SP \cup SP') &= Cl^{\mathcal{E}}_{Sig[SP]}(\mathcal{T}_{\mathcal{E}}(SP) \cup \mathcal{T}_{\mathcal{E}}(SP')) \\
\mathcal{T}_{\mathcal{E}}(SP \ \mathbf{with}\ \sigma : Sig[SP] \to \Sigma) &= Cl^{\mathcal{E}}_{\Sigma}(\sigma(\mathcal{T}_{\mathcal{E}}(SP))) \\
\mathcal{T}_{\mathcal{E}}(SP \ \mathbf{hide\ via}\ \sigma : \Sigma \to Sig[SP]) &= \sigma^{-1}(\mathcal{T}_{\mathcal{E}}(SP)).
\end{aligned}$$

**Proposition 6.3.** $\mathcal{T}_{\mathcal{E}}$ is an $\mathcal{E}$-sound $\mathcal{E}$-theory-oriented semantics for specifications built from flat specifications using union, translation and hiding. It is monotone, compositional and extensive. □

As in the case of $\mathcal{T}_{\mathbf{INS}}$ in Section 4, completeness does not hold unless the class of institutions (general logics) considered is subject to further requirements.

**Corollary 6.4.** Let $\mathcal{INS}$ be the class of institutions that are $\langle \mathcal{H}, \mathcal{W} \rangle$-exact and admit parameterised $\langle \mathcal{H}, \mathcal{W} \rangle$-interpolation. Then $\mathcal{T}_{\mathcal{E}}$ is $\mathcal{E}$-complete for specifications built from flat specifications using union, translation and hiding in the class $\mathcal{INS}$.

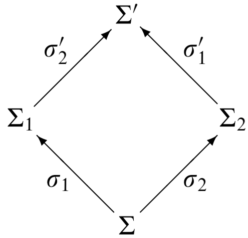
*Proof.* The statement follows from Theorem 4.4. □

Interpolation properties may be directly defined for an entailment system

$$\mathcal{E} = \langle \vdash_{\Sigma} \rangle_{\Sigma \in |\mathbf{Sign}|}$$

https://doi.org/10.1017/S0960129513000212 Published online by Cambridge University Press

for a sentence functor **Sen** : **Sign** → **Set**, without reference to the underlying institution. Specifically, consider again the following commuting diagram in **Sign**:




This diagram *admits parameterised* (or *Craig–Robinson*) *interpolation* if for any $\Phi_1 \subseteq$ **Sen**$(\Sigma_1)$, $\Phi_2 \subseteq$ **Sen**$(\Sigma_2)$ and $\varphi \in$ **Sen**$(\Sigma_2)$, if

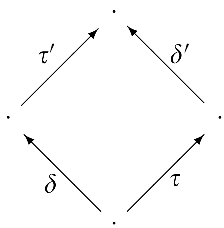
$$\sigma_2'(\Phi_1) \cup \sigma_1'(\Phi_2) \vdash_{\Sigma'} \sigma_1'(\varphi),$$

then for some $\Phi \subseteq$ **Sen**$(\Sigma)$ such that $\Phi_1 \vdash_{\Sigma_1} \sigma_1(\Phi)$, we have

$$\Phi_2 \cup \sigma_2(\Phi) \vdash_{\Sigma_2} \varphi.$$

The diagram *admits Craig interpolation* if it admits parameterised interpolation with 'parameter set' $\Phi_2 = \emptyset$.

Given classes $\mathcal{H}, \mathcal{W} \subseteq$ **Sign** of signature morphisms, we say that $\mathcal{E}$ *admits parameterised* (respectively, *Craig*) $\langle \mathcal{H}, \mathcal{W} \rangle$*-interpolation* if for any signature morphisms $\delta \in \mathcal{H}$ and $\tau \in \mathcal{W}$ with a common source, there are $\delta' \in \mathcal{H}$ and $\tau' \in \mathcal{W}$ forming a pushout in **Sign**




that admits parameterised (respectively, Craig) interpolation. Then any such pushout also admits parameterised (respectively, Craig) interpolation as well.

Clearly, if $\mathcal{E}$ is (sound and) complete for an institution **INS**, the above interpolation properties for $\mathcal{E}$ coincide with those for **INS** as defined in Section 2.

So, for an entailment system that admits parameterised $\langle \mathcal{H}, \mathcal{W} \rangle$-interpolation, the semantics $\mathcal{T}_{\mathcal{E}}$ is $\mathcal{E}$-complete for the class of institutions that are $\langle \mathcal{H}, \mathcal{W} \rangle$-exact.

Even though $\mathcal{T}_{\mathcal{E}}$ is not $\mathcal{E}$-complete in general, it is, in essence, the strongest compositional $\mathcal{E}$-theory oriented semantics.

**Corollary 6.5.** $\mathcal{T}_{\mathcal{E}}$ is at least as strong as any $\mathcal{E}$-sound, compositional, extensive $\mathcal{E}$-theory-oriented semantics for structured specifications built from flat specifications using union, translation and hiding.

*Proof.* Let $\mathcal{T}$ be an $\mathcal{E}$-sound, compositional, extensive $\mathcal{E}$-theory-oriented semantics. Consider the institution $\mathbf{INS}_{\mathcal{E}}$, where semantic entailment coincides with entailment in $\mathcal{E}$, as given by Proposition 6.2. Then $\mathcal{T}_{\mathcal{E}}$ coincides with $\mathcal{T}_{\mathbf{INS}_{\mathcal{E}}}$, and $\mathcal{T}$ is sound in $\mathbf{INS}_{\mathcal{E}}$ (as

https://doi.org/10.1017/S0960129513000212 Published online by Cambridge University Press

well as being compositional and extensive). Consequently, $\mathcal{T}_{\mathcal{E}}$ is at least as strong as $\mathcal{T}$ by Corollary 5.2. □

We cannot drop the assumption that the semantics under consideration is extensive. To see this, we can adapt Example 5.3 to the framework of an entailment system as follows.

**Example 6.6.** Consider an entailment system $\mathcal{E}^0$ that consists of semantic consequence for the institution constructed in Example 5.3. That is, we take the category of signatures $\mathbf{Sign}_0$ with exactly two objects $\Sigma$ and $\Sigma'$, and $\sigma : \Sigma \to \Sigma'$ as the only non-identity morphism. The sentence functor is given by

$$\mathbf{Sen}_0(\Sigma) = \{\varphi\}$$
$$\mathbf{Sen}_0(\Sigma') = \{\varphi, \psi\},$$

with $\sigma$-translation preserving $\varphi$. We now define $\mathcal{E}^0$ as the least entailment system for $\mathbf{Sen}_0$ such that $\psi \vdash^0_{\Sigma'} \varphi$.

Now consider the property-oriented semantics $\mathcal{T}'$ defined in Example 5.3. It is $\mathcal{E}^0$-sound (by the same reasoning as in Example 5.3), compositional and $\mathcal{E}^0$-theory-oriented. Moreover, for $SP_{bad}$ defined as

$$\langle \Sigma', \{\psi\}\rangle \textbf{ hide via } \sigma,$$

we have

$$\mathcal{T}'(SP_{bad} \textbf{ with } \sigma) = \{\varphi, \psi\},$$

while

$$\mathcal{T}_{\mathcal{E}^0}(SP_{bad} \textbf{ with } \sigma) = \{\varphi\}.$$

Similarly, the assumption that the semantics under consideration be $\mathcal{E}$-theory-oriented cannot be dropped, since Example 5.4 can also be adapted here as follows.

**Example 6.7.** Consider an entailment system $\mathcal{E}^1$ that consists of semantic consequence for the institution constructed in Example 5.4. That is, take the category of signatures $\mathbf{Sign}_1$ with exactly two objects $\Sigma$ and $\Sigma'$, and $\sigma : \Sigma \to \Sigma'$ as the only non-identity morphism. The sentence functor is given by

$$\mathbf{Sen}_1(\Sigma) = \{\varphi, \varphi'\}$$
$$\mathbf{Sen}_1(\Sigma') = \{\varphi, \varphi', \psi\},$$

with $\sigma$-translation preserving $\varphi$ and $\varphi'$. We now define $\mathcal{E}^1$ as the least entailment system for $\mathbf{Sen}_1$ such that

$$\psi \vdash^1_{\Sigma'} \varphi$$
$$\varphi \vdash^1_{\Sigma} \varphi'.$$

Now consider the property-oriented semantics $\mathcal{T}''$ defined in Example 5.4. It is $\mathcal{E}^1$-sound (for the same reason as in Example 5.4), compositional and extensive. Moreover, for $SP_{bad}$ defined as

$$\langle \Sigma', \{\psi\}\rangle \textbf{ hide via } \sigma,$$

https://doi.org/10.1017/S0960129513000212 Published online by Cambridge University Press

we have

$$\mathcal{T}''(SP_{bad} \textbf{ with } \sigma) = \{\psi, \varphi, \varphi'\},$$

while

$$\mathcal{T}_{\mathcal{E}^1}(SP_{bad} \textbf{ with } \sigma) = \{\varphi, \varphi'\}.$$

As in Section 5, this counterexample semantics has to be non-monotone.

**Corollary 6.8.** $\mathcal{T}_{\mathcal{E}}$ is at least as strong as any $\mathcal{E}$-sound, monotone and extensive property-oriented semantics for structured specifications built from flat specifications using union, translation and hiding.

*Proof.* Let $\mathcal{T}$ be a $\mathcal{E}$-sound, monotone and extensive property-oriented semantics. Consider the institution $\mathbf{INS}_{\mathcal{E}}$ where semantic entailment coincides with entailment in $\mathcal{E}$, as given by Proposition 6.2. Then $\mathcal{T}_{\mathcal{E}}$ coincides with $\mathcal{T}_{\mathbf{INS}_{\mathcal{E}}}$, and $\mathcal{T}$ is sound in $\mathbf{INS}_{\mathcal{E}}$ (as well as being monotone and extensive). Consequently, $\mathcal{T}_{\mathcal{E}}$ is at least as strong as $\mathcal{T}$ by Corollary 5.6. □

Clearly, by Example 6.6, we cannot drop the requirement that the semantics considered in the corollary be extensive since the semantics $\mathcal{T}'$ given there was monotone.

We conclude this section by comparing semantics in different entailment systems. Of course, if $\mathcal{E}$ is not at least as strong as another entailment system $\mathcal{E}'$ used to give a semantics for specifications, we cannot expect $\mathcal{T}_{\mathcal{E}}$ to be at least as strong as this other semantics; typically this would not hold even for flat specifications. However, we do have the following proposition.

**Proposition 6.9.** Consider the following two entailment systems for $\mathbf{Sen} : \mathbf{Sign} \to \mathbf{Set}$:

$$\mathcal{E} = \langle \vdash_\Sigma \rangle_{\Sigma \in |\mathbf{Sign}|}$$
$$\mathcal{E}' = \langle \vdash'_\Sigma \rangle_{\Sigma \in |\mathbf{Sign}|}.$$

Suppose $\mathcal{E}$ is at least as strong as $\mathcal{E}'$, that is, for each signature $\Sigma \in |\mathbf{Sign}|$, we have $\vdash'_\Sigma \subseteq \vdash_\Sigma$ (that is, all $\mathcal{E}'$-consequences of any set of sentences are also its $\mathcal{E}$-consequences). Then $\mathcal{T}_{\mathcal{E}}$ is at least as strong as $\mathcal{T}_{\mathcal{E}'}$ for specifications built from flat specifications by union, translation and hiding: that is, $\mathcal{T}_{\mathcal{E}'}(SP) \subseteq \mathcal{T}_{\mathcal{E}}(SP)$ for all $SP \in Spec^{UTH}$.

*Proof.* The proof is by an easy induction on the structure of specifications. □

**Corollary 6.10.** $\mathcal{T}_{\mathcal{E}}$ is at least as strong as any semantics for specifications built from flat specifications using union, translation and hiding that is compositional, extensive, $\mathcal{E}'$-theory-oriented and $\mathcal{E}'$-sound for some entailment system $\mathcal{E}'$ such that $\mathcal{E}$ is at least as strong as $\mathcal{E}'$.

*Proof.* The statement follows from Corollary 6.5 and Proposition 6.9. □

It may be considered somewhat unsatisfactory for us to require that the semantics we compare $\mathcal{T}_{\mathcal{E}}$ with is $\mathcal{E}'$-sound, rather than just $\mathcal{E}$-sound. However, this cannot be weakened, since the counterexample semantics $\mathcal{T}''$ given in Example 6.7 is $\mathcal{E}'$-theory-oriented, for instance, for the least entailment system $\mathcal{E}'$ generated by $\psi \vdash'_{\Sigma'} \varphi$ (and any system that is even weaker than such $\mathcal{E}'$).

However, as we discussed at the end of Section 5, the possibly unexpected requirements on the property-oriented semantics considered may be dropped if a stronger version of soundness and compositionality is assumed that persist when the specification framework is extended.

A property-oriented semantics $\mathcal{T}$ for specifications built using some specification-building operations in the context of an entailment system $\mathcal{E}$ is said to be *$\mathcal{E}$-persistently sound and compositional* if it is persistently sound and compositional in any institution **INS** for which $\mathcal{E}$ is sound.

**Corollary 6.11.** $\mathcal{T}_{\mathcal{E}}$ is at least as strong as any $\mathcal{E}$-persistently sound and compositional property-oriented semantics for structured specifications built from flat specifications using union, translation and hiding.

*Proof.* By Corollary 5.8, if we consider the institution $\mathbf{INS}_{\mathcal{E}}$ where semantic entailment coincides with entailment in $\mathcal{E}$, as given by Proposition 6.2, and $\mathcal{T}_{\mathcal{E}}$ coincides with $\mathcal{T}_{\mathbf{INS}_{\mathcal{E}}}$. □

## 7. Final remarks

We have studied property-oriented semantics for structured specifications in the context of an arbitrary institution, and then in the context of an arbitrary entailment system.

Considering specifications built from flat specifications using union, translation and hiding, we explained why the standard compositional property-oriented semantics given in Section 4 cannot be improved. On the one hand, we have sharpened the standard result (Borzyszkowski 2002) that this semantics is complete in any exact institution with an appropriate interpolation property (*cf.* Theorem 4.4). On the other hand, we have also shown that it is at least as strong as any other sound, compositional, extensive theory-oriented semantics, as well as any other sound, monotone, extensive property-oriented semantics (*cf.* Corollaries 5.2 and 5.6). These two results follow from more general theorems that state similar results for specifications built using an arbitrary collection of specification-building operations (*cf.* Theorems 5.1 and 5.5).

We have also given counterexamples that show that the unexpected and counter-intuitive requirements of extensiveness (the semantics under consideration must not 'forget' about axioms in flat specifications[†]) and theory-orientedness (the semantics must take regard of consequences of the properties derived) cannot be dropped in general (*cf.* Examples 5.3 and 5.4). However, they become superfluous if we require a stronger form of soundness and compositionality, that persist when the specification formalism is extended by new specification constants with arbitrary sound semantics (*cf.* Theorem 5.7 and Corollary 5.8). It is worth noting that a similar effect may be achieved if, instead of adding new specification constants, we require that the property-oriented semantics

[†] For this reason, extensiveness is called *non-absent-mindedness* in Sannella and Tarlecki (2012).

for structured specifications translated by any institution comorphism (Meseguer 1989; Tarlecki 2000; Goguen and Roşu 2002) extends in a sound and compositional way to structured specifications in the richer institution.

These results and counterexamples improve significantly on related results in Sannella and Tarlecki (2012). They carry over to the context of specifications built from flat specifications using union, translation and hiding in the context of an entailment system – see Section 6. The results also apply, *mutatis mutandis*, to any specification language that has at least the expressive power provided by these simple operations.

Although we have only discussed property-oriented semantics here, there is an intimate link between proof systems and property-oriented semantics that makes the results immediately applicable to proof systems as well. For instance, the standard compositional property-oriented semantics $\mathcal{T}_{\mathbf{INS}}$ for structured specifications built from flat specifications using union, translation and hiding in an institution **INS** given in Section 4 may be presented using the well-known proof system

$$\frac{}{\langle \Sigma, \Phi \rangle \vdash \varphi} \quad \varphi \in \Phi$$

$$\frac{SP_1 \vdash \varphi}{SP_1 \cup SP_2 \vdash \varphi} \qquad \frac{SP_2 \vdash \varphi}{SP_1 \cup SP_2 \vdash \varphi}$$

$$\frac{SP \vdash \varphi}{SP \ \mathbf{with}\ \sigma \vdash \sigma(\varphi)} \qquad \frac{SP \vdash \sigma(\varphi)}{SP \ \mathbf{hide\ via}\ \sigma \vdash \varphi}$$

together with the following rule to link consequences of specifications with semantic consequence in the underlying institution:

$$\frac{SP \vdash \varphi \text{ for each } \varphi \in \Phi \qquad \Phi \models \psi}{SP \vdash \psi} \quad (\models \text{closure})$$

Clearly, for any specification $SP \in Spec^{UTH}$ and $Sig[SP]$-sentence $\varphi$, we have that $\varphi \in \mathcal{T}_{\mathbf{INS}}(SP)$ if and only if $SP \vdash \varphi$ can be derived in the above proof system. The standard compositional property-oriented semantics $\mathcal{T}_{\mathcal{E}}$ for structured specifications built from flat specifications using union, translation and hiding in an entailment system $\mathcal{E}$ given in Section 6 may be presented by essentially the same proof system with the final rule ($\models$ closure) replaced by

$$\frac{SP \vdash \varphi \text{ for each } \varphi \in \Phi \qquad \Phi \vdash_{Sig[SP]} \psi}{SP \vdash \psi} \quad (\vdash \text{closure})$$

In this sense, any proof system for proving consequences of specifications generates a property-oriented semantics. (Note that this is different from proof systems for proving entailment between properties in the context of a structured specification, as studied in Diaconescu (2008, Section 14.2).)

On the one hand then, the notions we introduced for property-oriented semantics, like soundness, completeness, closed-completeness, compositionality, monotonicity, and so on, may be directly applied to proof systems. In particular, the proof system given by the rules above is sound, theory-oriented, monotone, compositional, extensive, flat-complete and closed-complete for the specification-building operations considered. The main results presented here for property-oriented semantics carry over to proof systems as well, and can be recast as establishing that the above proof system for consequences of structured specifications built from flat specifications using union, translation and hiding is the strongest compositional one possible. Improving on it requires compositionality to be sacrificed: for example, a non-compositional proof system that is stronger may be given by the following single rule, using Theorem 3.1:

$$\frac{\Phi' \vdash_{\Sigma'} \sigma(\varphi)}{SP \vdash \varphi} \quad \mathsf{nf}(SP) = \langle \Sigma', \Phi' \rangle \ \textbf{hide via} \ \sigma.$$

Another non-compositional approach, which uses additional axioms and rules that are derived from the form of the specification in question, can be found in Hennicker *et al.* (1997).

On the other hand, we may want to study formats of proof systems that ensure desirable properties of the semantics they generate. For instance, if all the proof rules in a proof system derive consequences of structured specifications from consequences of their immediate constituents, then the corresponding property-oriented semantics is compositional. Monotonicity follows if, furthermore, none of the proof rules involves 'negative' premises. Finally, the property-oriented semantics given by a proof system is theory-oriented (respectively, $\mathcal{E}$-theory-oriented) if and only if the rule ($\models$ closure) (respectively, ($\vdash$ closure)) is admissible.


## Acknowledgements

We wish to express our thanks to the anonymous reviewers for their perceptive comments, and, in particular, to one of them for reminding us to explain the relationship of our work to Goguen and Roşu (2004).


## References


Bergstra, J. A., Heering, J. and Klint, P. (1990) Module algebra. *Journal of the Association for Computing Machinery* **37** (2) 335–372.

Bidoit, M. and Mosses, P. D. (eds.) (2004) CASL User Manual. *Springer-Verlag Lecture Notes in Computer Science* **2900**. (Also available at `http://www.informatik.uni-bremen.de/cofi/wiki/index.php/CASL`.)

Borzyszkowski, T. (2002) Logical systems for structured specifications. *Theoretical Computer Science* **286** (2) 197–245.

Borzyszkowski, T. (2005) Generalized interpolation in first order logic. *Fundamenta Informaticae* **66** (3) 199–219.

Burstall, R. M. and Goguen, J. A. (1977) Putting theories together to make specifications. In: *Fifth International Joint Conference on Artificial Intelligence, Boston* 1045–1058.

Burstall, R. M. and Goguen, J. A. (1980) The semantics of Clear, a specification language. In: Bjørner, D. (ed.) Proceedings of the 1979 Copenhagen Winter School on Abstract Software Specification. *Springer-Verlag Lecture Notes in Computer Science* **86** 292–332.

Burstall, R. M. and Goguen, J. A. (1981) An informal introduction to specifications using Clear. In: Boyer, R. S. and Moore, J. S. (eds.) *The Correctness Problem in Computer Science*, Academic Press 185–213. (Also in: Gehani, N. and McGettrick, A. D. (eds.) (1986) *Software Specification Techniques*, Addison-Wesley.)

Chang, C.-C. and Keisler, H. J. (1990) *Model Theory*, third edition, North-Holland.

Diaconescu, R. (2008) *Institution-Independent Model Theory*, Birkhäuser.

Diaconescu, R., Goguen, J. A. and Stefaneas, P. (1993) Logical support for modularisation. In: Huet, G. and Plotkin, G. (eds.) *Logical Environments*, Cambridge University Press 83–130.

Ehrig, H., Wagner, E. G. and Thatcher, J. W. (1983) Algebraic specifications with generating constraints. In: Proceedings of the 10th International Colloquium on Automata, Languages and Programming, Barcelona. *Springer-Verlag Lecture Notes in Computer Science* **154** 188–202.

Goguen, J. A. and Burstall, R. M. (1992) Institutions: Abstract model theory for specification and programming. *Journal of the Association for Computing Machinery* **39** (1) 95–146.

Goguen, J. A. and Roşu, G. (2002) Institution morphisms. *Formal Aspects of Computing* **13** (3-5) 274–307.

Goguen, J. A. and Roşu, G. (2004) Composing hidden information modules over inclusive institutions. In: From Object-Orientation to Formal Methods. Essays in Memory of Ole-Johan Dahl. *Springer-Verlag Lecture Notes in Computer Science* **2635** 96–123.

Harper, R., Sannella, D. and Tarlecki, A. (1994) Structured presentations and logic representations. *Annals of Pure and Applied Logic* **67** 113–160.

Hennicker, R., Wirsing, M. and Bidoit, M. (1997) Proof systems for structured specifications with observability operators. *Theoretical Computer Science* **173** (2) 393–443.

MacQueen, D. and Sannella, D. (1985) Completeness of proof systems for equational specifications. *IEEE Transactions on Software Engineering* **SE-11** (5) 454–461.

Meseguer, J. (1989) General logics. In: Ebbinghaus, H.-D. (ed.) *Logic Colloquium '87, Granada*, North-Holland 275–329.

Mossakowski, T., Autexier, S. and Hutter, D. (2006) Development graphs – proof management for structured specifications. *Journal of Logic and Algebraic Programming* **67** (1-2) 114–145.

Mosses, P. D. (ed.) (2004) Casl Reference Manual. *Springer-Verlag Lecture Notes in Computer Science* **2960**.

Rodenburg, P. H. (1991) A simple algebraic proof of the equational interpolation theorem. *Algebra Universalis* **28** 48–51.

Roşu, G. and Goguen, J. A. (2000) On equational Craig interpolation. *Journal of Universal Computer Science* **6** (1) 194–200.

Sannella, D. and Tarlecki, A. (1988) Specifications in an arbitrary institution. *Information and Computation* **76** (2-3) 165–210.

Sannella, D. and Tarlecki, A. (2012) *Foundations of Algebraic Specification and Formal Software Development*, Monographs in Theoretical Computer Science, an EATCS Series.

Sannella, D. and Wirsing, M. (1983) A kernel language for algebraic specification and implementation. In: Karpinski, M. (ed.) Proceedings of the 1983 International Conference on Foundations of Computation Theory, Borgholm. *Springer-Verlag Lecture Notes in Computer Science* **158** 413–427.

Sannella, D., Sokołowski, S. and Tarlecki, A. (1992) Toward formal development of programs from algebraic specifications: Parameterisation revisited. *Acta Informatica* **29** (8) 689–736.

Tarlecki, A. (1986) Bits and pieces of the theory of institutions. In: Pitt, D. H., Abramsky, S., Poigné, A. and Rydeheard, D. E. (eds.) Proceedings of the Tutorial and Workshop on Category Theory and Computer Programming, Guildford. *Springer-Verlag Lecture Notes in Computer Science* **240** 334–360.

Tarlecki, A. (2000) Towards heterogeneous specifications. In: Gabbay, D. and de Rijke, M. (eds.) *Frontiers of Combining Systems 2 Studies in Logic and Computation* **7**, Research Studies Press 337–360.

Tarlecki, A. (2011) Some nuances of many-sorted universal algebra: A review. *Bulletin of the European Association for Theoretical Computer Science* **104** 89–111.