Chapter 1

(Heterogeneous) Structured Specifications in Logics Without Interpolation

Carlos G. Lopez Pombo[1]
clpombo@dc.uba.ar
Universidad de Buenos Aires. Facultad de Ciencias Exactas y Naturales. Departamento de Computación and CONICET–Universidad de Buenos Aires. Instituto de Investigación en Ciencias de la Computación (ICC).
and
Marcelo F. Frias
mfrias@itba.edu.ar
Department of Software Engineering, Buenos Aires Institute of Technology (ITBA) and Consejo Nacional de Investigaciones Científicas y Tecnológicas (CONICET)

Abstract The world of software development has become intrinsically heterogeneous. Many formal languages have been made available to help analysts and designers model different aspects of software. Some examples in the logic realm are equational logic and classical first-order logic, propositional temporal logics such as LTL and CTL (and their first-order versions), multimodal logics such as the dynamic logic PDL and its first-order version, etc.
One important feature of a specification language is the existence of structuring mechanisms enabling the modular construction of system descriptions. Structured specifications were introduced by Wirsing for first-order logic, and later presented in the language-independent setting of institutions by Sannella and Tarlecki. Afterwards, Borzyszkowski presented sufficient conditions for a calculus for (homogeneous) structured specifications to be complete. These conditions include some form of Craig's interpolation, which results in a scenario that excludes many formalisms employed in the description of software.
The contributions of this article are then summarised as follows: *a*) We present a calculus for structured specifications whose completeness proof does not require any form of interpolation. *b*) We extend this calculus to a complete calculus for heterogeneous structured specifications.

Key words: Structured specifications, Heterogeneous specifications, Institutions

1 Introduction and Motivation

Many languages and notations have been designed with the aim of helping software analysts and designers capture and model different aspects of software development. Among the formal approaches, logics have always been a distinguished tool in software specification, analysis and verification. In (Goguen & Burstall, 1984), Goguen and Burstall present *institutions* as a categorical formalization of the abstract model theory of a logical system. Institutions provide an abstract view of a logic that enables the study of properties of a formalism independently of notational issues. For instance, (Tarlecki, 1986) surveys several interesting results about well-known properties, such as interpolation, within the framework of institutions.

Of utmost importance in software development is the composition of partial models of software into complete, consistent ones. Since the foundational work of Parnas (Parnas, 1972, 1979), practitioners build software artifacts (and particularly software specifications), modularly. In (Sannella & Tarlecki, 1988) Sannella and Tarlecki provide a set of structure-building operations that enable the modular construction of specifications from theories taken from a given institution. They also propose a set of rules that enable reason-ing in terms of the modules involved in the design. In a different direction (but with the same purpose), Bergstra et al. propose in (Bergstra, Heering, & Klint, 1990) an algebraic formalization of modules recalling, for the first time, that in order to have a complete calculus for modular specifications, Craig's interpolation property is required (in this particular case expressed in an implicit, but equivalent way, by the presence of two axioms). Wirsing presents in (Wirsing, 1991) a calculus similar to the one given in (Sannella & Tarlecki, 1988), but restricted to structured specifications whose constituent parts are written in first-order predicate logic. This calculus is proved to be complete in the absence of hidden symbols. Interpolation is explicitly used in the completeness proof. Borzyszkowski (Borzyszkowski, 1997) presents a logical system for the structure building operations (SBOs) introduced by Sannella and Tarlecki, as well as an extensive discussion on the conditions under which the proposed calculus is complete. One of these conditions is that the underlying institution must either satisfy Craig's interpolation, or a combination of a weaker form of interpolation with other properties such as compactness. In (Dimitrakos & Maibaum, 2000) Dimitrakos and Maibaum show some links between restrictions on the collection of morphisms acting over signatures (i.e., stability of faithful morphisms under pushouts), and Craig's interpolation; and in (Diaconescu, Goguen, & Stefaneas, 1993), Diaconescu and Goguen show that whenever the logical system is compact, Craig's interpolation is equivalent to certain distributive laws.

Also in (Goguen & Burstall, 1984), Goguen and Burstall call the attention on the diversity of languages used in computer science:

> "There is a population explosion among the logical systems used in computer science. Examples include first-order logic, equational logic, Horn-clause logic, higher-order

> logic, infinitary logic, dynamic logic, intuitionistic logic, order sorted logic, and temporal logic; moreover, there is a tendency for each theorem prover to have its own idiosyncratic logical system."

Institutions provide the formal machinery needed to present the notion of logic (from a model-theoretical point of view), in an abstract and compact way. If we consider the work on structured specifications under the light of this phrase, Borzyszkowski's work provides general conditions for the existence of a complete calculus for structured specifications over a given institution. Unfortunately, most of the logics used in computer science to describe system behaviors (linear time temporal logics like LTL(Pnueli, 1981) and its first-order version FOLTL(Manna & Pnueli, 1995), branching-time temporal logics like CTL(Clarke, Emerson, & Sistla, 1986; Emerson & Halpern, 1985) and CTL^*(Pnueli, 1977; Emerson & Halpern, 1986), and dynamic logics such as PDL(Harel, 2001; Harel, Kozen, & Tiuryn, 2000) and its first-order counterpart FODL(Harel et al., 2000)), do not comply with these conditions.

First Contribution: We present a calculus for structured specifications whose completeness is subject to weaker properties (more specifically, no form of Craig's interpolation is required), enabling its use in the verification of properties of structured specifications in the previously mentioned logics.

Modeling languages such as the Unified Modeling Language (UML) (Booch, Rumbaugh, & Jacobson, 1998) allow us to model a system using a combination of diagrammatic notations. Each diagram provides a (partial) view of the system under development. This view-centric approach to software modeling has two clear advantages: *a*) decentralization of the modeling process (several engineers may be modeling different views of the same system simultaneously), and *b*) separation of concerns is enforced. This approach to software modeling requires the existence of mechanisms for integrating these partial views in a complete description of the system. Institutions support mechanisms for dealing with the heterogeneity arising from choosing different languages or different aspects of a software system. In (Tarlecki, 1996), one of the most influential papers on moving specifications between logical systems, Tarlecki wrote:

> "... this suggests that we should strive at a development of a convenient to use proof theory (with support tools!) for a sufficiently rich "universal" institution, and then reuse it for other institutions linked to it by institution representations."

In (Mossakowski & Tarlecki, 2009) Mossakowski and Tarlecki define the concept of *heterogeneous logical environment*, which results in a handy tool in the formalization of the "universal" approach we pursue. We will also discuss other approaches, those presented in (Diaconescu & Futatsugi, 2002; Mossakowski, Maeder, & Luttich, 2007; Tarlecki, 2000; Cengarle, Knapp, Tarlecki, & Wirsing, 2008), in Sec. 4.1, where we propose an extension of our calculus for structured specifications capable of dealing with heterogeneous structured specifications.

In most of these approaches the integration of partial descriptions written in different languages is carried out by resorting to semantics-preserving mappings between institutions. Among these, co-morphisms of institutions expose a very natural relation between logical systems because they show how a possibly less expressive logic can be interpreted into a richer one. In (Borzyszkowski, 1998, 2002), Borzyszkowski extended his work on structured specifications and proved that, under appropriate conditions, structured specifications over a given logic can be translated into structured specifications in another logic, provided a co-morphism between the institutions exists.

Second Contribution: We present an extension of the calculus for structured specifications to settings in which heterogeneous specifications are mapped to a "universal" institution.

The remaining parts of the article are organized as follows. In Sec. 2 we provide basic definitions (including central notions such as that of institution, entailment system, and structure building operations), as well as Borzyszkowski's calculus. In Sec. 3 we develop one of the main contributions of this article by analyzing the calculus proposed by Borzyszkowski and discussing its possibilities and limitations. We show a modified version of Borzyszkowski's calculus that is complete and requires weaker conditions, thus providing a complete calculus for many logics ubiquitous in software modeling. In Sec. 4.1 we present Borzyszkowski's calculus for heterogeneous structured specifications, and discuss its limitations. In Sec. 4.2 we extend our calculus in order to deal with structured heterogeneous specifications related via institution representations. Finally, in Sec. 5, we draw some conclusions.

2 Institutions and Structured Specifications

The theory of institutions was introduced by Goguen and Burstall in (Goguen & Burstall, 1984). Institutions provide a formal and generic definition of logical system, and allow one to describe ways in which specifications in a logical system can be structured (Sannella & Tarlecki, 1988). Institutions have evolved in a number of directions, from an abstract theory of software specification and development (Tarlecki, 2003) to a general version of abstract model theory (Diaconescu, 2008), and offered a suitable formal framework for addressing heterogeneity (Mossakowski et al., 2007; Tarlecki, 2000), including applications to the UML (Cengarle et al., 2008).

In this section we present the basic definitions and results we will use throughout the rest of the paper.

Definition 1 (Entailment system (Meseguer, 1989)).
A structure $\left\langle \mathsf{Sign}, \mathbf{Sen}, \{\vdash^{\Sigma}\}_{\Sigma \in |\mathsf{Sign}|} \right\rangle$ is said to be an *entailment system* if it satisfies the following conditions:

- Sign is a category of signatures,
- $\mathbf{Sen} : \mathsf{Sign} \to \mathsf{Set}$ is a functor (let $\Sigma \in |\mathsf{Sign}|$, then $\mathbf{Sen}(\Sigma)$ is the set of Σ-sentences),
- $\{\vdash^{\Sigma}\}_{\Sigma \in |\mathsf{Sign}|}$, where $\vdash^{\Sigma} \subseteq 2^{\mathbf{Sen}(\Sigma)} \times \mathbf{Sen}(\Sigma)$, is a family of binary relations such that for any $\Sigma, \Sigma' \in |\mathsf{Sign}|$, $\phi \in \mathbf{Sen}(\Sigma)$, $\{\phi_i\}_{i \in \mathcal{I}} \subseteq \mathbf{Sen}(\Sigma)$ and $\Gamma, \Gamma' \subseteq \mathbf{Sen}(\Sigma)$ the following conditions are satisfied:

  1. reflexivity: $\{\phi\} \vdash^{\Sigma} \phi$,
  2. monotonicity: if $\Gamma \vdash^{\Sigma} \phi$ and $\Gamma \subseteq \Gamma'$, then $\Gamma' \vdash^{\Sigma} \phi$,
  3. transitivity: if $\Gamma \vdash^{\Sigma} \phi_i$ for all $i \in \mathcal{I}$ and $\{\phi_i\}_{i \in \mathcal{I}} \vdash^{\Sigma} \phi$, then $\Gamma \vdash^{\Sigma} \phi$, and
  4. $\vdash$-translation: if $\Gamma \vdash^{\Sigma} \phi$, then for any morphism $\sigma : \Sigma \to \Sigma'$ in Sign, $\mathbf{Sen}(\sigma)(\Gamma) \vdash^{\Sigma'} \mathbf{Sen}(\sigma)(\phi)$.

Definition 2. Let $\left\langle \mathsf{Sign}, \mathbf{Sen}, \{\vdash^{\Sigma}\}_{\Sigma \in |\mathsf{Sign}|} \right\rangle$ be an entailment system. Its category of theories (denoted by Th), is a structure $\langle \mathcal{O}, \mathcal{A} \rangle$ such that:

- $\mathcal{O} = \{ \langle \Sigma, \Gamma \rangle \mid \Sigma \in |\mathsf{Sign}| \text{ and } \Gamma \subseteq \mathbf{Sen}(\Sigma) \}$, and
- $\mathcal{A} = \left\{ \sigma : \langle \Sigma, \Gamma \rangle \to \langle \Sigma', \Gamma' \rangle \;\middle|\; \begin{array}{l} \langle \Sigma, \Gamma \rangle , \langle\ \Sigma', \Gamma'\ \rangle \in \mathcal{O}, \\ \sigma : \Sigma \to \Sigma' \text{ is a morphism in } \mathsf{Sign} \text{ and} \\ \text{for all } \gamma \in \Gamma, \Gamma' \vdash^{\Sigma'} \mathbf{Sen}(\sigma)(\gamma) \end{array} \right\}$.

Definition 3 (Institution (Goguen & Burstall, 1992)). A structure $\left\langle \mathsf{Sign}, \mathbf{Sen}, \mathbf{Mod}, \{\models^{\Sigma}\}_{\Sigma \in |\mathsf{Sign}|} \right\rangle$ is said to be an *institution* if it satisfies the following conditions:

- Sign is a category of signatures,
- $\mathbf{Sen} : \mathsf{Sign} \to \mathsf{Set}$ is a functor (let $\Sigma \in |\mathsf{Sign}|$, then $\mathbf{Sen}(\Sigma)$ returns the set of Σ-sentences),
- $\mathbf{Mod} : \mathsf{Sign}^{\mathsf{op}} \to \mathsf{Cat}$ is a functor (let $\Sigma \in |\mathsf{Sign}|$, then $\mathbf{Mod}(\Sigma)$ returns the category of Σ-models), and
- $\{\models^{\Sigma}\}_{\Sigma \in |\mathsf{Sign}|}$, where $\models^{\Sigma} \subseteq |\mathbf{Mod}(\Sigma)| \times \mathbf{Sen}(\Sigma)$, is a family of binary relations,

such that for any signature morphism $\sigma : \Sigma \to \Sigma'$, Σ-sentence $\phi \in \mathbf{Sen}(\Sigma)$ and Σ'-model $\mathcal{M}' \in |\mathbf{Mod}(\Sigma)|$, the following $\models$-invariance condition holds:[2]

$$\mathcal{M}' \models^{\Sigma'} \mathbf{Sen}(\sigma)(\phi) \quad \text{iff} \quad \mathbf{Mod}(\sigma^{\mathsf{op}})(\mathcal{M}') \models^{\Sigma} \phi \ .$$

Let $\Sigma \in |\mathsf{Sign}|$, $\Gamma \subseteq \mathbf{Sen}(\Sigma)$ and let $T = \langle \Sigma, \Gamma \rangle \in |\mathsf{Th}|$, then we define the category $\mathbf{Mod}(T)$ as the full subcategory of $\mathbf{Mod}(\Sigma)$ determined by those models $\mathcal{M} \in |\mathbf{Mod}(\Sigma)|$ such that for all $\gamma \in \Gamma$, $\mathcal{M} \models^{\Sigma} \gamma$. In addition, it is possible to extend the relation $\models^{\Sigma}$ to sets of sentences and a sentence as follows: $\Gamma \models^{\Sigma} \alpha$ if and only if $\mathcal{M} \models^{\Sigma} \alpha$, for all $\mathcal{M} \in |\mathbf{Mod}(\Sigma, \Gamma)|$.

From now on, whenever we make a reference to an institution (resp. entailment system) $\mathbb{I}$, we will assume the structure we are referring to is of the

[2] Given $\sigma : \Sigma \to \Sigma'$ a morphism in Sign, the corresponding morphism in the opposite category $\mathsf{Sign}^{\mathsf{op}}$ will be denoted as σ^{op}.

form $\left\langle \mathsf{Sign}^{\mathbb{I}}, \mathbf{Sen}^{\mathbb{I}}, \mathbf{Mod}^{\mathbb{I}}, \{\models^{\mathbb{I}^{\Sigma}}\}_{\Sigma \in |\mathsf{Sign}^{\mathbb{I}}|} \right\rangle$ (resp. $\left\langle \mathsf{Sign}^{\mathbb{I}}, \mathbf{Sen}^{\mathbb{I}}, \{\vdash^{\mathbb{I}^{\Sigma}}\}_{\Sigma \in |\mathsf{Sign}^{\mathbb{I}}|} \right\rangle$) univocally determining the components of the structure.

Next, we formalise some well-known properties of models.

Definition 4 (Conjunction (Borzyszkowski, 2002)). An institution $\mathbb{I}$ is said to have *conjunction* if for all $\Sigma \in |\mathsf{Sign}^{\mathbb{I}}|$, finite set of formulas $\{\varphi_i\}_{i \in \mathcal{I}} \subseteq \mathbf{Sen}^{\mathbb{I}}(\Sigma)$ and $\mathcal{M} \in |\mathbf{Mod}^{\mathbb{I}}(\Sigma)|$, there exists a formula $\psi \in \mathbf{Sen}^{\mathbb{I}}(\Sigma)$ (usually denoted as $\bigwedge_{i \in \mathcal{I}} \varphi_i$) such that: $\mathcal{M} {\models^{\mathbb{I}^{\Sigma}}} \psi$ iff for all $i \in \mathcal{I}$, $\mathcal{M} {\models^{\mathbb{I}^{\Sigma}}} \varphi_i$. $\mathbb{I}$ is said to have *infinite conjunction* if $\{\varphi_i\}_{i \in \mathcal{I}} \subseteq \mathbf{Sen}^{\mathbb{I}}(\Sigma)$ may be infinite.

Definition 5 (Negation (Borzyszkowski, 2002)). An institution $\mathbb{I}$ is said to have *negation* if for all $\Sigma \in |\mathsf{Sign}^{\mathbb{I}}|$, finite set of formulas $\{\varphi_i\}_{i \in \mathcal{I}} \subseteq \mathbf{Sen}^{\mathbb{I}}(\Sigma)$ and $\mathcal{M} \in |\mathbf{Mod}^{\mathbb{I}}(\Sigma)|$, there exists a formula $\psi \in \mathbf{Sen}^{\mathbb{I}}(\Sigma)$ (usually denoted as $\neg\varphi$) such that: $\mathcal{M} {\models^{\mathbb{I}^{\Sigma}}} \psi$ iff it is not true that $\mathcal{M} {\models^{\mathbb{I}^{\Sigma}}} \varphi$.

Definition 6 (Implication (Borzyszkowski, 2002)). An institution $\mathbb{I}$ is said to have *implication* if for all $\Sigma \in |\mathsf{Sign}^{\mathbb{I}}|$, finite set of formulas $\{\varphi_i\}_{i \in \mathcal{I}} \subseteq \mathbf{Sen}^{\mathbb{I}}(\Sigma)$ and $\mathcal{M} \in |\mathbf{Mod}^{\mathbb{I}}(\Sigma)|$, there exists a formula $\psi \in \mathbf{Sen}^{\mathbb{I}}(\Sigma)$ (usually denoted as $\varphi \implies \varphi'$) such that: $\mathcal{M} {\models^{\mathbb{I}^{\Sigma}}} \psi$ iff $\mathcal{M} {\models^{\mathbb{I}^{\Sigma}}} \varphi$ implies $\mathcal{M} {\models^{\mathbb{I}^{\Sigma}}} \varphi'$.

Fact 1 *If an institution has negation and conjunction, it has implication.*

Definition 7 (Compactness (Borzyszkowski, 2002)). An institution $\mathbb{I}$ is said to be *compact* if for all $\Sigma \in |\mathsf{Sign}^{\mathbb{I}}|$, $\alpha \in \mathbf{Sen}^{\mathbb{I}}(\Sigma)$ and $\Gamma \subseteq \mathbf{Sen}^{\mathbb{I}}(\Sigma)$ such that $\Gamma {\models^{\mathbb{I}^{\Sigma}}} \alpha$, there exists $\Gamma' \subseteq \mathbf{Sen}^{\mathbb{I}}(\Sigma)$ such that $\Gamma' \subseteq \Gamma$, Γ' is finite and $\Gamma' {\models^{\mathbb{I}^{\Sigma}}} \alpha$.

Definition 8 (Logic (Meseguer, 1989)).
A structure $\left\langle \mathsf{Sign}, \mathbf{Sen}, \mathbf{Mod}, \{\vdash^{\Sigma}\}_{\Sigma \in |\mathsf{Sign}|}, \{\models^{\Sigma}\}_{\Sigma \in |\mathsf{Sign}|} \right\rangle$ is said to be a *logic* if it satisfies the following conditions:

- $\left\langle \mathsf{Sign}, \mathbf{Sen}, \{\vdash^{\Sigma}\}_{\Sigma \in |\mathsf{Sign}|} \right\rangle$ is an entailment system,
- $\left\langle \mathsf{Sign}, \mathbf{Sen}, \mathbf{Mod}, \{\models^{\Sigma}\}_{\Sigma \in |\mathsf{Sign}|} \right\rangle$ is an institution, and
- the following *soundness* condition is satisfied: for any $\Sigma \in |\mathsf{Sign}|$, $\phi \in \mathbf{Sen}(\Sigma)$, and $\Gamma \subseteq \mathbf{Sen}(\Sigma)$,

$$\Gamma \vdash^{\Sigma} \phi \text{ implies } \Gamma \models^{\Sigma} \phi\ .$$

A logic is said to be *complete* if in addition the following condition is also satisfied: for any $\Sigma \in |\mathsf{Sign}|$, $\phi \in \mathbf{Sen}(\Sigma)$, and $\Gamma \subseteq \mathbf{Sen}(\Sigma)$,

$$\Gamma \models^{\Sigma} \phi \text{ implies } \Gamma \vdash^{\Sigma} \phi\ .$$

We provide next some definitions that will be necessary in further sections.

Definition 9 (Interpolation and weak interpolation (Borzyszkowski, 2002)). An institution $\mathbb{I}$ is said to have the *interpolation property* if for any pushout $\langle t'_1 : \Sigma_1 \to \Sigma', t'_2 : \Sigma_2 \to \Sigma' \rangle$ for $\langle t_1 : \Sigma \to \Sigma_1, t_2 : \Sigma \to \Sigma_2 \rangle$ in $\mathsf{Sign}^{\mathbb{I}}$, and $\varphi_i \in \mathbf{Sen}^{\mathbb{I}}(\Sigma_i)$ for $i = 1, 2$, if $\mathbf{Sen}^{\mathbb{I}}(t'_1)(\varphi_1) \models^{\mathbb{I}^{\Sigma'}} \mathbf{Sen}^{\mathbb{I}}(t'_2)(\varphi_2)$, there exists $\varphi \in \mathbf{Sen}^{\mathbb{I}}(\Sigma)$ (called the interpolant of φ_1 and φ_2) such that $\varphi_1 \models^{\mathbb{I}^{\Sigma_1}} \mathbf{Sen}^{\mathbb{I}}(t_1)(\varphi)$ and $\mathbf{Sen}^{\mathbb{I}}(t_2)(\varphi) \models^{\mathbb{I}^{\Sigma_2}} \varphi_2$. In a similar way, $\mathbb{I}$ is said to have the *weak interpolation property* if whenever $\mathbf{Sen}^{\mathbb{I}}(t'_1)(\varphi_1) \models^{\mathbb{I}^{\Sigma'}} \mathbf{Sen}^{\mathbb{I}}(t'_2)(\varphi_2)$, then there exists $\Gamma \subseteq \mathbf{Sen}^{\mathbb{I}}(\Sigma)$ (called the interpolant of φ_1 and φ_2) such that $\varphi_1 \models^{\mathbb{I}^{\Sigma_1}} \mathbf{Sen}^{\mathbb{I}}(t_1)(\Gamma)$ and $\mathbf{Sen}^{\mathbb{I}}(t_2)(\Gamma) \models^{\mathbb{I}^{\Sigma_2}} \varphi_2$.

The original statement of interpolation for first-order logic (Craig, 1957, Lemma 1), due to Craig, states that whenever a property φ written in a language L_1 follows from a set of formulas Γ written in a (possibly different) language L_2, there exists a formula ψ (called the *interpolant*), that belongs to $L_1 \cap L_2$ and serves as a bridge between Γ and φ, i.e., $\Gamma \models \psi$ and $\psi \models \varphi$. Definition 9 states the same property but formalized as a category-theoretical construction in which the intersection of the languages is represented as a span (McLane, 1971) in the category of signatures.

Definition 10 (Weak amalgamation (Borzyszkowski, 2002)). An institution $\mathbb{I}$ is said to have the *weak amalgamation property* if for any pushout $\langle t'_1 : \Sigma_1 \to \Sigma', t'_2 : \Sigma_2 \to \Sigma' \rangle$ for $\langle t_1 : \Sigma \to \Sigma_1, t_2 : \Sigma \to \Sigma_2 \rangle$ in $\mathsf{Sign}^{\mathbb{I}}$ and for any models $\mathcal{M}_1 \in |\mathbf{Mod}^{\mathbb{I}}(\Sigma_1)|$ and $\mathcal{M}_2 \in |\mathbf{Mod}^{\mathbb{I}}(\Sigma_2)|$ such that $\mathbf{Mod}^{\mathbb{I}}({t_1}^{\mathsf{op}})(\mathcal{M}_1) = \mathbf{Mod}^{\mathbb{I}}({t_2}^{\mathsf{op}})(\mathcal{M}_2)$, there exists $\mathcal{M}' \in |\mathbf{Mod}^{\mathbb{I}}(\Sigma')|$ such that $\mathbf{Mod}^{\mathbb{I}}({t'_1}^{\mathsf{op}})(\mathcal{M}') = \mathcal{M}_1$ and $\mathbf{Mod}^{\mathbb{I}}({t'_2}^{\mathsf{op}})(\mathcal{M}') = \mathcal{M}_2$.

In order to understand the weak amalgamation property, let us consider two models from (possibly different) languages L_1 and L_2 related by a span in the category of signatures. Let us also assume that the models have a common reduct in $L_1 \cap L_2$. Then, the models are reducts of a model of the language of the pushout for the span. Although in (Borzyszkowski, 2002) there is no concrete explanation of why the property is referred to as *weak amalgamation property*, and the author only says the definition is inspired in the classic definition of *amalgamation property*, one can speculate that it is due to the dropping of the requirement of t_1 and t_2 to be injective.

In many works (cf. (Borzyszkowski, 2002) and specially (Sannella & Tarlecki, 2014)) interpolation and amalgamation are treated as relative properties by only requiring the existence of pushouts for a subclass of spans in the category of signatures receiving the name of parameterised Craig's interpolation. There, institutions are generalised by introducing $(\mathcal{D}, \mathcal{T})$-institutions, whose class of signature morphisms is partitioned into those used to translate specifications, leading to specifications over a richer set of symbols (i.e. $\mathcal{T}$-morphisms) and those used to derive them, leading to specifications in which some of the symbols were hidden (i.e. $\mathcal{D}$-morphisms). Other works, like (Diaconescu, 2008), explore a more general form of interpolation (Craig-Robinson's interpolation). In this work we will stick to a simpler, and more absolute, version of interpolation (i.e. the usual Craig's interpolation formulation) and amalgamation properties, leading to a more classical understanding of the results but keeping in mind that the aforementioned relativisation can be done without invalidating any of the results presented in the forthcoming sections, specially considering that certain relativisation in this direction will be useful in proving Prop. 4.

The definitions and results appearing in the remaining of this section were orginally introduced in (Borzyszkowski, 2002).

Definition 11 (Structure building operations (Borzyszkowski, 2002)). The class of specifications over a logic $\mathbb{I}$ for a given signature $\Sigma \in |\mathsf{Sign}^{\mathbb{I}}|$, denoted as $\mathsf{Spec}^{\mathbb{I}}_{\Sigma}$, and the operators **Sig** and **Mod**, are defined as follows:

- Any pair $\langle \Sigma, \Gamma \rangle$, where $\Sigma \in |\mathsf{Sign}^{\mathbb{I}}|$ and $\Gamma \subseteq \mathbf{Sen}^{\mathbb{I}}(\Sigma)$ is a specification (called *flat specification* or *presentation*), such that:

$$\mathbf{Sig}[\langle \Sigma, \Gamma \rangle] = \Sigma, \text{ and } \mathbf{Mod}[\langle \Sigma, \Gamma \rangle] = |\mathbf{Mod}^{\mathbb{I}}(\Sigma, \Gamma)|.$$

- Let $\Sigma \in |\mathsf{Sign}^{\mathbb{I}}|$ and $SP_1, SP_2 \in \mathsf{Spec}^{\mathbb{I}}_{\Sigma}$. Then, $SP_1 \cup SP_2 \in \mathsf{Spec}^{\mathbb{I}}_{\Sigma}$ is such that:

$$\mathbf{Sig}[SP_1 \cup SP_2] = \Sigma, \text{ and } \mathbf{Mod}[SP_1 \cup SP_2] = \mathbf{Mod}[SP_1] \cap \mathbf{Mod}[SP_2].$$

- Let $\Sigma, \Sigma' \in |\mathsf{Sign}^{\mathbb{I}}|$, $SP \in \mathsf{Spec}^{\mathbb{I}}_{\Sigma}$ and $\sigma : \Sigma \to \Sigma' \in ||\mathsf{Sign}^{\mathbb{I}}||$. Then, **translate** SP **by** $\sigma \in \mathsf{Spec}^{\mathbb{I}}_{\Sigma'}$ is such that:

$$\begin{aligned} &\mathbf{Sig}[\textbf{translate } SP \textbf{ by } \sigma] = \Sigma', \text{ and} \\ &\mathbf{Mod}[\textbf{translate } SP \textbf{ by } \sigma] = \left\{ \mathcal{M}' \;\middle|\; \mathbf{Mod}^{\mathbb{I}}(\sigma^{\mathsf{op}})(\mathcal{M}') \in \mathbf{Mod}[SP] \right\}. \end{aligned}$$

- Let $\Sigma, \Sigma' \in |\mathsf{Sign}^{\mathbb{I}}|$, $SP \in \mathsf{Spec}^{\mathbb{I}}_{\Sigma'}$ and $\sigma : \Sigma \to \Sigma' \in ||\mathsf{Sign}^{\mathbb{I}}||$. Then, **derive from** SP **by** $\sigma \in \mathsf{Spec}^{\mathbb{I}}_{\Sigma}$ such that:

$$\begin{aligned} &\mathbf{Sig}[\textbf{derive from } SP \textbf{ by } \sigma] = \Sigma, \text{ and} \\ &\mathbf{Mod}[\textbf{derive from } SP \textbf{ by } \sigma] = \left\{ \mathbf{Mod}^{\mathbb{I}}(\sigma^{\mathsf{op}})(\mathcal{M}') \;\middle|\; \mathcal{M}' \in \mathbf{Mod}[SP] \right\}. \end{aligned}$$

The operations introduced in Def. 11 are referred to as *structure building operations*, or SBOs, and provide a mechanism to put specifications together in a structured way. The operators **Sig** and **Mod** help us retrieve both the signature and the corresponding class of models for a given structured specification. Intuitively, $SP_1 \cup SP_2$ is a specification that contains the axioms of SP_1 and SP_2. Similarly, **translate** SP **by** σ is a specification in which axioms are (syntactically) translated according to morphism σ. Finally, specification **derive from** SP **by** σ can be understood as characterizing reducts (according to σ) of models of SP.

Since $\mathbf{Mod}[SP]$ is a class of models, we define $\mathbf{Mod}[SP] \models^{\mathbb{I}^{\Sigma}} \alpha$ if for all $\mathcal{M} \in \mathbf{Mod}[SP]$, $\mathcal{M} \models^{\mathbb{I}^{\Sigma}} \alpha$. Also, we will use the notation $\models^{\mathbb{I}}{}_{\Sigma}$ to denote the satisfaction relation between structured specifications and formulas. Recall that $\models^{\mathbb{I}^{\Sigma}}$ (notice the notational difference with $\models^{\mathbb{I}}{}_{\Sigma}$), denotes the satisfaction relation of the underlying institution $\mathbb{I}$.

Definition 12. Let $\mathbb{I}$ be a logic, $\Sigma \in |\mathsf{Sign}^{\mathbb{I}}|$, $SP \in \mathsf{Spec}^{\mathbb{I}}_{\Sigma}$ and $\alpha \in \mathbf{Sen}^{\mathbb{I}}(\Sigma)$. α is a semantic consequence of SP (denoted $SP {\models^{\mathbb{I}}}_{\Sigma} \alpha$) if $\mathbf{Mod}[SP] \models^{\mathbb{I}^{\Sigma}} \alpha$.

Definition 13. Let $\mathbb{I}$ be an logic, $\Sigma \in |\mathsf{Sign}^{\mathbb{I}}|$ and $SP_1, SP_2 \in \mathsf{Spec}^{\mathbb{I}}_{\Sigma}$, we say that SP_1 is equivalent to SP_2 (denoted $SP_1 \equiv^{\mathbb{I}} SP_2$) if $\mathbf{Sig}[SP_1] = \mathbf{Sig}[SP_2]$ and $\mathbf{Mod}[SP_1] = \mathbf{Mod}[SP_2]$.

Definition 14 (Normal form (Borzyszkowski, 2002)). Let $\mathbb{I}$ be a logic and $\Sigma \in |\mathsf{Sign}^{\mathbb{I}}|$, then $SP \in \mathsf{Spec}^{\mathbb{I}}_{\Sigma}$ is in *normal form* if it has the form **derive from** $\langle \Sigma', \Gamma' \rangle$ **by** σ, where $\Sigma' \in |\mathsf{Sign}^{\mathbb{I}}|$, $\sigma : \Sigma \to \Sigma' \in ||\mathsf{Sign}^{\mathbb{I}}||$ and $\Gamma' \subseteq \mathbf{Sen}^{\mathbb{I}}(\Sigma')$.

Given a structured specification, its normal form is obtained by the application of the operator **nf** (Borzyszkowski, 2002, Def. 3.7). The intuition behind operator **nf** is that it flattens the specification by translating the axioms to the "richest" signature using pushouts in $\mathsf{Sign}^{\mathbb{I}}$, followed by the derivation of the resulting flat specification to a signature having only those symbols that must remain visible.

Thus, from now on we will only consider institutions whose category of signatures has pushouts.

Definition 15 (nf operation (Borzyszkowski, 2002)). Let $\mathbb{I}$ be a logic. We define **nf** as follows:

- If SP is a flat specification $\langle \Sigma, \Gamma \rangle$, with $\Sigma \in |\mathsf{Sign}^{\mathbb{I}}|$ and $\Gamma \subseteq \mathbf{Sen}^{\mathbb{I}}(\Sigma)$, then
$$\mathbf{nf}(SP) = \textbf{derive from } \langle \Sigma, \Gamma \rangle \textbf{ by } id_{\Sigma},$$
- Let $\Sigma \in |\mathsf{Sign}^{\mathbb{I}}|$ and $SP_1, SP_2 \in \mathsf{Spec}^{\mathbb{I}}_{\Sigma}$ then, whenever $SP \in \mathsf{Spec}^{\mathbb{I}}_{\Sigma}$ of the form $SP_1 \cup SP_2$ with normal forms $\mathbf{nf}(SP_i) = \textbf{derive from } \langle \Sigma_i, \Gamma_i \rangle \textbf{ by } \sigma_i$, for $i = 1, 2$, we define:
$$\mathbf{nf}(SP) = \textbf{derive from } \left\langle \Sigma', \mathbf{Sen}^{\mathbb{I}}(\sigma'_1)(\Gamma_1) \cup \mathbf{Sen}^{\mathbb{I}}(\sigma'_2)(\Gamma_2) \right\rangle \textbf{ by } \sigma,$$
such that $\sigma = \sigma_1 \circ \sigma'_1 = \sigma_2 \circ \sigma'_2$, and $\langle \sigma'_1 : \Sigma_1 \to \Sigma', \sigma'_2 : \Sigma_2 \to \Sigma' \rangle$ is the pushout for $\langle \sigma_1 : \Sigma \to \Sigma_1, \sigma_2 : \Sigma \to \Sigma_2 \rangle$ in $\mathsf{Sign}^{\mathbb{I}}$,
- Let $\Sigma \in |\mathsf{Sign}^{\mathbb{I}}|$ and $SP_1 \in \mathsf{Spec}^{\mathbb{I}}_{\Sigma}$ then, whenever SP is **translate** SP_1 **by** σ and $\mathbf{nf}(SP_1) = \textbf{derive from } \langle \Sigma_1, \Gamma_1 \rangle \textbf{ by } \sigma_1$, we define:
$$\mathbf{nf}(SP) = \textbf{derive from } \langle \Sigma', \mathbf{Sen}(\sigma'_1)(\Gamma_1) \rangle \textbf{ by } \sigma',$$
such that the pair of morphisms $\langle \sigma' : \mathbf{Sig}[SP] \to \Sigma', \sigma'_1 : \Sigma_1 \to \Sigma' \rangle$ is a pushout for $\langle \sigma : \Sigma \to \mathbf{Sig}[SP], \sigma_1 : \Sigma \to \Sigma_1 \rangle$ in $\mathsf{Sign}^{\mathbb{I}}$, and

- Let $\Sigma, \Sigma' \in |\mathsf{Sign}^{\mathbb{I}}|$, $SP' \in \mathsf{Spec}^{\mathbb{I}}_{\Sigma}$ and $\sigma : \Sigma \to \Sigma' \in ||\mathsf{Sign}^{\mathbb{I}}||$ then, whenever $SP \in \mathsf{Spec}^{\mathbb{I}}_{\Sigma}$, **derive from** SP' **by** σ and $\mathbf{nf}(SP') =$ **derive from** $\langle \Sigma_1, \Gamma_1 \rangle$ **by** σ_1, we define:

$$\mathbf{nf}(SP) = \textbf{derive from } \langle \Sigma_1, \Gamma_1 \rangle \textbf{ by } \sigma \circ \sigma_1 .$$

Theorem 1 ((Borzyszkowski, 2002)). *Let SP be a Σ-specification over an institution $\mathbb{I}$. If $\mathbb{I}$ has the weak amalgamation property, then $\mathbf{nf}(SP) \equiv^{\mathbb{I}} SP$.*

In (Borzyszkowski, 2002), Borzyszkowski presented a calculus for structured specifications and gave sufficient conditions for his calculus to be complete. We reproduce Borzyszkowski's calculus in Def. 16. In Sec. 3 we will present a variant of Borzyszkowski's calculus that is proved sound and complete under weaker conditions than those required in (Borzyszkowski, 2002), making the proposed calculus suitable for reasoning about structured specifications in logics ubiquitous in computer science.

Definition 16 ((Borzyszkowski, 2002)). Let $\mathbb{I}$ be a logic. Then, the following rules define a $\mathsf{Sign}^{\mathbb{I}}$-indexed family of entailment relations $\{\vdash^{\mathbb{I}}{}_{\Sigma}\}_{\Sigma \in |\mathsf{Sign}^{\mathbb{I}}|}$:[3]

$$\frac{\Gamma \vdash^{\mathbb{I}^{\Sigma}} \varphi}{\langle \Sigma, \Gamma \rangle \vdash^{\mathbb{I}}{}_{\Sigma} \varphi}\ [\text{basic}] \qquad \frac{\{SP \vdash^{\mathbb{I}}{}_{\Sigma} \psi\}_{\psi \in \Delta} \qquad \Delta \vdash^{\mathbb{I}^{\Sigma}} \varphi}{SP \vdash^{\mathbb{I}}{}_{\Sigma} \varphi}\ [\text{CR}]$$

$$\frac{SP_1 \vdash^{\mathbb{I}}{}_{\Sigma} \varphi}{SP_1 \cup SP_2 \vdash^{\mathbb{I}}{}_{\Sigma} \varphi}\ [\text{sum1}] \qquad \frac{SP_2 \vdash^{\mathbb{I}}{}_{\Sigma} \varphi}{SP_1 \cup SP_2 \vdash^{\mathbb{I}}{}_{\Sigma} \varphi}\ [\text{sum2}]$$

$$\frac{SP \vdash^{\mathbb{I}}{}_{\Sigma'} \mathbf{Sen}^{\mathbb{I}}(\sigma)(\varphi)}{\textbf{derive from } SP \textbf{ by } \sigma \vdash^{\mathbb{I}}{}_{\Sigma} \varphi}\ [\text{derive}] \qquad \frac{SP \vdash^{\mathbb{I}}{}_{\Sigma} \varphi}{\textbf{translate } SP \textbf{ by } \sigma \vdash^{\mathbb{I}}{}_{\Sigma'} \mathbf{Sen}^{\mathbb{I}}(\sigma)(\varphi)}\ [\text{translate}]$$

Lemma 1. *Let $\mathbb{I}$ be a logic and $\Sigma \in |\mathsf{Sign}^{\mathbb{I}}|$. Then, given $SP_1, SP_2 \in \mathsf{Spec}^{\mathbb{I}}_{\Sigma}$ such that $SP_1 \equiv^{\mathbb{I}} SP_2$, and $\alpha \in \mathbf{Sen}^{\mathbb{I}}(\Sigma)$, $SP_1 \vdash^{\mathbb{I}}{}_{\Sigma} \alpha$ if and only if $SP_2 \vdash^{\mathbb{I}}{}_{\Sigma} \alpha$.*

Theorem 2 (Soundness (Borzyszkowski, 2002)). *Let $\mathbb{I}$ be a logic having infinite conjunction and implication, $\Sigma \in |\mathsf{Sign}^{\mathbb{I}}|$, $SP \in \mathsf{Spec}^{\mathbb{I}}_{\Sigma}$ and let $\alpha \in \mathbf{Sen}^{\mathbb{I}}(\Sigma)$. Then,*

$$SP \vdash^{\mathbb{I}}{}_{\Sigma} \alpha \quad \textit{implies} \quad SP \models^{\mathbb{I}}{}_{\Sigma} \alpha \ .$$

Theorem 3 (Completeness (Borzyszkowski, 2002)). *Let $\mathbb{I}$ be a logic having infinite conjunction and implication, $\Sigma \in |\mathsf{Sign}^{\mathbb{I}}|$, $SP \in \mathsf{Spec}^{\mathbb{I}}_{\Sigma}$ and let $\alpha \in \mathbf{Sen}^{\mathbb{I}}(\Sigma)$. Then, if*

1. *$\mathbb{I}$ satisfies interpolation and weak-amalgamation, and*
2. *for all $\Sigma \in |\mathsf{Sign}^{\mathbb{I}}|$, $\vdash^{\mathbb{I}^{\Sigma}}$ is complete for $\models^{\mathbb{I}^{\Sigma}}$,*

$$SP \models^{\mathbb{I}}{}_{\Sigma} \alpha \quad \textit{implies} \quad SP \vdash^{\mathbb{I}}{}_{\Sigma} \alpha \ .$$

[3] Once again, the reader should note the difference between $\vdash^{\mathbb{I}^{\Sigma}}$, the entailment relation associated to $\mathbb{I}$, and $\vdash^{\mathbb{I}}{}_{\Sigma}$, the entailment relation for structured specifications over $\mathbb{I}$.

Several corollaries are derived from Thm. 3. The goal of these corollaries is to present other conditions under which the completeness theorem also holds. The reader may notice that these conditions are, in one way or another, equivalent to the hypotheses of the previous theorem. The equivalences follow from the proof of the previous theorem. The proof follows by induction on the structure of the specification. The case in which the specification is a union requires the use of the interpolation theorem together with infinite conjunction and implication. In this context the problem of not having interpolation can be overcome by combining compactness with conjunction and implication, or by restricting specifications to be finite.

Corollary 1 ((Borzyszkowski, 2002)). *Let $\mathbb{I}$ be a compact logic, $\Sigma \in |\mathsf{Sign}^{\mathbb{I}}|$, $SP \in \mathsf{Spec}^{\mathbb{I}}_{\Sigma}$ and $\alpha \in \mathbf{Sen}^{\mathbb{I}}(\Sigma)$. Then, if*

1. *$\mathbb{I}$ satisfies weak-interpolation and weak-amalgamation, and*
2. *for all $\Sigma \in |\mathsf{Sign}^{\mathbb{I}}|$, $\vdash^{\mathbb{I}^{\Sigma}}$ is complete for $\models^{\mathbb{I}^{\Sigma}}$,*

$$SP \models^{\mathbb{I}}{}_{\Sigma} \alpha \quad \textit{implies} \quad SP \vdash^{\mathbb{I}}{}_{\Sigma} \alpha \ .$$

Corollary 2 ((Borzyszkowski, 2002)). *Let $\mathbb{I}$ be a logic that has infinite conjunction and implication, $\Sigma \in |\mathsf{Sign}^{\mathbb{I}}|$, $SP \in \mathsf{Spec}^{\mathbb{I}}_{\Sigma}$ and $\alpha \in \mathbf{Sen}^{\mathbb{I}}(\Sigma)$. Then, if*

1. *$\mathbb{I}$ satisfies weak-interpolation and weak-amalgamation, and*
2. *for all $\Sigma \in |\mathsf{Sign}^{\mathbb{I}}|$, $\vdash^{\mathbb{I}^{\Sigma}}$ is complete for $\models^{\mathbb{I}^{\Sigma}}$,*

$$SP \models^{\mathbb{I}}{}_{\Sigma} \alpha \quad \textit{implies} \quad SP \vdash^{\mathbb{I}}{}_{\Sigma} \alpha \ .$$

Definition 17. Let $\mathbb{I}$ be a logic, $\Sigma \in |\mathsf{Sign}^{\mathbb{I}}|$, $SP \in \mathsf{Spec}^{\mathbb{I}}_{\Sigma}$ is *finite* if and only if any flat specification $\langle \Sigma, \Gamma \rangle$ occurring as part of SP satisfies that Γ is finite.

Fact 2 *Let $\mathbb{I}$ be a logic, $\Sigma \in |\mathsf{Sign}^{\mathbb{I}}|$, if $SP \in \mathsf{Spec}^{\mathbb{I}}_{\Sigma}$ is finite, then $\mathbf{nf}(SP)$ is also finite.*

Corollary 3 ((Borzyszkowski, 2002)). *Let $\mathbb{I}$ be a logic that has conjunction and implication, $\Sigma \in |\mathsf{Sign}^{\mathbb{I}}|$, $SP \in \mathsf{Spec}^{\mathbb{I}}_{\Sigma}$ finite and $\alpha \in \mathbf{Sen}^{\mathbb{I}}(\Sigma)$. Then, if*

1. *$\mathbb{I}$ satisfies weak-interpolation and weak-amalgamation, and*
2. *for all $\Sigma \in |\mathsf{Sign}^{\mathbb{I}}|$, $\vdash^{\mathbb{I}^{\Sigma}}$ is complete for $\models^{\mathbb{I}^{\Sigma}}$,*

$$SP \models^{\mathbb{I}}{}_{\Sigma} \alpha \quad \textit{implies} \quad SP \vdash^{\mathbb{I}}{}_{\Sigma} \alpha \ .$$

3 Beyond Interpolation

We already mentioned in the introduction that many logics used to describe software behavior do not satisfy Borzyszkowski's conditions for having a complete calculus for structured specifications. In Table 1 we review some interesting results on the satisfaction of these conditions.

	Infinite conjunction	Complete calculus	Compact	Interpolation
PDL	**No**	**Yes, infinitary(7)** (de Lavalette et al., 2008)	**No** (Harel et al., 2000, pp. 181)	**?(1)**
FODL	**No**	**Yes, infinitary(7)** (Harel et al., 2000)	**No** (Harel et al., 2000, pp. 303)	**?(2)**
LTL	**No**	**Yes** (Manna & Pnueli, 1995, pp. 214–231) (Pnueli, 1977) (Abadi & Manna, 1990)	**No(3)**	**No** (Maksimova, 1990)
FOLTL	**No**	**No** (Manna & Pnueli, 1995, pp. 270) (Abadi & Manna, 1990) (Abadi, 1988)	**No(4)**	**No** (Maksimova, 1990)
CTL	**No**	**Yes** (Reynolds, 2001)	**No(5)**	**No** (Maksimova, 1990)
CTL$*$	**No**	**Yes** (Emerson & Halpern, 1985)	**No(6)**	**No** (Maksimova, 1990)

Table 1 Satisfaction of Borzyszkowski's general conditions

(1) In (Kowalski, 2002) Kowalski published a positive result but later in (Kowalski, 2004) published an errata. To our knowledge, the problem remains open.

(2) To our knowledge this problem also remains open.

With respect to **(1)** and **(2)**, it was proved in (Fine, 1979) that quantified S5 modal logic fails to have Craig's interpolation property, contributing some negative insights on the result for PDL and FODL.

(3) Following the construction presented in (Harel et al., 2000, pp. 181, pp. 303) consider the set of formulas $\{\diamond\neg\phi\} \cup \{\phi, \mathsf{X}\phi, \mathsf{XX}\phi, \ldots\}$ which is finitely satisfiable but not satisfiable.

(4) The construction is analogous to the one presented in **(3)**.

(5) Consider the set of formulas $\{\mathsf{E}(\top\mathsf{U}\neg\phi\} \cup \{\phi, \mathsf{AX}\phi, \mathsf{AXAX}\phi, \ldots\}$ which is finitely satisfiable but not satisfiable.

(6) The construction is analogous to the one presented in **(5)**.

(7) In the cases of logics PDL and FODL, the ones that are capable of handling properties of programs involving loops, the calculi are strongly complete under the presence of a potentially infinite set of axioms.

The results reviewed in Table 1 constitute enough evidence that, if one commits to provide modularisation mechanisms for software specifications, as well as to provide a complete calculi for proving properties about them, then an extra effort must be done. Those results show how many logics, ubiquitous in software specification, fail in meeting Borzyszkowski's conditions for the calculus of Def. 16 to be complete, therefore limiting the usefulness of the calculus.

In Def. 18 we introduce a new calculus exploiting the way proofs are developed providing a methodological insight on theorem proving for structured specifications over an institution. With the newly added and/or modified rules we will prove that the resulting calculus is sound and complete. Our calculus differs from the one presented in Def. 16 in two ways: *a*) we added Rule *[equiv]*, allowing the replacement of a specification by another, provided that they are equivalent in the sense of Def. 13, and *b*) Rules *[CR]*, *[sum1]* and *[sum2]* are replaced by a single, slightly more complex, rule for $\cup$ (*[sum]*).

Definition 18. Let $\mathbb{I}$ be a logic. Then, the following rules define a $\mathsf{Sign}^{\mathbb{I}}$-indexed family of entailment relations $\{\vdash^{\mathbb{I}}{}_{\Sigma}\}_{\Sigma\in|\mathsf{Sign}^{\mathbb{I}}|}$:

$$\frac{\Gamma \vdash^{\mathbb{I}^{\Sigma}} \varphi}{\langle \Sigma, \Gamma \rangle \vdash^{\mathbb{I}}_{\Sigma} \varphi}\,[\text{basic}] \qquad \frac{SP_2 \vdash^{\mathbb{I}}_{\Sigma} \varphi \qquad SP_1 \equiv^{\mathbb{I}} SP_2}{SP_1 \vdash^{\mathbb{I}}_{\Sigma} \varphi}\,[\text{equiv}]$$

$$\frac{SP' \vdash^{\mathbb{I}}_{\Sigma'} \mathbf{Sen}(\sigma)(\varphi)}{\mathbf{derive\ from}\ SP'\ \mathbf{by}\ \sigma \vdash^{\mathbb{I}}_{\Sigma} \varphi}\,[\text{derive}] \qquad \frac{SP \vdash^{\mathbb{I}}_{\Sigma} \varphi}{\mathbf{translate}\ SP\ \mathbf{by}\ \sigma \vdash^{\mathbb{I}}_{\Sigma'} \mathbf{Sen}(\sigma)(\varphi)}\,[\text{translate}]$$

$$\frac{\{SP_1 \vdash^{\mathbb{I}}_{\Sigma} \psi\}_{\psi \in \Delta} \qquad \langle \Sigma, \Delta \rangle \cup SP_2 \vdash^{\mathbb{I}}_{\Sigma} \varphi}{SP_1 \cup SP_2 \vdash^{\mathbb{I}}_{\Sigma} \varphi}\,[\text{sum}]$$

Theorem 4 (Soundness). *Let $\mathbb{I}$ be a logic and $SP \in \mathsf{Spec}^{\mathbb{I}}_{\Sigma}$. Then, if $SP \vdash^{\mathbb{I}}_{\Sigma} \varphi$, $SP \models^{\mathbb{I}}_{\Sigma} \varphi$.*

Proof. The proof follows analogous to that about the soundness of similar inference rules appearing (Sannella & Tarlecki, 1988, Sec. 6), by observing that each one of the rules in Def. 18 is sound with respect to the semantics presented in Def. 11.

Theorem 5 (Completeness). *Let $\mathbb{I}$ be a logic satisfying completeness of $\vdash^{\mathbb{I}}_{\Sigma}$, for all $\Sigma \in |\mathsf{Sign}^{\mathbb{I}}|$, weak-amalgamation, and $SP \in \mathsf{Spec}^{\mathbb{I}}_{\Sigma}$. If $SP \models^{\mathbb{I}}_{\Sigma} \varphi$, then $SP \vdash^{\mathbb{I}}_{\Sigma} \varphi$.*

Proof. By definition of $\models^{\mathbb{I}}_{\Sigma}$, if $SP \models^{\mathbb{I}}_{\Sigma} \varphi$, then $\mathbf{Mod}[SP] \models^{\mathbb{I}^{\Sigma}} \varphi$. Let $\mathbf{nf}(SP) =$ **derive from** $\langle \Sigma', \Gamma' \rangle$ **by** σ. Then, by Thm. 1, $\mathbf{nf}(SP) \equiv^{\mathbb{I}} SP$ and, consequently, $\mathbf{Mod}[\mathbf{derive\ from}\ \langle \Sigma', \Gamma' \rangle\ \mathbf{by}\ \sigma] \models^{\mathbb{I}^{\Sigma}} \varphi$.

$$\begin{aligned}
&\mathbf{Mod}[\mathbf{derive\ from}\ \langle \Sigma', \Gamma' \rangle\ \mathbf{by}\ \sigma] \models^{\mathbb{I}^{\Sigma}} \varphi \\
&\text{iff } \{\, \mathbf{Mod}(\sigma^{\mathsf{op}})(\mathcal{M}') \mid \mathcal{M}' \in \mathbf{Mod}[\langle \Sigma', \Gamma' \rangle] \,\} \models^{\mathbb{I}^{\Sigma}} \varphi \\
&\text{iff } \{\, \mathbf{Mod}(\sigma^{\mathsf{op}})(\mathcal{M}') \mid \mathcal{M}' \in |\mathbf{Mod}(\Sigma', \Gamma')| \,\} \models^{\mathbb{I}^{\Sigma}} \varphi \\
&\text{iff } \left\{ \mathbf{Mod}(\sigma^{\mathsf{op}})(\mathcal{M}') \;\middle|\; \mathcal{M}' \in |\mathbf{Mod}(\Sigma')| \text{ and } \mathcal{M}' \models^{\mathbb{I}^{\Sigma'}} \Gamma' \right\} \models^{\mathbb{I}^{\Sigma}} \varphi \\
&\text{iff for all } \mathcal{M}' \in |\mathbf{Mod}(\Sigma')|,\ \text{ if } \mathcal{M}' \models^{\mathbb{I}^{\Sigma'}} \Gamma' \text{ then } \mathbf{Mod}(\sigma^{\mathsf{op}})(\mathcal{M}') \models^{\mathbb{I}^{\Sigma}} \varphi \\
&\text{iff for all } \mathcal{M}' \in |\mathbf{Mod}(\Sigma')|,\ \text{ if } \mathcal{M}' \models^{\mathbb{I}^{\Sigma'}} \Gamma' \text{ then } \mathcal{M}' \models^{\mathbb{I}^{\Sigma'}} \mathbf{Sen}(\sigma)(\varphi) \\
&\text{iff } \Gamma' \models^{\mathbb{I}^{\Sigma'}} \mathbf{Sen}(\sigma)(\varphi) \\
&\text{iff } \Gamma' \vdash^{\mathbb{I}^{\Sigma'}} \mathbf{Sen}(\sigma)(\varphi)\ .
\end{aligned}$$

The proof is completed with the following derivation:

$$\cfrac{(\sigma : \Sigma \to \Sigma')\ \cfrac{\cfrac{\Gamma' \vdash^{\mathbb{I}^{\Sigma'}} \mathbf{Sen}(\sigma)(\varphi)}{\left\langle \Sigma', \Gamma' \right\rangle \vdash^{\mathbb{I}}_{\Sigma'} \mathbf{Sen}(\sigma)(\varphi)}\,[\text{basic}]}{\mathbf{derive\ from}\ \left\langle \Sigma', \Gamma' \right\rangle\ \mathbf{by}\ \sigma \vdash^{\mathbb{I}}_{\Sigma} \varphi}\,[\text{derive}] \qquad SP \equiv^{\mathbb{I}} \mathbf{nf}(SP)}{SP \vdash^{\mathbb{I}}_{\Sigma} \varphi}\,[\text{equiv}]$$

Observing the proof, specially regarding the use of Rule *[equiv]*, two questions immediately arise. In the first place Rule *[equiv]* can be regarded as a semantic rule because checking whether $SP \equiv^{\mathbb{I}} SP'$, by Def. 13, requires checking that both, SP and SP', share the same signature and class of models while, on the other hand, it can be completely axiomatized by:

1. the equations introduced in (Wirsing, 1991, Thm. 4.1, and Coro. 4.2) (as a sufficient set of equations for deriving the normal form of structured specifications),
2. rules characterising $\equiv^{\mathbb{I}}$ as a congruence:

$$\frac{}{SP \equiv^{\mathbb{I}} SP}\ [\text{refl.}] \quad \frac{SP \equiv^{\mathbb{I}} SP'}{SP' \equiv^{\mathbb{I}} SP}\ [\text{symm.}] \quad \frac{SP \equiv^{\mathbb{I}} SP' \qquad SP' \equiv^{\mathbb{I}} SP''}{SP \equiv^{\mathbb{I}} SP''}\ [\text{trans.}]$$

$$\frac{A \equiv^{\mathbb{I}} B}{SP[A] \equiv^{\mathbb{I}} SP[B]}\ [\text{repl.}]$$

   assuming SP is a structured specification with a placeholder for a structured specification, and $SP[A]$ (resp. $SP[B]$) denotes the replacement of such a placeholder for specification A (resp. B).
3. a rule for checking $\equiv^{\mathbb{I}}$ for specifications in normal form:

$$\frac{\left\{\mathbf{Sen}^{\mathbb{I}}(\sigma_1')(\Gamma_1)\vdash^{\mathbb{I}}{}_{\Sigma'}\psi\right\}_{\psi\in\mathbf{Sen}^{\mathbb{I}}(\sigma_2')(\Gamma_2)} \qquad \left\{\mathbf{Sen}^{\mathbb{I}}(\sigma_2')(\Gamma_2)\vdash^{\mathbb{I}}{}_{\Sigma'}\psi\right\}_{\psi\in\mathbf{Sen}^{\mathbb{I}}(\sigma_1')(\Gamma_1)}}{\textbf{derive from } \langle\Sigma_1,\Gamma_1\rangle \textbf{ by } \sigma_1 \equiv^{\mathbb{I}} \textbf{derive from } \langle\Sigma_2,\Gamma_2\rangle \textbf{ by } \sigma_2}\ [\text{basic equiv}]$$

   such that $\left\langle\sigma_1' : \Sigma_1 \to \Sigma', \sigma_2' : \Sigma_2 \to \Sigma'\right\rangle$ is the pushout for $\langle\sigma_1 : \Sigma \to \Sigma_1, \sigma_2 : \Sigma \to \Sigma_2\rangle$ in $\mathsf{Sign}^{\mathbb{I}}$.

The second question, and probably the one requiring a longer justification, is: does a calculus like this enjoy any usefulness? The answer to this question can be given from two perspectives. From a theoretical point of view, the completeness of this calculus reduces, almost trivially, to the completeness of the calculus of the underlying logic suggesting that, at the end, proofs are carried out in a non-compositional calculus for flat specifications in the underlying institutions. An equivalent result is shown in (Sannella & Tarlecki, 2012) where the authors consider a calculus with the following single rule:

$$\mathbf{nf}(SP) = \textbf{derive from } \left\langle\Sigma',\Gamma'\right\rangle \textbf{ by } \sigma \ \frac{\Gamma'\vdash^{\mathbb{I}\,\Sigma'}\mathbf{Sen}(\sigma)(\varphi)}{SP\vdash^{\mathbb{I}}{}_{\Sigma'}\varphi}$$

which, by (Sannella & Tarlecki, 2012, Thm. 9.2.16) is sound and complete, whenever $\mathbb{I}$ is *exact* (Sannella & Tarlecki, 2012, Def. 4.4.6).

Also from this point of view, the absence of use of Rules *[sum]* and *[translate]* of the calculus introduced in Def. 18 in the proof of Thm. 5, suggest that they can be derived from a calculus consisting only of Rules *[basic]*, *[derive]* and *[equiv]* of that same calculus. For the sake of the following results, we assume that the underlying calculus have two specific rules. On the one hand, we assume some form of weakening rule equivalent to: $\Gamma\vdash^{\mathbb{I}}{}_{\Sigma}\alpha$ implies $\Gamma \cup \Delta\vdash^{\mathbb{I}}{}_{\Sigma}\alpha$ and, on the other hand, a structural rule for combining sets of proofs relaying on the same set of hypothesis equivalent to: $\{\Gamma\vdash^{\mathbb{I}}{}_{\Sigma}\alpha\}_{\alpha\in\Delta_1}$ and $\{\Gamma\vdash^{\mathbb{I}}{}_{\Sigma}\alpha\}_{\alpha\in\Delta_2}$ implies $\{\Gamma\vdash^{\mathbb{I}}{}_{\Sigma}\alpha\}_{\alpha\in\Delta_1\cup\Delta_2}$. Even when we did not make these two conditions explicit before, the first assumption is justified on the fact that we strongly relayed on a definition of entailment system satisfying monotonicity thus, any complete calculus for it must have some sort of weakening rule; the second assumption is justified because the underlying calculus may involve infinitary rules and/or do not satisfy neither interpolation, nor weak interpolation and compactness, as it is witnessed by Table 1, so the application of Rules *[equiv]* and *[sum]* may require to prove infinitely many formulae.

Lemma 2. *Let* $\mathbb{I}$ *be a logic* $\sigma : \Sigma \to \Sigma' \in ||\mathsf{Sign}^{\mathbb{I}}||$, $SP \in \mathsf{Spec}^{\mathbb{I}}_{\Sigma}$ *such that* $\mathbf{nf}(SP) =$ **derive from** $\langle \Sigma', \Gamma' \rangle$ **by** σ *and* $\varphi \in |\mathbf{Sen}(\Sigma)|$. *Then,* $SP{\vdash^{\mathbb{I}}}_{\Sigma}\varphi$ *implies* $\Gamma'{\vdash^{\mathbb{I}}}_{\Sigma'}\mathbf{Sen}^{\mathbb{I}}(\sigma)(\varphi)$.

Proof. By Thm. 1 $\mathbf{nf}(SP) \equiv SP$ and consequently by Rule [*equiv*], $SP{\vdash^{\mathbb{I}}}_{\Sigma}\varphi$ in and only if **derive from** $\langle \Sigma', \Gamma' \rangle$ **by** $\sigma{\vdash^{\mathbb{I}}}_{\Sigma}\varphi$.

Now, lets assume that $\Gamma'{\not\vdash^{\mathbb{I}}}^{\Sigma'}\mathbf{Sen}^{\mathbb{I}}(\sigma)(\varphi)$ then, there exists $\mathcal{M}' \in |\mathbf{Mod}^{\mathbb{I}}(\Sigma')|$ such that $\mathcal{M}'{\models^{\mathbb{I}}}^{\Sigma'}\Gamma'$ and $\mathcal{M}'{\not\models^{\mathbb{I}}}^{\Sigma'}\mathbf{Sen}^{\mathbb{I}}(\sigma)(\varphi)$. If $\mathcal{M}' \in |\mathbf{Mod}^{\mathbb{I}}(\Sigma')|$ and $\mathcal{M}'{\models^{\mathbb{I}}}^{\Sigma'}\Gamma'$ then, by Def. 11, $\mathcal{M}' \in \mathbf{Mod}[\langle \Sigma', \Gamma' \rangle]$, and consequently $\mathbf{Mod}^{\mathbb{I}}(\sigma)(\mathcal{M}') \in \mathbf{Mod}[$**derive from** $\langle \Sigma', \Gamma' \rangle$ **by** $\sigma]$. By $\models$-invariance condition of Def. 3, $\mathcal{M}'{\models^{\mathbb{I}}}^{\Sigma'}\mathbf{Sen}^{\mathbb{I}}(\sigma)(\varphi)$ if and only if $\mathbf{Mod}^{\mathbb{I}}(\sigma)(\mathcal{M}'){\models^{\mathbb{I}}}^{\Sigma}\varphi$. Finally, if $\mathbf{Mod}^{\mathbb{I}}(\sigma)(\mathcal{M}') \in \mathbf{Mod}[$**derive from** $\langle \Sigma', \Gamma' \rangle$ **by** $\sigma]$ and $\mathbf{Mod}^{\mathbb{I}}(\sigma)(\mathcal{M}'){\not\models^{\mathbb{I}}}^{\Sigma}\varphi$ then, by Thm. 4, **derive from** $\langle \Sigma', \Gamma' \rangle$ **by** $\sigma{\not\vdash^{\mathbb{I}}}_{\Sigma}\varphi$.

Theorem 6. *In the calculus presented in Def. 18, Rules* [*sum*] *and* [*translate*] *can be derived by resorting only to Rules* [*basic*], [*derive*] *and* [*equiv*].

Proof. Let us first consider Rule [*sum*]. Rule [*sum*] allows us to conclude $SP_1 \cup SP_2 \vdash^{\mathbb{I}}_{\Sigma} \varphi$ provided that $\{SP_1 \vdash^{\mathbb{I}}_{\Sigma} \psi\}_{\psi \in \Delta}$ for some $\Delta \subseteq \mathbf{Sen}^{\mathbb{I}}(\Sigma)$, and $\langle \Sigma, \Delta \rangle \cup SP_2 \vdash^{\mathbb{I}}_{\Sigma} \varphi$, for $SP_1, SP_2 \in \mathsf{Spec}^{\mathbb{I}}_{\Sigma}$. Let us assume that $\mathbf{nf}(SP_i) =$ **derive from** $\langle \Sigma_i, \Gamma_i \rangle$ **by** σ_i with $\sigma_i : \Sigma \to \Sigma_i \in ||\mathsf{Sign}^{\mathbb{I}}||$ for $i \in \{1, 2\}$.

The proof is completed with the following derivations:

$$\dfrac{\dfrac{\vdots\,\pi}{\left\{\mathbf{Sen}^{\mathbb{I}}(\sigma_1')(\Gamma_1){\vdash^{\mathbb{I}}}^{\Sigma'}\psi\right\}_{\psi\in\mathbf{Sen}^{\mathbb{I}}(\sigma)(\Delta)}}\ \left[\text{by Lemma 2, and using } \left\{SP_1{\vdash^{\mathbb{I}}}^{\Sigma}\psi\right\}_{\psi\in\Delta}\right]}{[\mathbf{1}]\ \left\{\mathbf{Sen}^{\mathbb{I}}(\sigma_1')(\Gamma_1)\cup\mathbf{Sen}^{\mathbb{I}}(\sigma_2')(\Gamma_2){\vdash^{\mathbb{I}}}^{\Sigma'}\psi\right\}_{\psi\in\mathbf{Sen}^{\mathbb{I}}(\sigma)(\Delta)}}\ [\mathbf{W}]$$

$$\dfrac{\left\{\mathbf{Sen}^{\mathbb{I}}(\sigma_2')(\Gamma_2){\vdash^{\mathbb{I}}}^{\Sigma'}\psi\right\}_{\psi\in\mathbf{Sen}^{\mathbb{I}}(\sigma_2')(\Gamma_2)}}{[\mathbf{2}]\ \left\{\mathbf{Sen}^{\mathbb{I}}(\sigma_1')(\Gamma_1)\cup\mathbf{Sen}^{\mathbb{I}}(\sigma_2')(\Gamma_2){\vdash^{\mathbb{I}}}^{\Sigma'}\psi\right\}_{\psi\in\mathbf{Sen}^{\mathbb{I}}(\sigma_2')(\Gamma_2)}}\ [\mathbf{W}]$$

The reader should note that at points in the proof marked with [**W**] we use te assumption that the underlying calculus have some form of weakening rule.

$$\dfrac{[\mathbf{1}] \qquad [\mathbf{2}]}{[\mathbf{3}]\ \left\{\mathbf{Sen}^{\mathbb{I}}(\sigma_1')(\Gamma_1)\cup\mathbf{Sen}^{\mathbb{I}}(\sigma_2')(\Gamma_2){\vdash^{\mathbb{I}}}^{\Sigma'}\psi\right\}_{\psi\in\mathbf{Sen}^{\mathbb{I}}(\sigma)(\Delta)\cup\mathbf{Sen}^{\mathbb{I}}(\sigma_2')(\Gamma_2)}}\ [\mathbf{P}]$$

The point in the proof marked with [**P**] is where we resort to the assumption of existence of a structural rule for combining sets of proofs relaying on the same set of hypothesis.

$$\frac{\vdots\ \pi'}{[\mathbf{4}]\ \mathbf{Sen}^{\mathbb{I}}(\sigma)(\Delta) \cup \mathbf{Sen}^{\mathbb{I}}(\sigma_2')(\Gamma_2) \vdash^{\mathbb{I}\,\Sigma'} \mathbf{Sen}^{\mathbb{I}}(\sigma)(\varphi)}\ [\text{by Lemma 2, and using } \langle \Sigma, \Delta \rangle \cup SP_2 \vdash^{\mathbb{I}}{}_{\Sigma}\varphi]$$

$$\frac{[\mathbf{3}] \qquad [\mathbf{4}]}{\mathbf{Sen}^{\mathbb{I}}(\sigma_1')(\Gamma_1) \cup \mathbf{Sen}^{\mathbb{I}}(\sigma_2')(\Gamma_2) \vdash^{\mathbb{I}\,\Sigma'} \mathbf{Sen}^{\mathbb{I}}(\sigma)(\varphi)}\ [\text{CR}]$$

$$(\sigma : \Sigma \to \Sigma')\ \frac{\dfrac{\mathbf{Sen}^{\mathbb{I}}(\sigma_1')(\Gamma_1) \cup \mathbf{Sen}^{\mathbb{I}}(\sigma_2')(\Gamma_2) \vdash^{\mathbb{I}\,\Sigma'} \mathbf{Sen}^{\mathbb{I}}(\sigma)(\varphi)}{\left\langle \Sigma', \mathbf{Sen}^{\mathbb{I}}(\sigma_1')(\Gamma_1) \cup \mathbf{Sen}^{\mathbb{I}}(\sigma_2')(\Gamma_2) \right\rangle \vdash^{\mathbb{I}}{}_{\Sigma'} \mathbf{Sen}^{\mathbb{I}}(\sigma)(\varphi)}\ [\text{basic}]}{[\mathbf{5}]\ \mathbf{derive\ from}\ \left\langle \Sigma', \mathbf{Sen}^{\mathbb{I}}(\sigma_1')(\Gamma_1) \cup \mathbf{Sen}^{\mathbb{I}}(\sigma_2')(\Gamma_2) \right\rangle\ \mathbf{by}\ \sigma \vdash^{\mathbb{I}}{}_{\Sigma}\varphi}\ [\text{derive}]$$

such that $\sigma = \sigma_1 \circ \sigma_1' = \sigma_2 \circ \sigma_2'$, and $\left\langle \sigma_1' : \Sigma_1 \to \Sigma', \sigma_2' : \Sigma_2 \to \Sigma' \right\rangle$ is the pushout for $\langle \sigma_1 : \Sigma \to \Sigma_1, \sigma_2 : \Sigma \to \Sigma_2 \rangle$ in $\mathsf{Sign}^{\mathbb{I}}$.

$$\frac{[\mathbf{5}] \qquad SP_1 \cup SP_2 \equiv^{\mathbb{I}} \mathbf{nf}(SP_1 \cup SP_2)}{SP_1 \cup SP_2 \vdash^{\mathbb{I}}{}_{\Sigma}\varphi}\ [\text{equiv}]$$

Next we prove that Rule [*translate*] can be derived resorting to Rules [*basic*], [*derive*] and [*equiv*]. Rule [*translate*] allows us to conclude that, if $SP_1 \vdash^{\mathbb{I}\,\Sigma} \varphi$ where $SP_1 \in \mathsf{Spec}^{\mathbb{I}}_{\Sigma}$, $\sigma : \Sigma \to \widehat{\Sigma}$ and $\varphi \in \mathbf{Sen}^{\mathbb{I}}(\Sigma)$, then $SP \vdash^{\mathbb{I}}{}_{\Sigma} \mathbf{Sen}(\sigma)(\varphi)$ where $SP = \mathbf{translate}\ SP_1\ \mathbf{by}\ \sigma$. Let us assume that $\mathbf{nf}(\mathit{SP_1})$ is the specification **derive from** $\langle \Sigma_1, \Gamma_1 \rangle$ **by** σ_1 with $\sigma_1 : \Sigma \to \Sigma_1 \in ||\mathsf{Sign}^{\mathbb{I}}||$.

$$\frac{\vdots\ \pi}{\Gamma_1 \vdash^{\mathbb{I}\,\Sigma_1} \mathbf{Sen}^{\mathbb{I}}(\sigma_1)(\varphi)}\ [\text{by Lemma 2, and using } SP_1 \vdash^{\mathbb{I}}{}_{\Sigma_1}\varphi]$$

$$\frac{\vdots\ \pi'}{[\mathbf{4}]\ \mathbf{Sen}^{\mathbb{I}}(\sigma_1')(\Gamma_1) \vdash^{\mathbb{I}\,\Sigma'} \mathbf{Sen}^{\mathbb{I}}(\sigma_1')(\mathbf{Sen}^{\mathbb{I}}(\sigma_1)(\varphi))}\ [\text{by } \vdash\text{-translation and } \Gamma_1 \vdash^{\mathbb{I}\,\Sigma_1} \mathbf{Sen}^{\mathbb{I}}(\sigma_1)(\varphi)]$$

$$(\sigma : \Sigma \to \mathbf{Sig}[SP])\ \frac{\dfrac{\dfrac{[\mathbf{4}]}{\mathbf{Sen}^{\mathbb{I}}(\sigma_1')(\Gamma_1) \vdash^{\mathbb{I}\,\Sigma'} \mathbf{Sen}^{\mathbb{I}}(\sigma')(\mathbf{Sen}^{\mathbb{I}}(\sigma)(\varphi))}\ [\text{by pushouts prop.}]}{\left\langle \Sigma', \mathbf{Sen}^{\mathbb{I}}(\sigma_1')(\Gamma_1) \right\rangle \vdash^{\mathbb{I}}{}_{\Sigma'} \mathbf{Sen}^{\mathbb{I}}(\sigma')(\mathbf{Sen}^{\mathbb{I}}(\sigma)(\varphi))}\ [\text{basic}]}{[\mathbf{5}]\ \mathbf{derive\ from}\ \left\langle \Sigma', \mathbf{Sen}^{\mathbb{I}}(\sigma_1')(\Gamma_1) \right\rangle\ \mathbf{by}\ \sigma' \vdash^{\mathbb{I}}{}_{\Sigma'} \mathbf{Sen}^{\mathbb{I}}(\sigma)(\varphi)}\ [\text{derive}]$$

$$\frac{[\mathbf{5}] \qquad \mathbf{translate}\ SP\ \mathbf{by}\ \sigma \equiv^{\mathbb{I}} \mathbf{nf}(\mathbf{translate}\ SP\ \mathbf{by}\ \sigma)}{\mathbf{translate}\ SP_1\ \mathbf{by}\ \sigma \vdash^{\mathbb{I}}{}_{\Sigma} \mathbf{Sen}(\sigma)(\varphi)}\ [\text{equiv}]$$

such that $\left\langle \sigma' : \mathbf{Sig}[SP] \to \Sigma', \sigma_1' : \Sigma_1 \to \Sigma' \right\rangle$ is the pushout for $\langle \sigma : \Sigma \to \mathbf{Sig}[SP], \sigma_1 : \Sigma \to \Sigma_1 \rangle$ in $\mathsf{Sign}^{\mathbb{I}}$.

From a practical perspective, we do not advocate for reducing structured specifications to flat ones as would be understood from the proof of Thm. 5; the structure inside specifications should be a valuable aid during the process of building proofs for a long as possible. Thus moving to a flat specification should be a resource reserved to be used when no better alternatives are at hand. In Ex. 1 we show an application of the calculus introduced in Def. 18

in a context where Borzyszkowski's calculus is not complete. Afterwards we will discuss the differences with Borzyszkowski's calculus, emphasising the reasons why the proofs in the example are possible in our calculus.

Proposition 1 shows some useful properties that can be used together with rule *[equiv]* in order to avoid flattening the specification preserving most of its structural properties. Propositions 1.4 to 1.6 were taken from (Sannella & Tarlecki, 2012).

Proposition 1. [Properties of SBOs] *Let $\mathbb{I}$ be a logic satisfying weak-interpolation and weak-amalgamation. Let $\Sigma, \Sigma' \in |\mathsf{Sign}^{\mathbb{I}}|$, $\sigma : \Sigma \to \Sigma' \in ||\mathsf{Sign}^{\mathbb{I}}||$, $\Gamma \subseteq \mathbf{Sen}^{\mathbb{I}}(\Sigma)$, $SP, SP_1, SP_2, SP_3 \in \mathsf{Spec}^{\mathbb{I}}_{\Sigma}$, $SP' \in \mathsf{Spec}^{\mathbb{I}}_{\Sigma'}$. Then, the following properties hold:*

1. $\langle \Sigma', \mathbf{Sen}(\sigma)(\Gamma)\rangle \cup \textbf{translate } SP \textbf{ by } \sigma \equiv \textbf{translate } \langle \Sigma, \Gamma\rangle \cup SP \textbf{ by } \sigma$,
2. $\textbf{derive from } \langle \Sigma', \mathbf{Sen}(\sigma)(\Gamma)\rangle \cup SP' \textbf{by } \sigma \equiv \langle \Sigma, \Gamma\rangle \cup \textbf{derive from } SP' \textbf{by } \sigma$,
3. $\langle \Sigma, \Gamma\rangle \cup (SP_1 \cup SP_2) \equiv (\langle \Sigma, \Gamma\rangle \cup SP_1) \cup SP_2$,
4. $SP_1 \cup SP_2 \equiv SP_2 \cup SP_1$,
5. $(SP_1 \cup SP_2) \cup SP_3 \equiv SP_1 \cup (SP_2 \cup SP_3)$,
6. $\langle \Sigma, \Gamma_1\rangle \cup \langle \Sigma, \Gamma_2\rangle \equiv \langle \Sigma, \Gamma_1 \cup \Gamma_2\rangle$.

Proof. Properties 1 and 2 follow by Def. 11 and set-theoretical reasoning on the classes of models. Property 3 is an instance of 5. The proofs of 4, 5 and 6 can be found in (Sannella & Tarlecki, 2012, Prop. 5.6.2).

Borzyszkowski's completeness proof (Borzyszkowski, 2002) suggests that proofs of properties of a union of specifications should be organized by resorting to Rules *[CR]*, *[sum1]* and *[sum2]*. This is possible because of the (implicit) use of the Craig's interpolation property in the elimination of the union of two specifications. In this sense, interpolation, compactness, infinite conjunction or finiteness are (strong) requirements in the completeness of Borzyszkowski's calculus. In the case of logics that do not meet any of these conditions (as the ones shown in Table 1), that construction is not possible because either the interpolant does not exists or it is not a formula, but rather a (possibly infinite) set of formulae, when the underlying logic only satisfies the weak interpolation property.

Rule *[sum]* exhibits an interesting use of the weak interpolation property as a means for decomposing structured specifications resulting from the application of the union operator. The rule makes explicit the construction used by Borzyszkowski in the proof of completeness of his calculus; in effect, Rule *[sum]* eliminates a union between two structured specifications, but at the expense of introducing another one between a structured specification and a flat one. This responds to the need of keeping the (possibly infinite) interpolant to complete the proof. This is done, of course, at the cost of moving to a calculus that must support an infinitary structural rule allowing to draw conclusions from possibly infinite sets of proofs.

Next, we show the use of the calculus of Def. 18 to prove a property from a structured specification over the language of *Propositional Dynamic Logic* (Harel et al., 2000).

Example 1 (Reasoning in Propositional Dynamic Logic *– PDL (Harel et al., 2000)).* Let $\Sigma = \{R\} \in |\mathsf{Sign}|$. Let $SP_1 = \langle \Sigma, \Gamma_1 \rangle$, $SP_2' = \langle \Sigma, \Gamma_2' \rangle$ and $SP_2'' = \langle \Sigma, \Gamma_2'' \rangle$ be PDL flat specifications such that:

- $\{\Gamma_1 \vdash^\Sigma \gamma\}_{\gamma \in \Gamma_1'}$,
- $\Gamma_1' = \{\beta \to [Skip]\alpha\} \cup \big\{(\beta \to [R^{i-1}]\alpha) \to (\beta \to [R^i]\alpha)\big\}_{0<i<\omega}$,
- $\Gamma_2' \vdash^\Sigma \beta$, and
- $\Gamma_2'' \vdash^\Sigma [R^*](\alpha \to \gamma)$.

Then, $SP_1 \cup (SP_2' \cup SP_2'') \vdash_\Sigma [R^*]\gamma$.

The following tree depicts the dependencies among proofs, being π the main proof.

$$\dfrac{\dfrac{\dfrac{\dfrac{\pi_{1.1.1.1(i)}}{\pi_{1.1.1(i)}}}{\pi_{1.1}}}{\pi_1} \qquad \dfrac{\dfrac{\dfrac{\pi_{2.1.1.1}}{\pi_{2.1.1}} \quad \dfrac{\pi_{2.1.1.1}}{\pi_{2.1.2}}}{\pi_{2.1}} \qquad \dfrac{\dfrac{\pi_{2.2.2.1} \qquad \pi_{2.2.2.2}}{\pi_{2.2.2}}}{\pi_{2.2}}}{\pi_2}}{\pi}$$

We will present the proof in a top-down fashion, starting from proof π.
π:

$$\dfrac{\dfrac{\pi_1}{\left\{SP_1 \vdash_\Sigma \beta \to [R^i]\alpha\right\}_{i<\omega}} \qquad \dfrac{\pi_2}{\left\langle \Sigma, \left\{\beta \to [R^i]\alpha\right\}_{i<\omega}\right\rangle \cup (SP_2' \cup SP_2'') \vdash_\Sigma [R^*]\gamma}}{SP_1 \cup (SP_2' \cup SP_2'') \vdash_\Sigma [R^*]\gamma}\,[\text{sum}]$$

π_1:

$$\left\{\dfrac{\dfrac{\pi_{1.1}}{SP_1 \cup \langle\Sigma,\emptyset\rangle \vdash_\Sigma \beta \to [R^i]\alpha} \qquad \dfrac{\dfrac{\dfrac{\langle\Sigma,\Gamma_1\rangle \equiv \langle\Sigma,\Gamma_1\rangle}{\langle\Sigma,\Gamma_1 \cup \emptyset\rangle \equiv \langle\Sigma,\Gamma_1\rangle}\,[\text{by set theory}]}{\langle\Sigma,\Gamma_1\rangle \cup \langle\Sigma,\emptyset\rangle \equiv \langle\Sigma,\Gamma_1\rangle}\,[\text{by Prop. 1.(6)}]}{SP_1 \cup \langle\Sigma,\emptyset\rangle \equiv SP_1}\,[\text{by Def. } SP_1]}{SP_1 \vdash_\Sigma \beta \to [R^i]\alpha}\,[\text{equiv}]\right\}_{i<\omega}$$

$\pi_{1.1}$:

$$\left\{\dfrac{\dfrac{\{\Gamma_1 \vdash^\Sigma \gamma\}_{\gamma\in\Gamma_1'}}{\{SP_1 \vdash_\Sigma \gamma\}_{\gamma\in\Gamma_1'}}\,[\text{basic}] \qquad \pi_{1.1.1(i)}}{SP_1 \cup \langle\Sigma,\emptyset\rangle \vdash_\Sigma \beta \to [R^i]\alpha}\,[\text{sum}]\right\}_{i<\omega}$$

$\pi_{1.1.1(i)}$:

$$\dfrac{\dfrac{\dfrac{\pi_{1.1.1.1(i)}}{\Gamma_1' \vdash^\Sigma \beta \to [R^i]\alpha}\,[\text{if } 0<i]}{\left\langle\Sigma,\Gamma_1'\right\rangle \vdash_\Sigma \beta \to [R^i]\alpha}\,[\text{basic}] \qquad \dfrac{\dfrac{\left\langle\Sigma,\Gamma_1'\right\rangle \equiv \left\langle\Sigma,\Gamma_1'\right\rangle}{\left\langle\Sigma,\Gamma_1' \cup \emptyset\right\rangle \equiv \left\langle\Sigma,\Gamma_1'\right\rangle}\,[\text{set theory}]}{\left\langle\Sigma,\Gamma_1'\right\rangle \cup \langle\Sigma,\emptyset\rangle \equiv \left\langle\Sigma,\Gamma_1'\right\rangle}\,[\text{by Prop. 1.(6)}]}{\left\langle\Sigma,\Gamma_1'\right\rangle \cup \langle\Sigma,\emptyset\rangle \vdash_\Sigma \beta \to [R^i]\alpha}\,[\text{equiv}]$$

$\pi_{1.1.1.1(i)}$:

From hypothesis $\beta \to [Skip]\alpha$ and $\left\{\beta \to [R^{i-1}]\alpha) \to (\beta \to [R^i]\alpha)\right\}_{0<i<\omega}$ in Γ'_1 it is possible to prove $\Gamma'_1 \vdash^{\Sigma} \beta \to [R^i]\alpha$, for every $0 < i < \omega$.

π_2:

$$\dfrac{\dfrac{\dfrac{\pi_{2.1}}{\left\{\left\langle \Sigma, \left\{\beta \to [R^i]\alpha\right\}_{i<\omega}\right\rangle \cup SP'_2 \vdash_{\Sigma} [R^j]\alpha\right\}_{j<\omega}} \qquad \dfrac{\pi_{2.2}}{\left\langle \Sigma, \left\{[R^i]\alpha\right\}_{i<\omega}\right\rangle \cup SP''_2 \vdash_{\Sigma} [R^*]\gamma}}{(\left\langle \Sigma, \left\{\beta \to [R^i]\alpha\right\}_{i<\omega}\right\rangle \cup SP'_2) \cup SP''_2 \vdash_{\Sigma} [R^*]\gamma}\,[\text{sum}]}{\left\langle \Sigma, \left\{\beta \to [R^i]\alpha\right\}_{i<\omega}\right\rangle \cup (SP'_2 \cup SP''_2) \vdash_{\Sigma} [R^*]\gamma}\,[\text{by Prop. 1.(5)}]$$

$\pi_{2.1}$:

$$\left\{\ \dfrac{\dfrac{\pi_{2.1.1}}{\left\langle \Sigma, \left\{\beta \to [R^i]\alpha\right\}_{i<\omega}\right\rangle \cup SP'_2 \vdash_{\Sigma} \beta} \qquad \dfrac{\pi_{2.1.2}}{\left\langle \Sigma, \left\{\beta \to [R^i]\alpha\right\}_{i<\omega}\right\rangle \cup SP'_2 \vdash_{\Sigma} \beta \to [R^j]\alpha}}{\left\langle \Sigma, \left\{\beta \to [R^i]\alpha\right\}_{i<\omega}\right\rangle \cup SP'_2 \vdash_{\Sigma} [R^j]\alpha}\,[\text{MP}]\ \right\}_{j<\omega}$$

$\pi_{2.1.1}$:

$$\dfrac{\dfrac{\dfrac{\Gamma'_2 \vdash^{\Sigma} \beta}{\left\{\beta \to [R^i]\alpha\right\}_{i<\omega} \cup \Gamma'_2 \vdash^{\Sigma} \beta}\,[\text{mon. of } \vdash^{\Sigma}]}{\left\langle \Sigma, \left\{\beta \to [R^i]\alpha\right\}_{i<\omega} \cup \Gamma'_2\right\rangle \vdash_{\Sigma} \beta}\,[\text{basic}] \qquad \pi_{2.1.1.1}}{\left\langle \Sigma, \left\{\beta \to [R^i]\alpha\right\}_{i<\omega}\right\rangle \cup SP'_2 \vdash_{\Sigma} \beta}\,[\text{equiv}]$$

$\pi_{2.1.1.1}$:

$$\dfrac{\dfrac{\left\langle \Sigma, \left\{\beta \to [R^i]\alpha\right\}_{i<\omega} \cup \Gamma'_2\right\rangle \equiv \left\langle \Sigma, \left\{\beta \to [R^i]\alpha\right\}_{i<\omega} \cup \Gamma'_2\right\rangle}{\left\langle \Sigma, \left\{\beta \to [R^i]\alpha\right\}_{i<\omega}\right\rangle \cup \left\langle \Sigma, \Gamma'_2\right\rangle \equiv \left\langle \Sigma, \left\{\beta \to [R^i]\alpha\right\}_{i<\omega} \cup \Gamma'_2\right\rangle}\,[\text{by Prop. 1.(6)}]}{\left\langle \Sigma, \left\{\beta \to [R^i]\alpha\right\}_{i<\omega}\right\rangle \cup SP'_2 \equiv \left\langle \Sigma, \left\{\beta \to [R^i]\alpha\right\}_{i<\omega} \cup \Gamma'_2\right\rangle}\,[\text{by Def. } SP'_2]$$

$\pi_{2.1.2}$:

$$\dfrac{\dfrac{\dfrac{\left\{\beta \to [R^i]\alpha\right\}_{i<\omega} \vdash^{\Sigma} \beta \to [R^j]\alpha}{\left\{\beta \to [R^i]\alpha\right\}_{i<\omega} \cup \Gamma'_2 \vdash^{\Sigma} \beta \to [R^j]\alpha}\,[\text{mon. of } \vdash^{\Sigma}]}{\left\langle \Sigma, \left\{\beta \to [R^i]\alpha\right\}_{i<\omega} \cup \Gamma'_2\right\rangle \vdash_{\Sigma} \beta \to [R^j]\alpha}\,[\text{basic}] \qquad \pi_{2.1.1.1}}{\left\langle \Sigma, \left\{\beta \to [R^i]\alpha\right\}_{i<\omega}\right\rangle \cup SP'_2 \vdash_{\Sigma} \beta \to [R^j]\alpha}\,[\text{equiv}]$$

$\pi_{2.2}$:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\left\{[R^i]\alpha\right\}_{i<\omega} \vdash^{\Sigma} [R^*]\alpha}\,[\text{Inf*}]}{\left\{[R^i]\alpha\right\}_{i<\omega} \vdash^{\Sigma} [R^*]\alpha}\,[\text{mon. of } \vdash^{\Sigma}]}{\left\langle \Sigma, \left\{[R^i]\alpha\right\}_{i<\omega} \right\rangle \vdash_{\Sigma} [R^*]\alpha}\,[\text{basic}] \qquad \pi_{2.2.2}}{\left\langle \Sigma, \left\{[R^i]\alpha\right\}_{i<\omega} \right\rangle \cup SP''_2 \vdash_{\Sigma} [R^*]\gamma}\,[\text{sum}]$$

$\pi_{2.2.2}$:

$$\dfrac{\dfrac{\dfrac{\pi_{2.2.2.1}}{[R^*]\alpha \cup \Gamma''_2 \vdash^{\Sigma} [R^*]\gamma}}{\left\langle \Sigma, [R^*]\alpha \cup \Gamma''_2 \right\rangle \vdash_{\Sigma} [R^*]\gamma}\,[\text{basic}] \qquad \dfrac{\pi_{2.2.2.2}}{\langle \Sigma, [R^*]\alpha \rangle \cup SP''_2 \equiv \left\langle \Sigma, [R^*]\alpha \cup \Gamma''_2 \right\rangle}}{\langle \Sigma, [R^*]\alpha \rangle \cup SP''_2 \vdash_{\Sigma} [R^*]\gamma}\,[\text{equiv}]$$

$\pi_{2.2.2.1}$:

$$\dfrac{\dfrac{[R^*]\alpha \vdash^{\Sigma} [R^*]\alpha}{[R^*]\alpha \cup \Gamma''_2 \vdash^{\Sigma} [R^*]\alpha}\,[\text{mon. of } \vdash^{\Sigma}] \qquad \dfrac{\dfrac{\Gamma''_2 \vdash^{\Sigma} [R^*](\alpha \to \gamma)}{\Gamma''_2 \vdash^{\Sigma} [R^*]\alpha \to [R^*]\gamma}\,[\text{Distr}]}{[R^*]\alpha \cup \Gamma''_2 \vdash^{\Sigma} [R^*]\alpha \to [R^*]\gamma}\,[\text{mon. of } \vdash^{\Sigma}]}{[R^*]\alpha \cup \Gamma''_2 \vdash^{\Sigma} [R^*]\gamma}\,[\text{modus ponens}]$$

$\pi_{2.2.2.2}$:

$$\dfrac{\dfrac{\left\langle \Sigma, [R^*]\alpha \cup \Gamma''_2 \right\rangle \equiv \left\langle \Sigma, [R^*]\alpha \cup \Gamma''_2 \right\rangle}{\langle \Sigma, [R^*]\alpha \rangle \cup \left\langle \Sigma, \Gamma''_2 \right\rangle \equiv \left\langle \Sigma, [R^*]\alpha \cup \Gamma''_2 \right\rangle}\,[\text{by Prop. 1.(6)}]}{\langle \Sigma, [R^*]\alpha \rangle \cup SP''_2 \equiv \left\langle \Sigma, [R^*]\alpha \cup \Gamma''_2 \right\rangle}\,[\text{by Def. } SP''_2]$$

Looking at the uses of Rules *[sum]* and *[equiv]* in Example 1, we can conclude:

1. Rule *[sum]* allows us to share required hypotheses between subproofs of a sum. This is particularly clear in proofs π and π_5.
2. In some occasions, Rule *[sum]* forces us to introduce a flat specification that may not be necessary (i.e., proofs π_1 and π_3). Rule *[equiv]* allows us to get rid of the spurious specifications.
3. Rule *[equiv]* has to be used carefully. If abused, it allows one to completely ignore the specification structure. We advocate for a responsible use of this rule. Notice that the remaining uses of Rule *[equiv]* in proofs π_5, π_7 and π_8 are meant to unite two flat specifications into a single flat specification in order to prove a property in the complete calculus of the underlying logic. In essence, our rationale for using this rule is not to remove structure, but rather change the structure in a way that might help during the proving process.

4 Heterogeneity and Structured Specifications

As we mentioned before, heterogeneous specifications are ubiquitous in modern software development. Different software artifacts are developed with the aid of different modeling languages. Logics fall within the “formal” subclass of these modeling languages. A variety of logics for software modeling exist. Classical first-order logic is generally used in the specification of structural properties of software. Dynamic logics are used for method specification through requires/ensures clauses. A number of temporal logics are used in software analysis in the context of model checking, etc. Therefore, calculi such as the ones presented in Defs. 16 and 18, while relevant in an homogeneous setting, do not constitute a solution for incorporating the diversity of formalisms used nowadays.

4.1 Foundations and Limitations

The need to reason about heterogeneous structured specifications is not novel. Among the many proposals we can cite (Mossakowski & Tarlecki, 2014), where Mossakowski and Tarlecki extend Borzyszkowski’s results and definitions to the heterogeneous case, or (Cerioli & Meseguer, 1997), where Cerioli and Meseguer connect different proof calculi (Meseguer, 1989) via their entailment systems.

In (Mossakowski & Tarlecki, 2009), the authors distinguish three classes of heterogeneous environments for software specification, namely: *a*) heterogeneous specification in some logical environments representable in a “universal” institution, *b*) focused heterogeneous specifications, and *c*) distributed heterogeneous specifications. The first two approaches do not differ in practice from each other. The main difference is philosophical, and relates to how the languages intervening in the framework are conceived. The “universal” institution approach considers the existence of a multiplicity of languages, all of them related to a particular logic that serves as common language in which the interaction of the bits and pieces of a specification can be expressed. Instead, the focused approach conceives the language as being itself heterogeneous.

In this section we will focus on structured specifications over an heterogeneous logical environments where heterogeneity is resolved by the translation to a “universal” institution. Thus, this corresponds to the first approach presented in (Mossakowski & Tarlecki, 2009). Semantics-preserving translations have been used for many reasons, even before the concept of institution was introduced in (Goguen & Burstall, 1984) and developed in (Goguen & Burstall, 1992), and, different kinds of arrows between institutions were studied in (Astesiano & Cerioli, 1991; Cerioli, 1993; Goguen & Roşu, 2002; Martini & Wolter, 1997, 1998). A relevant application is to provide a formal semantics for diagrammatic notations, in particular for the UML (see (Broy & Cengarle, 2011) and the citations therein). When such translations are used between two logics in a more classical logic setting, they are usually called interpretability results. One important application consists in mapping a logic to another in order to borrow the deduction system the latter provides. Quoting (Orlowska & Golińska-Pilarek, 2011), one way to pursue this goal, in the context of tableaux calculi, is the following:

- "*First, given a theory, a truth preserving translation is defined of the language of the theory into an appropriate language of relations (most often binary);*
- *Second, a dual tableau is constructed for this relational language so that it provides a deduction system for the original theory.*"

Generalizing (Orlowska & Golińska-Pilarek, 2011), an interpretation of a logic L into another logic L' consists of a mapping $T_L : Sen_L \to Sen_{L'}$ (translating sentences from logic L to sentences of logic L'), satisfying the following interpretability condition: For all $\alpha \in Sen_L$ and $\Gamma \subseteq Sen_L$, $\Gamma \models_L \alpha$ if and only if $\Gamma' \cup \{ T_L(\gamma) \mid \gamma \in \Gamma \} \models_{L'} T_L(\alpha)$, where Γ' is a set of axioms, only depending on the signature of the source theory presentation, used to establish basic restrictions on the class of models.

Proving this interpretability condition usually requires more that just a translation between sentences. This translation must be proved to be semantics-preserving. Thus, also a relation between classes of models must be established. In general, if L is a logic and Σ_L is any L-signature, an interpretability result of L in L' is presented by providing:

1. a mapping S from Σ_L to a particular signature Σ in L',
2. a translation $T_{L \to L'}$ of Σ_L-formulas to Σ-formulas,
3. a mapping $M_{L' \to L}$ of Σ-models (i.e., $Mod_{L'}(\Sigma)$) to Σ_L-models (i.e., $Mod_L(\Sigma_L)$), satisfying that for all $\mathfrak{B} \in Mod_{L'}(\Sigma)(M_{L' \to L}(\mathfrak{B})) \models_L \alpha$ if and only if $\mathfrak{B} \models_{L'} T_{L \to L'}(\alpha)$, and
4. a mapping $M_{L \to L'}$ of Σ_L-models to Σ-models, satisfying that for all $\mathfrak{A} \in Mod_L(\Sigma_L)(\mathfrak{A}) \models_L \alpha$ if and only if $M_{L \to L'}(\mathfrak{A}) \models_{L'} T_{L \to L'}(\alpha)$.

Interpretability results and institution co-morphisms are closely related. Institutions provide a more general setting, though. This is made evident in Cond. 1, where a mapping between signatures is required for interpretation, while Def. 19 requires a functor between the categories of signatures. The same occurs in Cond. 2 with respect to the natural family of functions required by Def. 19 to map sentences, and Cond. 3 with respect to the natural transformation required to map categories of models. Condition 4 is equivalent to the ρ-expansion of models along the institution co-morphism $\rho : \mathbb{I}_L \to \mathbb{I}_{L'}$ (see Def. 20), modulo the need to take into account the morphisms. Finally, under these four conditions, proving interpretability becomes an instance of the more general proof required in Thm. 7.

The following definition was taken from (Tarlecki, 1996), where it was called *institution representation*, but we will adopt the more modern, and widely used, name of co-morphism.

Definition 19 (Institution co-morphism (Goguen & Roşu, 2002)). Let $\mathbb{I}$ and $\mathbb{I}'$ be institutions. Then, $\langle \gamma^{Sign}, \gamma^{Sen}, \gamma^{Mod} \rangle : I \to I'$ is an *institution co-morphism* if

- $\gamma^{Sign} : \mathsf{Sign} \to \mathsf{Sign}'$ is a functor,
- $\gamma^{Sen} : \mathbf{Sen} \to \gamma^{Sign} \circ \mathbf{Sen}'$ is a natural transformation (i.e., a natural family of functions $\gamma^{Sen}_{\Sigma} : \mathbf{Sen}(\Sigma) \to \mathbf{Sen}'(\gamma^{Sign}(\Sigma))$),
- $\gamma^{Mod} : (\gamma^{Sign})^{\mathsf{op}} \circ \mathbf{Mod}' \to \mathbf{Mod}$ is a natural transformation (i.e., the family of functors $\gamma^{Mod}_{\Sigma} : \mathbf{Mod}'((\gamma^{Sign})^{\mathsf{op}}(\Sigma)) \to \mathbf{Mod}(\Sigma)$ is natural),

Such that, for any $\Sigma \in |\mathsf{Sign}|$, the following *satisfaction condition* holds: for any $\alpha \in \mathbf{Sen}(\Sigma)$ and $\mathcal{M}' \in |\mathbf{Mod}'((\gamma^{Sign})^{\mathsf{op}}(\Sigma))$,

$$\mathcal{M}' \models^{\gamma^{Sign}(\Sigma)} \gamma^{Sen}_{\Sigma}(\alpha) \quad \text{if and only if} \quad \gamma^{Mod}_{\Sigma}(\mathcal{M}') \models^{\Sigma} \alpha \ .$$

The reader should note that the direction of the arrows show how the whole of $\mathbb{I}$ is represented by some parts of $\mathbb{I}'$. The following two results, presented in (Tarlecki, 1996), provide the relationship between $\mathbb{I}$ and $\mathbb{I}'$.

Conditions 1 to 4 presented before, once they are extended to form an institution co-morphism, guarantee the satisfaction of the hypotheses of Thm. 8, thus providing the necessary hypotheses to extend the calculus of Def. 18 with Rule (1) below. In (Lopez Pombo, 2007) we developed this extension in detail for the interpretability of first-order linear temporal logic into ω-closure fork algebras (Frias, Baum, & Maibaum, 2002).

Next we reproduce some interesting results about institution co-morphisms from (Tarlecki, 1996).

Proposition 2 ((Tarlecki, 1996)). *Let $\mathbb{I}$ and $\mathbb{I}'$ be institutions. Let $\rho : \mathbb{I} \to \mathbb{I}'$ be an institution co-morphism. For all $\Sigma \in |\mathsf{Sign}|$, $\Gamma \subseteq \mathbf{Sen}(\Sigma)$ and $\varphi \in \mathbf{Sen}(\Sigma)$, if $\Gamma \models^{\Sigma} \varphi$, then $\rho^{Sen}(\Gamma) \models^{\rho^{Sign}(\Sigma)} \rho^{Sen}(\varphi)$.*

Definition 20 ((Tarlecki, 1996)). Let $\mathbb{I}$ and $\mathbb{I}'$ be institutions. Let $\rho : \mathbb{I} \to \mathbb{I}'$ be an institution co-morphism. $\mathcal{M} \in |\mathbf{Mod}(\Sigma)|$ has a *ρ-expansion* if there exists $\mathcal{M}' \in |\mathbf{Mod}'(\rho^{Sign}(\Sigma))|$ such that $\mathcal{M} = \rho^{Mod}(\mathcal{M}')$.

Theorem 7 ((Tarlecki, 1996)). *Let $\mathbb{I}$ and $\mathbb{I}'$ be institutions. Let $\rho : \mathbb{I} \to \mathbb{I}'$ be an institution co-morphism. For all $\Sigma \in |\mathsf{Sign}|$, $\Gamma \subseteq \mathbf{Sen}(\Sigma)$ and $\varphi \in \mathbf{Sen}(\Sigma)$, if every $\mathcal{M} \in \mathbf{Mod}[\langle \Sigma, \Gamma \rangle]$ has a ρ-expansion, then $\Gamma \models^{\Sigma} \varphi$ if and only if $\rho^{Sen}(\Gamma) \models^{\rho^{Sign}(\Sigma)} \rho^{Sen}(\varphi)$.*

In (Borzyszkowski, 1998) and (Borzyszkowski, 2002, Sec. 5) Borzyszkowski provides a good insight on how these relations between institutions affect the use of structured specifications and, in particular, the conditions under which one can move structured specifications between logical systems. Let us review Borzyszkowski's definitions and results.

Definition 21 (Weak amalgamation of institution co-morphisms (Borzyszkowski, 2002)). Let $\mathbb{I}$ and $\mathbb{I}'$ be institutions. Let $\rho : \mathbb{I} \to \mathbb{I}'$ be an institution co-morphism. We say that ρ has the *weak amalgamation property* if for all $\Sigma_1, \Sigma_2 \in |\mathsf{Sign}|$, $\sigma : \Sigma_1 \to \Sigma_2$, $\mathcal{M}_1 \in |\mathbf{Mod}'((\rho^{Sign})^{\mathsf{op}}(\Sigma_1))|$ and $\mathcal{M}_2 \in |\mathbf{Mod}(\Sigma_2)|$, if $\mathbf{Mod}(\sigma^{\mathsf{op}})(\mathcal{M}_2) = \rho^{Mod}_{\Sigma_1}(\mathcal{M}_1)$, there exists $\mathcal{M} \in$

$|\mathbf{Mod}'((\rho^{Sign})^{\mathsf{op}}(\Sigma_2))|$ such that $\rho^{Mod}_{\Sigma_2}(\mathcal{M}) = \mathcal{M}_2$ and $\mathbf{Mod}'((\rho^{Sign})^{\mathsf{op}}(\sigma^{\mathsf{op}}))(\mathcal{M}) = \mathcal{M}_1$.

The weak amalgamation property essentially establishes that, given an institution co-morphism ρ, the class of models of the translation through ρ of a signature of the less expressive logic to the more expressive one, completely characterises the class of models of the institution being represented. This means that ρ preserves the relation between the models obtained by the application of ρ^{Mod}.

The next definition extends the notion of institution co-morphism to structured specifications.

Definition 22 (Specification co-morphism (Borzyszkowski, 2002)). Let $\mathbb{I}$ and $\mathbb{I}'$ be institutions and let $\rho : \mathbb{I} \to \mathbb{I}'$ be an institution co-morphism. We define $\{\widehat{\rho}_\Sigma : \mathsf{Spec}^{\mathbb{I}}_\Sigma \to \mathsf{Spec}^{\mathbb{I}'}_{\rho^{Sign}(\Sigma)}\}_{\Sigma \in |\mathsf{Sign}|}$ as follows:

- if $SP = \langle \Sigma, \Gamma \rangle$, then $\widehat{\rho}_\Sigma(SP) = \left\langle \rho^{Sign}(\Sigma), \rho^{Sen}(\Gamma) \right\rangle$.
- if $SP = SP_1 \cup SP_2$, then $\widehat{\rho}_\Sigma(SP) = \widehat{\rho}_\Sigma(SP_1) \cup \widehat{\rho}_\Sigma(SP_2)$.
- let $\sigma_1 : \Sigma_1 \to \Sigma \in ||\mathsf{Sign}^{\mathbb{I}}||$, if $SP =$ **translate** SP_1 **by** σ_1, then $\widehat{\rho}_\Sigma(SP) =$ **translate** $\widehat{\rho}_{\Sigma_1}(SP_1)$ **by** $\rho^{Sign}(\sigma_1)$.
- let $\sigma_1 : \Sigma \to \Sigma_1 \in ||\mathsf{Sign}^{\mathbb{I}}||$, if $SP =$ **derive from** SP_1 **by** σ_1, then $\widehat{\rho}_\Sigma(SP) =$ **derive from** $\widehat{\rho}_{\Sigma_1}(SP_1)$ **by** $\rho^{Sign}(\sigma_1)$.

It is easy to see that if $SP \in \mathsf{Spec}^{\mathbb{I}}_\Sigma$ and $\rho : \mathbb{I} \to \mathbb{I}'$ is an institution co-morphism, then $\widehat{\rho}_\Sigma(SP) \in \mathsf{Spec}^{\mathbb{I}'}_{\rho^{Sign}(\Sigma)}$ (see (Borzyszkowski, 2002, Remark 5.13)). The following results, taken from (Borzyszkowski, 2002), show the relationships that exist between a specification and its translation to a different institution.

Theorem 8 ((Borzyszkowski, 2002)). *Let $\mathbb{I}$ and $\mathbb{I}'$ be institutions. Let $\rho : \mathbb{I} \to \mathbb{I}'$ be an institution co-morphism and $SP \in Spec^{\mathbb{I}}_\Sigma$. Then, if $\rho^{Sign} : \mathsf{Sign} \to \mathsf{Sign}'$ preserves pushouts, $\mathbf{nf}(\widehat{\rho}_\Sigma(SP)) \equiv \widehat{\rho}_\Sigma(\mathbf{nf}(SP))$.*

Corollary 4 ((Borzyszkowski, 2002)). *Let $\mathbb{I}$ and $\mathbb{I}'$ be institutions such that $\mathbb{I}'$ satisfies the weak-amalgamation property. Let $\rho : \mathbb{I} \to \mathbb{I}'$ be an institution co-morphism and $SP \in \mathsf{Spec}^{\mathbb{I}}_\Sigma$. Then, if $\rho^{Sign} : \mathsf{Sign} \to \mathsf{Sign}'$ preserves pushouts, $\widehat{\rho}_\Sigma(SP) \equiv \widehat{\rho}_\Sigma(\mathbf{nf}(SP))$.*

So far we have reviewed definitions and results relating structured specifications in one logic. The following results compare the classes of models obtained by translations across logics. The key concept in this comparison is the weak amalgamation of institution co-morphisms (Borzyszkowski, 2002).

Lemma 3 ((Borzyszkowski, 2002)).
Let $\mathbb{I}$ and $\mathbb{I}'$ be institutions. Let $\rho : \mathbb{I} \to \mathbb{I}'$ be an institution co-morphism and $SP \in \mathsf{Spec}^{\mathbb{I}}_\Sigma$. Then, $\rho^{Mod}_\Sigma(\mathbf{Mod}[\widehat{\rho}_\Sigma(SP)]) \subseteq \mathbf{Mod}[SP]$.

Lemma 4 ((Borzyszkowski, 2002)). *Let $\mathbb{I}$ and $\mathbb{I}'$ be institutions. Let $\rho : \mathbb{I} \to \mathbb{I}'$ be an institution co-morphism satisfying the weak amalgamation property, and $SP \in \mathsf{Spec}^{\mathbb{I}}_{\Sigma}$. Then, if for all $\mathcal{M} \in \mathbf{Mod}[SP]$ there exists $\mathcal{M}' \in \mathbf{Mod}[\widehat{\rho}_{\Sigma}(SP)]$ such that $\rho^{Mod}_{\Sigma}(\mathcal{M}') = \mathcal{M}$, $\mathbf{Mod}[SP] \subseteq \rho^{Mod}_{\Sigma}(\mathbf{Mod}[\widehat{\rho}_{\Sigma}(SP)])$.*

Finally, (Borzyszkowski, 2002, Thm. 7.1) proves that, given institutions $\mathbb{I}$ and $\mathbb{I}'$ and an institution co-morphism $\rho : \mathbb{I} \to \mathbb{I}'$ satisfying the weak amalgamation property such that the models of $\mathbb{I}$ have ρ-expansions, for all $SP \in \mathsf{Spec}^{\mathbb{I}}_{\Sigma}$ and $\alpha \in \mathbf{Sen}(\Sigma)$, $SP \models^{\Sigma} \alpha$ iff $\widehat{\rho}_{\Sigma}(SP) \models^{\rho^{Sign}(\Sigma)} \rho^{Sen}(\alpha)$. From this result, Borzyszkowski extends the calculus presented in Def. 16 with a new rule, obtaining a sound and complete proof system that enables the possibility of using institution co-morphisms in order to complete proofs:

$$\frac{\widehat{\rho}_{\Sigma}(SP) \vdash_{\rho^{Sign}(\Sigma)} \rho^{Sen}(\varphi)}{SP \vdash_{\Sigma} \varphi}\ [\rho\text{-entailment}] \tag{1}$$

Extending the calculi presented in Defs. 16 and 18 with rule *[ρ-entailment]* solves the problem of borrowing a proof system of one logic to prove properties of another. In our case we want to find a way of dealing with a multiplicity of logics linked by institution co-morphisms to a powerful proof system acting as “universal” institution. In this sense, the addition of this rule does not incorporate the possibility of structuring a specification around pieces written in different languages. In Section 4.2 we will address this limitation.

4.2 A Purely Heterogeneous Calculus

In this section we present the second contribution of this article; a calculus for reasoning about heterogeneous structured specifications. In (Diaconescu, 1998), the concept of extra theory morphisms was presented. This was the cornerstone of what later, in (Diaconescu, 2002), turned into a more sophisticated and solid construction presented under the name of Grothendieck institutions. Resorting to this construction the author established the relationship between specifications (theories) coming from different institutions. In (Mossakowski & Tarlecki, 2009), Mossakowski and Tarlecki address the problem of having structured specifications in an heterogeneous environment; solved by introducing *heterogeneous logical environments.*

Our approach relies on a definition close to the one presented in (Mossakowski & Tarlecki, 2009), with the sole exception of not considering institution morphisms among the arrows in the diagram but only focusing on institution co-morphisms. From now on we denote by $\mathbb{R}epr$ the category whose objects are institutions and whose morphisms are institution co-morphisms.

Definition 23 (Heterogeneous logical environment (Mossakowski & Tarlecki, 2009)). An *heterogeneous logical environment* is a diagram $D : G \to \mathbb{R}epr$. The class formed by this diagrams will be denoted as $\mathcal{HLE}$.

Linking theories through co-morphisms into a "universal" institution has two main consequences: 1. it provides a common language in which all the elements of a system specification can be interpreted, resulting in a joint semantics, and 2. the use of an appropriate "universal" institution enables the reuse of its proof system to prove properties, despite the language in which they were originally written. Definition 24 below extends the definition of SBOs (see Def. 11), to heterogeneous SBOs (in which the building operators act over an $\mathcal{HLE}$). Recalling notation, given a graph G, G_0 denotes its set of nodes and G_1 denotes the edges. For a given diagram $D : G \to \mathbb{R}epr$ and nodes $v, v' \in G_0$, $D(v)$ denotes an institution, and $D(\langle v, v' \rangle)$ denotes the co-morphism between institutions $D(v)$ and $D(v')$.

Definition 24 (Heterogeneous structure building operations). Let $D : G \to \mathbb{R}epr$ be an $\mathcal{HLE}$. The class of specifications over D is defined by extending Def. 11 as follows: Let $v, v' \in G_0$, $SP \in \mathsf{Spec}_{\Sigma}^{D(v)}$, $SP' \in \mathsf{Spec}_{\Sigma'}^{D(v')}$, $D(\langle v, v' \rangle)^{Sign}(\Sigma) = \Sigma'$, then

- if $SP' =$ **translate** SP **by** $D(\langle v, v' \rangle) \in \mathsf{Spec}_{\Sigma'}^{D(v')}$, then $\mathbf{Sig}[SP'] = \Sigma'$, and $\mathbf{Mod}[SP'] = \{\mathcal{M}' | D(\langle v, v' \rangle)_{\Sigma}^{Mod}(\mathcal{M}') \in \mathbf{Mod}[SP]\}$,
- if $SP =$ **derive from** SP' **by** $D(\langle v, v' \rangle) \in \mathsf{Spec}_{\Sigma}^{D(v)}$, then $\mathbf{Sig}[SP] = \Sigma$, and $\mathbf{Mod}[SP] = \{D(\langle v, v' \rangle)_{\Sigma}^{Mod}(\mathcal{M}') | \mathcal{M}' \in \mathbf{Mod}[SP']\}$.

Intuitively, the heterogeneous **translate** operator allows us to use an institution co-morphism in order to move from a given theory into a richer one. Similarly, the heterogeneous **derive** operator allows us to move from a rich theory to one that is represented in it.

Definition 25 (Heterogeneous specification co-morphism). Let $D : G \to \mathbb{R}epr$ be an $\mathcal{HLE}$ such that G is a join-semilattice with top. Then, for all $\langle v, v' \rangle \in G_1$, we extend the map between specifications of Def. 22 of $\{\widehat{D(\langle v, v' \rangle)}_{\Sigma} : Spec_{\Sigma} \to Spec_{D(\langle v, v' \rangle)^{Sign}(\Sigma)}\}_{\Sigma \in |\mathsf{Sign}^{D(v)}|}$ by adding the following rules: Let $\langle v, v' \rangle, \langle v', v'' \rangle \in G_1$, $SP \in Spec_{\Sigma}^{D(v)}$ and let $SP' \in Spec_{\Sigma'}^{D(v')}$, where $\Sigma' = D(\langle v, v' \rangle)^{Sign}(\Sigma)$

- if $SP' =$ **translate** SP **by** $D(\langle v, v' \rangle)$ then $\widehat{D(\langle v', v'' \rangle)}_{\Sigma}(SP') =$ **translate** SP **by** $D(\langle v, v' \rangle) \circ D(\langle v', v'' \rangle)$, and
- if $SP =$ **derive from** SP' **by** $D(\langle v, v' \rangle)$ then $\widehat{D(\langle v, v'' \rangle)}_{\Sigma}(SP) =$ **derive from** $\widehat{D(\langle v', v' \vee v'' \rangle)}_{\Sigma'}(SP')$ **by** $D(\langle v'', v' \vee v'' \rangle)$.

Definition 26. Let $D : G \to \mathbb{R}epr$ be an $\mathcal{HLE}$ such that G is a join-semilattice with top, $v, v_1, v_2 \in G_0$ and $\langle v_1, v \rangle, \langle v_2, v \rangle \in G_1$. Then, given specifications $SP_1 \in \mathsf{Spec}^{D(v_1)}_{\Sigma_1}$ and $SP_2 \in \mathsf{Spec}^{D(v_2)}_{\Sigma_2}$, we say that SP_1 is equivalent in $D(v)$ to SP_2 (denoted $SP_1 \equiv_{D(v)} SP_2$) if and only if $\widehat{D(\langle v_1, v\rangle)}_{\Sigma_1}(SP_1) \equiv \widehat{D(\langle v_2, v\rangle)}_{\Sigma_2}(SP_2)$.

Proposition 3. *Let $D : G \to \mathbb{R}epr$ be an $\mathcal{HLE}$ such that G is a join-semilattice with top.*

1. *For each $v \in G_0$, $\equiv_{D(v)}$ is an equivalence relation.*
2. *For each $v \in G_0$, if $SP_1 \equiv_{D(v)} SP'_1$ and $SP_2 \equiv_{D(v)} SP'_2$, then $SP_1 \cup SP_2 \equiv_{D(v)} SP'_1 \cup SP'_2$.*
3. *Let $v \in G_0$, $D(\langle v, v\rangle)$ the identity institution representation and $SP \in \mathsf{Spec}^{D(v)}_{\Sigma}$, then* **translate** *$SP$* **by** *$D(\langle v, v\rangle) \equiv_{D(v)} SP$.*
4. *Let $\langle v, v'\rangle \in G_1$ and $SP \in \mathsf{Spec}^{D(v)}_{\Sigma}$, then* **translate** *$SP$* **by** *$D(\langle v, v'\rangle) \equiv_{D(v')} \widehat{D(\langle v, v'\rangle)}_{\Sigma}(SP)$.*

Proof. 1. The proof follows by definition of $\equiv_{D(v)}$ for any given $v \in G_0$.

2. Let $SP_1 \equiv_{D(v)} SP'_1$ and $SP_2 \equiv_{D(v)} SP'_2$. Then,

$$\begin{array}{l} \widehat{D(\langle v_0, v\rangle)}_{\Sigma_0}(SP_1) \equiv \widehat{D(\langle v_1, v\rangle)}_{\Sigma_1}(SP'_1) \\ \widehat{D(\langle v_0, v\rangle)}_{\Sigma_0}(SP_2) \equiv \widehat{D(\langle v_1, v\rangle)}_{\Sigma_1}(SP'_2) \end{array} \tag{2}$$

Relation $\equiv$ stands for equivalence in institution $D(v)$ (c.f. Def. 13).

$$\begin{array}{ll} & SP_1 \cup SP_2 \equiv_{D(v)} SP'_1 \cup SP'_2 \\ \iff & \widehat{D(\langle v_0, v\rangle)}_{\Sigma_0}(SP_1 \cup SP_2) \equiv \widehat{D(\langle v_1, v\rangle)}_{\Sigma_1}(SP'_1 \cup SP'_2) \\ \iff & \widehat{D(\langle v_0, v\rangle)}_{\Sigma_0}(SP_1) \cup \widehat{D(\langle v_0, v\rangle)}_{\Sigma_0}(SP_2) \equiv \widehat{D(\langle v_1, v\rangle)}_{\Sigma_1}(SP'_1) \cup \widehat{D(\langle v_1, v\rangle)}_{\Sigma_1}(SP'_2) \ . \end{array}$$

In order to prove the equivalence we must prove that both sides coincide in signature and models, which immediately follows from (2).

3. Follows trivially by Defs. 24 and 26

4.

$$\begin{array}{ll} & \textbf{translate } SP \textbf{ by } D(\langle v, v'\rangle) \\ & \quad \text{[using that } D(\langle v, v\rangle) \text{ is the identity institution representation]} \\ = & \textbf{translate } SP \textbf{ by } D(\langle v, v\rangle) \circ D(\langle v, v'\rangle) \\ & \quad \text{[by Def. 25]} \\ = & \widehat{D(\langle v, v'\rangle)}_{\Sigma}(\textbf{translate } SP \textbf{ by } D(\langle v, v\rangle)) \\ & \quad \text{[by Prop. 3.3]} \\ \equiv_{D(v')} & \widehat{D(\langle v, v'\rangle)}_{\Sigma}(SP) \end{array}$$

The following definition extends the calculus presented in Def. 18 to $\mathcal{HLE}$.

Definition 27. Let $D : G \to \mathbb{R}epr$ be an $\mathcal{HLE}$ such that G is a join-semilattice with top. Then, the following rules define a D-indexed family of entailment relations:

$$v \in G_0 \wedge \langle \Sigma, \Gamma \rangle \in |\mathsf{Th}_0^{D(v)}| \; \frac{\Gamma \vdash^{D(v)^{\Sigma}} \varphi}{\langle \Sigma, \Gamma \rangle \vdash^{D(v)}_{\Sigma} \varphi} \text{[basic]} \; \frac{SP_2 \vdash^{D(v)}_{\Sigma} \varphi \qquad SP_1 \equiv SP_2}{SP_1 \vdash^{D(v)}_{\Sigma} \varphi} \text{[equiv]}$$

$$\frac{\{SP_1 \vdash^{D(v)}_{\Sigma} \psi\}_{\psi \in \Delta} \qquad \langle \Sigma, \Delta \rangle \cup SP_2 \vdash^{D(v)}_{\Sigma} \varphi}{SP_1 \cup SP_2 \vdash^{D(v)}_{\Sigma} \varphi} \text{[sum]}$$

$$\frac{SP \vdash^{D(v)}_{\Sigma} \varphi}{\textbf{translate } SP \textbf{ by } \sigma \vdash^{D(v)}_{\Sigma'} \mathbf{Sen}^{D(v)}(\sigma)(\varphi)} \text{[translate]} \; \frac{SP \vdash^{D(v)}_{\Sigma'} \mathbf{Sen}^{D(v)}(\sigma)(\varphi)}{\textbf{derive from } SP \textbf{ by } \sigma \vdash^{D(v)}_{\Sigma} \varphi} \text{[derive]}$$

$$\frac{\widehat{D(\langle v_2, v \rangle)}_{\Sigma_2}(SP_2) \vdash^{D(v)}_{D(\langle v_1, v \rangle)^{Sign}(\Sigma_1)} D(\langle v_1, v \rangle)^{Sen}_{\Sigma_1}(\varphi) \qquad SP_1 \equiv_{D(v)} SP_2}{SP_1 \vdash^{D(v_1)}_{\Sigma_1} \varphi} \text{[}\rho\text{-equiv],}$$

provided that $v \in G_0$, $\langle v_1, v \rangle, \langle v_2, v \rangle \in G_1$, and $D(\langle v_1, v \rangle)^{Sign}(\Sigma_1) = D(\langle v_2, v \rangle)^{Sign}(\Sigma_2)$.

$$\frac{SP \vdash^{D(v')}_{D(\left\langle v, v' \right\rangle)^{Sign}(\Sigma)} D(\left\langle v, v' \right\rangle)^{Sen}_{\Sigma}(\varphi)}{\textbf{derive from } SP \textbf{ by } D(\left\langle v, v' \right\rangle) \vdash^{D(v)}_{\Sigma} \varphi} \text{[}\rho\text{-derive]}$$

$$\frac{SP \vdash^{D(v)}_{\Sigma} \varphi}{\textbf{translate } SP \textbf{ by } D(\left\langle v, v' \right\rangle) \vdash^{D(v')}_{D(\left\langle v, v' \right\rangle)^{Sign}(\Sigma)} D(\left\langle v, v' \right\rangle)^{Sen}_{\Sigma}(\varphi)} \text{[}\rho\text{-translate]}$$

The last three rules are meant to deal with heterogeneity. Rule *[ρ-equiv]* is a version of *[equiv]* that enables the possibility of using a different proof system to complete the proof. Notice that Rule 1 is an instance of Rule *[ρ-equiv]* provided the proper instantiation of specifications. Rules *[ρ-derive]* and *[ρ-translate]* are added as mechanisms for dealing with the heterogeneous structure induced by the heterogeneous derive and translate operations.

Theorem 9 (Soundness). *Let $D : G \to \mathbb{R}epr$ be an $\mathcal{HLE}$ such that G is a join-semilattice with top. Assume also that for all $\langle v, v' \rangle \in G_1$, $\mathcal{M} \in |\mathbf{Mod}^{D(v)}(\langle \Sigma, \Gamma \rangle)|$ has the $D(\langle v, v' \rangle)$-expansion property. If $SP \vdash^{D(v)} \varphi$ then $SP \models^{D(v)} \varphi$, with $\vdash^{D(v)}$ defined via the rules presented in Def. 27.*

Proof. The soundness proof relies on the soundness of each one of the rules. The proof is completed by an induction on the height of the proof-tree. We know by Thm. 4 that rules *[basic]*, *[equiv]*, *[derive]*, *[sum]* and *[translate]* are sound with respect to the semantics. It then remains to prove that rules *ρ-equiv*, *ρ-derive* and *ρ-translate* are sound as well:

[ρ-equiv] Assume $\widehat{D(\langle v_2, v \rangle)}_{\Sigma_2}(SP_2) \models^{D(v)}_{D(\langle v_1, v \rangle)^{Sign}(\Sigma_1)} D(\langle v_1, v \rangle)^{Sen}_{\Sigma_1}(\varphi)$ and $SP_1 \equiv_v SP_2$. By Def. 26, $\widehat{D(\langle v_1, v \rangle)}_{\Sigma_1}(SP_1) \equiv \widehat{D(\langle v_2, v \rangle)}_{\Sigma_2}(SP_2)$. Therefore, $\widehat{D(\langle v_1, v \rangle)}_{\Sigma_1}(SP_1) \models^{D(v)}_{D(\langle v_1, v \rangle)^{Sign}(\Sigma_1)} D(\langle v_1, v \rangle)^{Sen}_{\Sigma_1}(\varphi)$. By Thm. 7, since each $\mathcal{M} \in \mathbf{Mod}[SP_1]$ has the $D(\langle v_1, v \rangle)$-expansion property, $SP_1 \models^{D(v_1)}_{\Sigma_1} \varphi$,

[ρ-derive] Assume $SP' \models^{D(v')}_{D(\langle v, v' \rangle)^{Sign}(\Sigma)} D(\langle v, v' \rangle)^{Sen}_{\Sigma}(\varphi)$. Then, for all $\mathcal{M} \in \mathbf{Mod}[SP']$, $\mathcal{M} \models^{D(v')}_{D(\langle v, v' \rangle)^{Sign}(\Sigma)} D(\langle v, v' \rangle)^{Sen}_{\Sigma}(\varphi)$ which, by the satisfaction condition for institution co-morphisms (see Def. 19), is equivalent to $D(\langle v, v' \rangle)^{Mod}_{\Sigma}(\mathcal{M}) \models^{D(v)}_{\Sigma} \varphi$. Then, for all $\mathcal{M} = D(\langle v, v' \rangle)^{Mod}_{\Sigma}(\mathcal{M}')$

with $\mathcal{M}' \in \mathbf{Mod}[SP']$, $\mathcal{M} \models^{D(v)}_{\Sigma} \varphi$. Finally, by Def. 24, for all $\mathcal{M} \in \mathbf{Mod}[\textbf{derive from } SP' \textbf{ by } D(\langle v, v' \rangle)]$, $\mathcal{M} \models^{D(v)}_{\Sigma} \varphi$, and, consequently, **derive from** SP' **by** $D(\langle v, v' \rangle)] \models^{D(v)}_{\Sigma} \varphi$, and

[ρ**-translate**] Assume $SP \models^{D(v)} \varphi$. For all $\mathcal{M} \in \mathbf{Mod}[SP]$, $\mathcal{M} \models^{D(v)}_{\Sigma} \varphi$. By $D(\langle v, v' \rangle)$-expansion, for each $\mathcal{M} \in \mathbf{Mod}[SP]$ there exists $\mathcal{M}'$ such that $D(\langle v, v' \rangle)^{Mod}_{\Sigma}(\mathcal{M}') = \mathcal{M}$. Therefore, for each $\mathcal{M} \in \mathbf{Mod}[SP]$, there exists $\mathcal{M}'$ such that $D(\langle v, v' \rangle)^{Mod}_{\Sigma}(\mathcal{M}') \models^{D(v)}_{\Sigma} \varphi$. By the satisfaction condition for institution co-morphisms (see Def. 19), $\mathcal{M}' \models^{D(v')}_{D(\langle v, v' \rangle)^{Sign}(\Sigma)} D(\langle v, v' \rangle)^{Sen}_{\Sigma}(\varphi)$. Thus, for all $\mathcal{M}'$ such that $D(\langle v, v' \rangle)^{Mod}_{\Sigma}(\mathcal{M}') \in \mathbf{Mod}[SP]$, $\mathcal{M}' \models^{D(v')}_{D(\langle v, v' \rangle)^{Sign}(\Sigma)} D(\langle v, v' \rangle)^{Sen}_{\Sigma}(\varphi)$. By Def. 24, for all $\mathcal{M}' \in \mathbf{Mod}[\textbf{translate } SP \textbf{ by } D(\langle v, v' \rangle)]$, $\mathcal{M}' \models^{D(v')}_{D(\langle v, v' \rangle)^{Sign}(\Sigma)} D(\langle v, v' \rangle)^{Sen}_{\Sigma}(\varphi)$. Then, we obtain that:
translate SP **by** $D(\langle v, v' \rangle) \models^{D(v')}_{D(\langle v, v' \rangle)^{Sign}(\Sigma)} D(\langle v, v' \rangle)^{Sen}_{\Sigma}(\varphi)$.

Theorem 10 (Completeness). *Let $D : G \to \mathbb{R}epr$ be an $\mathcal{HLE}$ such that G is a join-semilattice with top $\top$. Assume that for all $\langle v, \top \rangle \in G_1$, $\mathcal{M} \in \mathbf{Mod}^{D(v)}[\langle \Sigma, \Gamma \rangle]$ has the $D(\langle v, \top \rangle)$-expansion property. Then, if $\vdash_{D(\top)}$ (the calculus for non-structured specifications of institution $D(\top)$) is complete and $D(\top)$ has the weak-interpolation and weak-amalgamation properties, $SP \models^{D(v)} \varphi$ implies that $SP \vdash^{D(v)} \varphi$, with $\vdash^{D(v)}$ defined by the rules presented in Def. 27.*

Proof. The calculus in Def. 27 extends the calculus presented in Def. 18. Then, by Thm. 5, $\vdash^{D(\top)}$ (the proof system for structured specifications in institution $D(\top)$), is complete. As for all $\langle v, \top \rangle \in G_1$, $\mathcal{M} \in \mathbf{Mod}^{D(v)}[\langle \Sigma, \Gamma \rangle]$ has the $D(\langle v, \top \rangle)$-expansion property, by Thm. 7, we obtain $SP \models^{D(v)}_{\Sigma} \varphi$ iff $\widehat{D(\langle v, \top \rangle)}_{\Sigma}(SP) \models^{D(\top)}_{D(\langle v, \top \rangle)^{Sign}(\Sigma)} D(\langle v, \top \rangle)^{Sen}_{\Sigma}(\varphi)$. Then, as $\vdash^{D(\top)}$ is complete, $\widehat{D(\langle v, \top \rangle)}_{\Sigma}(SP) \vdash^{D(\top)}_{D(\langle v, \top \rangle)^{Sign}(\Sigma)} D(\langle v, \top \rangle)^{Sen}_{\Sigma}(\varphi)$.

The following derivation using *[ρ-equiv]*, and Def. 26, completes the proof of the theorem:

$$\frac{\widehat{D(\langle v, \top \rangle)}_{\Sigma}(SP) \vdash^{D(\top)}_{D(\langle v, \top \rangle)^{Sign}(\Sigma)} D(\langle v, \top \rangle)^{Sen}_{\Sigma}(\varphi) \qquad SP \equiv_{\top} \widehat{D(\langle v, \top \rangle)}_{\Sigma}(SP)}{SP \vdash^{D(v)}_{\Sigma} \varphi}\ [\rho\text{-equiv}]$$

provided that $\langle v, \top \rangle \in G_1$.

The reader should note that this calculus and completeness proof admit the exact same considerations we made about the calculus presented in Def. 18 and Thm. 5 as the calculus in Def. 27 is just an heterogeneous extension of that of Def. 18.

4.3 Fork algebras as a "universal" institution

Fork algebras, presented by Haeberer and Veloso in (Haeberer & Veloso, 1991), are an extension of relation algebras obtained by adding a new operator called *fork* (typically represented as "∇"). They arose in the search for a formalism suitable for software specification and verification and have been used for program repre- sentation and derivation (Frias, Baum, & Haeberer, 1998). In (Frias, 2002, Chapter 3, pp. 20) Frias gave a detailed discussion on the evolution of fork algebras and called our attention to the concepts which were responsible of such evolution. Other attractive features of this class of algebras are that they are isomorphic to algebras whose domain is a set of binary relations (Frias et al. (Frias, Baum, & Haeberer, 1997) and Gyuris (Gyuris, 1997)), that they posses a complete equational calculus with finitely many proof rules (Frias et al. in (Frias, Haeberer, & Veloso, 1997) and (Frias, 2002, Chapter 4)), were proved to provide a general method for constructing Rasiowa-Sikorski-style deduction systems for nonclassical logics (Frias & Orlowska, 1997), and have enough expressive power to serve as the target of interpretations of several logical languages. Figure 1 depicts several interpretability results proved for a number of logics. Among those which interpret logics in extensions of Fork Algebras we can find, first-order logic with equality (Frias, 2002, Chapter 5), modal logics and propositional dynamic logic (Frias & Orlowska, 1998), first-order dynamic logic (Frias et al., 2002), propositional linear temporal logic (Frias & Lopez Pombo, 2003) and first-order linear temporal logic (Frias & Lopez Pombo, 2006).

In all cases the class of fork algebras (Frias, 2002) was used as the target of the interpretation. In (Lopez Pombo & Frias, 2006) we showed that these algebras provide an institution that is a good candidate as a "universal" institution.

Several of these results were obtained in colaboration with Ewa Orlowska. It was Ewa, with the humility and sharpness that always characterized her, who approached Frias while he was a graduate student and suggested that fork algebras might be useful to reason in non-classical logics as well. This led to several years of fruitful cooperation.

Recalling Thm. 10, the candidate to "universal" institution must have a complete calculus and satisfy the weak-interpolation and weak-amalgamation properties. Proposition 4 shows that this is indeed the case for the logic of fork algebras, defined in detail in (Lopez Pombo & Frias, 2006).

Proposition 4. *Let* $\mathsf{FA} = \left\langle \mathsf{Sign}, \mathbf{Sen}, \mathbf{Mod}, \left\{\vdash^{\Sigma}_{\mathsf{FA}}\right\}_{\Sigma \in |\mathsf{Sign}|}, \left\{\models^{\Sigma}_{\mathsf{FA}}\right\}_{\Sigma \in |\mathsf{Sign}|} \right\rangle$ *be the logic of fork algebras. Then,*

1. FA *has the weak-interpolation property,*
2. FA *has the weak-amalgamation property, and*
3. FA *has a complete calculus for flat specifications.*

Proof.

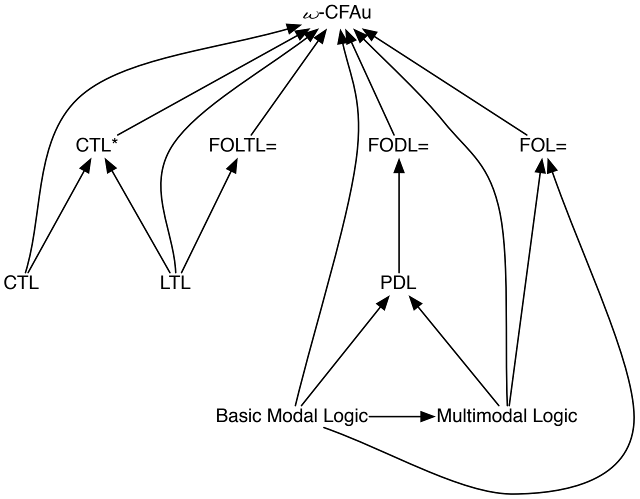

Fig. 1 Interpretability results of logics int Fork Algebras.

1. As it was done in (Roşu & Goguen, 2000), and formalised in the context of institutions by introducing $(\mathcal{D}, \mathcal{T})$-institutions (Borzyszkowski, 2002), we restrict ourselves to pushouts $\langle t'_1 : \Sigma_1 \to \Sigma', t'_2 : \Sigma_2 \to \Sigma' \rangle$ for $\langle t_1 : \Sigma \to \Sigma_1, t_2 : \Sigma \to \Sigma_2 \rangle$ in Sign such that $t'_2 : \Sigma \to \Sigma_2$ is injective. Then, the proof follows from (Roşu & Goguen, 2000, Coro. 6).
2. Let $\langle t'_1 : \Sigma_1 \to \Sigma', t'_2 : \Sigma_2 \to \Sigma' \rangle$ be a pushout in Sign for the span $\langle t_1 : \Sigma \to \Sigma_1, t_2 : \Sigma \to \Sigma_2 \rangle$, and $\mathcal{M}_1 = \left\langle M_1, \left\{ f_i^{\mathcal{M}_1} \right\}_{i \in \mathcal{I}_1} \right\rangle \in |\mathbf{Mod}\,(\Sigma_1)|$ and $\mathcal{M}_2 = \left\langle M_2, \left\{ f_i^{\mathcal{M}_2} \right\}_{i \in \mathcal{I}_2} \right\rangle \in |\mathbf{Mod}\,(\Sigma_2)|$, such that $\mathbf{Mod}\,(t_1)\,(\mathcal{M}_1) = \mathbf{Mod}\,(t_2)\,(\mathcal{M}_2)$. Let $\mathcal{M} = \left\langle M, \left\{ f_i^{\mathcal{M}} \right\}_{i \in \mathcal{I}} \right\rangle \in |\mathbf{Mod}\,(\Sigma)|$ such that $\mathcal{M} = \mathbf{Mod}\,(t_1)\,(\mathcal{M}_1) = \mathbf{Mod}\,(t_2)\,(\mathcal{M}_2)$. Define $\mathcal{M}' = \left\langle M, \left\{ {f'}_i^{\mathcal{M}'} \right\}_{i \in \mathcal{I}'} \right\rangle \in |\mathbf{Mod}\,(\Sigma')|$ such that:
   - $\mathcal{I}' = \mathcal{J} \cup \mathcal{J}_1 \cup \mathcal{J}_2$, (for the sake of simplifying notation let us assume that $\mathcal{I}, \mathcal{I}_1, \mathcal{I}_2$ are mutually disjoint), such that
     - $\mathcal{J} = \mathcal{I}$,
     - $\mathcal{J}_1 = \mathcal{I}_1 \setminus \{\, j \in \mathcal{I}_1 \mid \exists i \in \mathcal{I} \text{ s.t. } t_1(f_i) = f_j \,\}$,
     - $\mathcal{J}_2 = \mathcal{I}_2 \setminus \{\, j \in \mathcal{I}_2 \mid \exists i \in \mathcal{I} \text{ s.t. } t_2(f_i) = f_j \,\}$.
   - ${f'}_j^{\mathcal{M}'} = \begin{cases} f_j^{\mathcal{M}}, & \text{for all } j \in \mathcal{J}, \\ f_j^{\mathcal{M}_1}, & \text{for all } j \in \mathcal{J}_1, \\ f_j^{\mathcal{M}_2}, & \text{for all } j \in \mathcal{J}_2\ . \end{cases}$

   By construction $\mathbf{Mod}\,(t'_1)\,(\mathcal{M}') = \mathcal{M}_1$ and $\mathbf{Mod}\,(t'_2)\,(\mathcal{M}') = \mathcal{M}_2$.
3. Follows as a consequence of representability in (Frias et al., 2002, Thm. 3).

Next example shows the use of the calculus presented in Def. 27 to prove a property from an heterogeneous structured specification involving the languages of *Propositional Dynamic Logic* (Harel et al., 2000), *Linear Temporal Logic* (Pnueli, 1981), and using the *Fork algebras* (Frias, 2002) as a "universal" institution.

Example 2 (Reasoning in an heterogeneous logical environment). Let us consider the following simplified specification for the behavior of an electric golf car. The car can be started (in which case the engine is on), can be accelerated (which takes the car from being stopped to being in cruise speed), it can be stopped by pressing the brake pedal, and can be turned off. In order to write the model we will use state variables:

- $\mathit{engine\_is\_off}$ (modeling whether the engine is off), and
- $\mathit{speed\_is\_zero}$ (modeling whether the car is actually stopped).

This is a simplified model that, in particular, does not take into consideration electricity consumption or the need to recharge the car's battery. We will resort to a model describing the atomic actions in the specification, written in the propositional dynamic logic PDL. We only focus on state changes, being the remaining transitions invariant, i.e., $\neg \mathit{engine\_is\_off} \Rightarrow [\mathbf{Start}]\neg \mathit{engine\_is\_off}$. We use an LTL formula to impose a runtime constraint: cars must eventually turn their engines off (this may be due to a driver's decision or to lack of battery charge, but we will not model these aspects of the problem in the specification). The specification is the following:

PDL-Ax	$\mathit{init} \Rightarrow \mathit{engine\_is\_off}$ $\mathit{init} \Rightarrow \mathit{speed\_is\_zero}$ $\mathit{engine\_is\_off} \Rightarrow [\mathbf{Start}]\neg \mathit{engine\_is\_off}$ $(\mathit{speed\_is\_zero} \wedge \neg \mathit{engine\_is\_off}) \Rightarrow [\mathbf{Accelerate}]\neg \mathit{speed\_is\_zero}$ $\neg \mathit{speed\_is\_zero} \Rightarrow [\mathbf{Break}]\mathit{speed\_is\_zero}$ $\neg \mathit{engine\_is\_off} \Rightarrow [\mathbf{TurnOff}]\mathit{engine\_is\_off}$ $\neg \mathit{speed\_is\_zero} \Rightarrow [\mathbf{TurnOff}]\mathit{speed\_is\_zero}$
LTL-Ax	$\mathit{init} \Rightarrow \Box\,(\neg \mathit{engine\_is\_off} \Rightarrow \Diamond \mathit{engine\_is\_off})$

Let us consider the PDL signature

$$\Sigma_{\mathsf{PDL}} = \langle\{\mathbf{Start}, \mathbf{Accelerate}, \mathbf{Break}, \mathbf{TurnOff}\}, \{\mathit{init}, \mathit{engine\_is\_off}, \mathit{speed\_is\_zero}\}\rangle$$

and the LTL signature

$$\Sigma_{\mathsf{LTL}} = \langle\{\mathit{init}, \mathit{engine\_is\_off}, \mathit{speed\_is\_zero}\}\rangle \ .$$

Let us have a fork algebra signature

$$\Sigma_{\mathsf{FA}} = \langle\{\mathbf{S_0}, \mathbf{S}, \mathbf{tr}, \mathbf{T}, \mathbf{Start}, \mathbf{Accelerate}, \mathbf{Break}, \mathbf{TurnOff}, \mathit{init}, \mathit{engine\_is\_off}, \mathit{speed\_is\_zero}\}\rangle \ .$$

Let $\widehat{\mathsf{PDL} \to \mathsf{FA}}^{Sign}(\Sigma_{\mathsf{PDL}}) = \Sigma_{\mathsf{FA}}$ and $\widehat{\mathsf{LTL} \to \mathsf{FA}}^{Sign}(\Sigma_{\mathsf{LTL}}) = \Sigma_{\mathsf{FA}}$. We will not provide definitions for translations $\widehat{\mathsf{PDL} \to \mathsf{FA}}^{Sen}$ and $\widehat{\mathsf{LTL} \to \mathsf{FA}}^{Sen}$ in this article. Fully detailed definitions have been provided in (Frias, 2002) and (Frias & Lopez Pombo, 2003), respectively.

While there is a clear relationship between atomic actions in the PDL model and the transition relation in the LTL model, this relationship has to be made formal. Let us consider the specification

$$SP_{\mathsf{FA}} = \langle \Sigma_{\mathsf{FA}}, \{\mathbf{T} = \mathbf{Start} + \mathbf{Accelerate} + \mathbf{Break} + \mathbf{TurnOff}\}\rangle \ .$$

Let us consider the diagram:

$$\begin{tikzcd}
& \mathsf{FA} & \\
\mathsf{PDL} \arrow[ur, "\widehat{\mathsf{PDL} \to \mathsf{FA}}_{\Sigma_{\mathsf{PDL}}}"] & & \mathsf{LTL} \arrow[ul, "\widehat{\mathsf{LTL} \to \mathsf{FA}}_{\Sigma_{\mathsf{LTL}}}"']
\end{tikzcd}$$

Scenario 1: We want to prove an LTL assertion establishing that cars must eventually reach a motionless state.

$$\mathsf{Assert} : init \Rightarrow \Box\,(\neg speed\_is\_zero \Rightarrow \Diamond speed\_is\_zero)\,.$$

Verifying that the assertion holds reduces then to proving that:

$$\left(\begin{array}{l}\textbf{translate}\ \langle \Sigma_{\mathsf{PDL}}, \mathsf{PDL\text{-}Ax}\rangle\ \textbf{by}\ \widehat{\mathsf{PDL} \to \mathsf{FA}}_{\Sigma_{\mathsf{PDL}}} \cup \\ \textbf{translate}\ \langle \Sigma_{\mathsf{LTL}}, \mathsf{LTL\text{-}Ax}\rangle\ \textbf{by}\ \widehat{\mathsf{LTL} \to \mathsf{FA}}_{\Sigma_{\mathsf{LTL}}} \cup \\ SP_{\mathsf{FA}}\end{array}\right) \vdash^{\mathsf{FA}}_{\Sigma_{\mathsf{FA}}} \widehat{\mathsf{LTL} \to \mathsf{FA}}^{Sen}_{\Sigma_{\mathsf{LTL}}}(\mathsf{Assert})$$

In this scenario we move from the constituent specifications to their representations in the universal institution. Let us define:

$$\begin{array}{ll}
tr_1 & := \textbf{translate}\ \langle \Sigma_{\mathsf{PDL}}, \mathsf{PDL\text{-}Ax}\rangle\ \textbf{by}\ \widehat{\mathsf{PDL} \to \mathsf{FA}}_{\Sigma_{\mathsf{PDL}}}, \\
tr_2 & := \textbf{translate}\ \langle \Sigma_{\mathsf{LTL}}, \mathsf{LTL\text{-}Ax}\rangle\ \textbf{by}\ \widehat{\mathsf{LTL} \to \mathsf{FA}}_{\Sigma_{\mathsf{LTL}}}, \\
ass & := \widehat{\mathsf{LTL} \to \mathsf{FA}}^{Sen}_{\Sigma_{\mathsf{LTL}}}(\mathsf{Assert})\,, \\
P2F & := \widehat{\mathsf{PDL} \to \mathsf{FA}}_{\Sigma_{\mathsf{PDL}}}(\langle \Sigma_{\mathsf{PDL}}, \mathsf{PDL\text{-}Ax}\rangle), \\
L2F & := \widehat{\mathsf{LTL} \to \mathsf{FA}}_{\Sigma_{\mathsf{LTL}}}(\langle \Sigma_{\mathsf{LTL}}, \mathsf{LTL\text{-}Ax}\rangle).
\end{array}$$

The derivation then proceeds as follows (we make extensive use of Prop. 3):

$$\dfrac{P2F \cup L2F \cup SP_{\mathsf{FA}} \vdash^{\mathsf{FA}}_{\Sigma_{\mathsf{FA}}} ass \qquad P2F \cup tr_2 \cup SP_{\mathsf{FA}} \equiv_{\mathsf{FA}} P2F \cup L2F \cup SP_{\mathsf{FA}}}{\textbf{(1)}\ P2F \cup tr_2 \cup SP_{\mathsf{FA}} \vdash^{\mathsf{FA}}_{\Sigma_{\mathsf{FA}}} ass}\ [\rho\text{-eq}]$$

$$\dfrac{\textbf{(1)} \qquad tr_1 \cup tr_2 \cup SP_{\mathsf{FA}} \equiv_{\mathsf{FA}} P2F \cup tr_2 \cup SP_{\mathsf{FA}}}{tr_1 \cup tr_2 \cup SP_{\mathsf{FA}} \vdash^{\mathsf{FA}}_{\Sigma_{\mathsf{FA}}} ass}\ [\rho\text{-eq}]$$

The derivation is completed by proving $P2F \cup L2F \cup SP_{\mathsf{FA}} \vdash^{\mathsf{FA}}_{\Sigma_{\mathsf{FA}}} ass$. In doing so, we loose track of the original specifications. Notice also that, since no mechanism exist to build specifications from pieces coming from different institutions, rule [ρ-entailment] is of no practical use in this example.

Scenario 2: Let us suppose that along the verification of the assertion in Scenario 1, it becomes necessary to prove the fork algebra equation

$$1'_{St}\,;\overline{init}\ +\ 1'_{St}\,;\overline{(\mathbf{Accelerate}+\mathbf{Break}+\mathbf{TurnOff})^{*}\,;\overline{engine\_is\_off}} = 1'_{St}\,;1\ . \qquad (3)$$

According to the representation map presented in (Frias, 2002, Def. 6.25), formula (3) corresponds to

$$\widehat{\mathsf{PDL} \to \mathsf{FA}}^{Sen}_{\Sigma_{\mathsf{PDL}}}\Big(init\ \Rightarrow\ [(\mathbf{Accelerate}+\mathbf{Break}+\mathbf{TurnOff})^{*}]\,engine\_is\_off\Big)\ .$$

Since

$$\langle \Sigma_{\mathsf{PDL}}, \mathsf{PDL\text{-}Ax}\rangle \vdash^{\mathsf{PDL}}_{\Sigma_{\mathsf{PDL}}} init\ \Rightarrow\ [(\mathbf{Accelerate}+\mathbf{Break}+\mathbf{TurnOff})^{*}]\,engine\_is\_off\,,$$

we can proceed as follows:

$$\frac{\langle \Sigma_{\mathsf{PDL}}, \mathsf{PDL\text{-}Ax}\rangle \vdash^{\mathsf{PDL}}_{\Sigma_{\mathsf{PDL}}} init \Rightarrow [(\mathbf{Accelerate}+\mathbf{Break}+\mathbf{TurnOff})^{*}]\,engine\_is\_off}{\begin{array}{c}\textbf{translate}\ \langle \Sigma_{\mathsf{PDL}}, \mathsf{PDL\text{-}Ax}\rangle\ \textbf{by}\ \widehat{\mathsf{PDL} \to \mathsf{FA}}_{\Sigma_{\mathsf{PDL}}}\\ \vdash^{\mathsf{FA}}_{\Sigma_{\mathsf{FA}}}\\ \widehat{\mathsf{PDL} \to \mathsf{FA}}^{Sen}_{\Sigma_{\mathsf{PDL}}}(init \Rightarrow [(\mathbf{Accelerate}+\mathbf{Break}+\mathbf{TurnOff})^{*}]\,engine\_is\_off)\end{array}}\ [\rho\text{-tran.}]$$

Therefore, proving fork algebra properties that follow from some of the constituent specifications, reduces to proving the properties in the corresponding constituent institutions. Notice that such reasoning cannot be made in Borzyszkowski's calculus. The only rule that allows one to move between (different) institutions is rule [ρ-entailment], and it only allows one to move to a richer institution, which is not the case in this example.

Scenario 3: Let us go back to Scenario 1, but this time let us prove sequent

$$\langle \Sigma_{\mathsf{PDL}}, \mathsf{PDL\text{-}Ax}\rangle \cup \langle \Sigma_{\mathsf{LTL}}, \mathsf{LTL\text{-}Ax}\rangle \cup SP_{\mathsf{FA}} \vdash^{\mathsf{LTL}}_{\Sigma_{\mathsf{LTL}}} \mathsf{Assert}\ . \qquad (4)$$

Since union only applies to specifications from the same institution, sequent (4) is ill-formed. Yet it serves the purpose of illustrating a common need in software specification: proving a property in one institution using partial knowledge coming from other institutions. This cannot be done using the extension of Borzyszkowski's calculus with rule [ρ-entailment].

Using the structuring constructs [ρ-translate] and [ρ-derive] we can rewrite sequent (4) as follows (we use the same notation we used in Scenario 1):

$$\textbf{derive from}\ \left(tr_1 \cup tr_2 \cup SP_{\mathsf{FA}}\right)\ \textbf{by}\ \widehat{\mathsf{LTL} \to \mathsf{FA}}_{\Sigma_{\mathsf{LTL}}} \vdash^{\mathsf{LTL}}_{\Sigma_{\mathsf{LTL}}} \mathsf{Assert}\ . \qquad (5)$$

Notice that this is a very appropriate way of reflecting the intention expressed in sequent (4). From (5) we can proceed as follows:

$$\dfrac{tr_1 \cup tr_2 \cup SP_{\mathsf{FA}} \vdash^{\mathsf{FA}}_{\Sigma_{\mathsf{FA}}} \widehat{\mathsf{LTL} \to \mathsf{FA}}_{\Sigma_{\mathsf{LTL}}}(\mathsf{Assert})}{\textbf{derive from } (tr_1 \cup tr_2 \cup SP_{\mathsf{FA}}) \textbf{ by } \widehat{\mathsf{LTL} \to \mathsf{FA}}_{\Sigma_{\mathsf{LTL}}} \vdash^{\mathsf{LTL}}_{\Sigma_{\mathsf{LTL}}} \mathsf{Assert}} \;[\rho\text{-derive}]$$

Since $\widehat{\mathsf{LTL} \to \mathsf{FA}}_{\Sigma_{\mathsf{LTL}}}(\mathsf{Assert}) = ass$, the upper sequent in the proof is the same sequent we can find in Scenario 1. Therefore, the proof can proceed in the same way.

5 Conclusions

We analyzed Borzyszkowski's work on structured specifications and showed that the conditions that a logical system must meet for having a complete calculus are too restrictive and do not apply to some of the most popular logics in software specification. A consequence of this restrictiveness is that meaningful logical systems are not covered by Borzyszkowski's calculus. In order to overcome these limitations we presented a calculus for structured specifications whose completeness does not require interpolation or combinations of properties that result in equally restrictive calculi. Borzyszkowski's calculus was proved complete for finite structured specifications. In logics possessing (reflexive-)transitive closure, finite specifications may be useful but many times are not sufficient. We have presented an example where infinite specifications are required. The appropriateness of our proposal is supported by a methodological discussion on the relevance of having such calculus.

We introduced structure building operators suitable for structuring specifications in an heterogeneous logical system based on the "universal" institution approach. We analyzed this proposal in the light of the existing work on heterogeneous specifications, and concluded that it provides essential features not shared by Borzyszkowski's calculus. For instance, while Borzyszkowski's heterogeneous calculus allows one to borrow a calculus from a richer institution, we allow to move to the most suitable institution in which the proof should be made (even if this "most suitable" institution is poorer than the original one).

References

Abadi, M. (1988, August). *The power of temporal proofs* (Technical Report No. 30). System Research Center, Palo Alto, CA 94301 USA: System Research Center, Digital.

Abadi, M., & Manna, Z. (1990). Nonclausal deduction in first-order temporal logic. *Journal of the ACM*, *37*(2), 279–317.

Astesiano, E., & Cerioli, M. (1991). Relationships between logical frameworks. In M. Bidoit & C. Choppy (Eds.), *Selected papers from the 8th workshop on specification of abstract data types joint with the 3rd compass workshop on recent trends in data type specification* (Vol. 655, pp. 126–143). Springer-Verlag.

Bergstra, J. A., Heering, J., & Klint, P. (1990). Module algebra. *Journal of the ACM*, *37*(2), 335–372.

Booch, G., Rumbaugh, J., & Jacobson, I. (1998). *The unified modeling language user guide.* Boston, MA, USA: Addison–Wesley Longman Publishing Co., Inc.

Borzyszkowski, T. (1997, June). Completeness of a logical system for structured specifications. In F. Parisi-Presicce (Ed.), *Proceedings of the 12th. international workshop on recent trends in algebraic development techniques WADT 1997* (Vol. 1376, pp. 107–121). Tarquinia, Italy: Springer-Verlag.

Borzyszkowski, T. (1998, April). Moving specification structures between logical systems. In J. L. Fiadeiro (Ed.), *Proceedings of the 13th. international workshop on recent trends in algebraic development techniques WADT 1998* (Vol. 1589, pp. 16–30). Lisbon, Portugal: Springer-Verlag.

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

Broy, M., & Cengarle, M. V. (2011). Uml formal semantics: lessons learned. *Software and System Modeling*, *10*(4), 441–446.

Cengarle, M. V., Knapp, A., Tarlecki, A., & Wirsing, M. (2008). A heterogeneous approach to UML semantics. In P. Degano, R. DeNicola, & J. Meseguer (Eds.), *Proceedings of concurrency, graphs and models (essays dedicated to Ugo Montanari on the occasion of his 65th. birthday)* (pp. 383–402). Edinburgh, Scotland: Springer-Verlag.

Cerioli, M. (1993). *Relationships between logical formalisms* (Unpublished doctoral dissertation). Dipartamento di informatica, Universitá degli studi di Pisa. (Ph.D. Thesis: TD-4/93, Dottorato di ricerca in informatica, Universitá di Pisa-Genova-Udine)

Cerioli, M., & Meseguer, J. (1997). May i borrow your logic? (transporting logical structures along maps). *Theoretical Computer Science*, *173*(2), 311–347.

Clarke, E. M., Emerson, E. A., & Sistla, A. P. (1986). Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, *8*(2), 244–263.

Craig, W. (1957). Three uses of the herbrand-gentzen theorem in relating model theory and proof theory. *Journal of Symbolic Logic*, *22*(3), 269–285.

de Lavalette, G. R., Kooi, B., & Verbrugge, R. (2008). Strong completeness and limited canonicity for pdl. *Journal of logic, language and information*, *17*(1), 69–87.

Diaconescu, R. (1998). Extra theory morphisms for institutions: logical semantics for multi-paradigm languages. *Applied Categorical Structures*, *6*(4), 427–453.

Diaconescu, R. (2002). Grothendieck institutions. *Applied Categorical Structures*, *10*(4), 383–402.

Diaconescu, R. (Ed.). (2008). *Institution-independent model theory* (Vol. 2). Birkhäuser.

Diaconescu, R., & Futatsugi, K. (2002). Logical foundations of CafeOBJ. *Theoretical Computer Science*, *285*(2), 289–318.

Diaconescu, R., Goguen, J. A., & Stefaneas, P. (1993). Logical support for modularisation. In G. Huet & G. Plotkin (Eds.), *Proceedings of the 2nd. annual workshop on logical environments* (pp. 83–130). Edinburgh, Scotland: Cambridge University Press.

Dimitrakos, T., & Maibaum, T. S. E. (2000). On a generalized modularization theorem. *Information Processing Letters*, *74*(1–2), 65–71.

Emerson, E. A., & Halpern, J. Y. (1985). Decision procedures and expressiveness in the temporal logic of branching time. *Journal of Computer and System Sciences*, *30*(1), 1–24.

Emerson, E. A., & Halpern, J. Y. (1986). "sometimes" and "not never" revisited: on branching versus linear time temporal logic. *Journal of the ACM*, *33*(1), 151–178.

Fine, K. (1979). Failures of the interpolation lemma in quantified modal logic. *Journal of Symbolic Logic*, *44*(2), 201–206.

Frias, M. F. (2002). *Fork algebras in algebra, logic and computer science* (Vol. 2). Singapore: World Scientific Publishing Co.

Frias, M. F., Baum, G. A., & Haeberer, A. M. (1997). Fork algebras in algebra, logic and computer science. *Fundamenta Informaticae*, *32*, 1–25.

Frias, M. F., Baum, G. A., & Haeberer, A. M. (1998). Representability and program construction within fork algebras. *Logic Journal of the IGPL*, *6*(2), 227–257.

Frias, M. F., Baum, G. A., & Maibaum, T. S. E. (2002, October). Interpretability of first-order dynamic logic in a relational calculus. In H. de Swart (Ed.), *Proceedings of the 6th. conference on relational methods in computer science (RelMiCS) - TARSKI* (Vol. 2561, pp. 66–80). Oisterwijk, The Netherlands: Springer-Verlag.

Frias, M. F., Haeberer, A. M., & Veloso, P. A. (1997). A finite axiomatization for fork algebras. *Logic Journal of the IGPL*, *5*(3), 311–319.

Frias, M. F., & Lopez Pombo, C. G. (2003, May). Time is on my side. In R. Berghammer & B. Möller (Eds.), *Proceedings of the 7th. conference on relational methods in computer science (RelMiCS) - 2nd. international workshop on applications of kleene algebra* (pp. 105–111). Malente, Germany.

Frias, M. F., & Lopez Pombo, C. G. (2006). Interpretability of first-order linear temporal logics in fork algebras. *Journal of Logic and Algebraic Programming*, *66*(2), 161–184.

Frias, M. F., & Orlowska, E. (1997). A proof system for fork algebras and its applications to reasoning in logics based on intuitionism. In J. P. van Bendegem (Ed.), *Logique et analyse* (Vol. 38, pp. 239–284). New Goff.

Frias, M. F., & Orlowska, E. (1998). Equational reasoning in non-classical logics. *Journal of Applied Non-classical Logics*, *8*(1–2), 27–66.

Goguen, J. A., & Burstall, R. M. (1984). Introducing institutions. In E. M. Clarke & D. Kozen (Eds.), *Proceedings of the carnegie mellon workshop on logic of programs* (Vol. 184, pp. 221–256). Springer-Verlag.

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

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

Gyuris, V. (1997, November). A short proof of representability of fork algebra. *Theoretical Computer Science*, *188*(1–2), 211–220.

Haeberer, A. M., & Veloso, P. A. (1991). Partial relations for program derivation: adequacy, inevitability and expressiveness. In *Proceedings of IFIP TC2 working conference on constructing programs from specifications* (pp. 310–352). North Holland.

Harel, D. (2001). Dynamic logic. In D. Gabbay & F. Guenthner (Eds.), *Handbook of philosophical logic* (second ed., Vol. 2, pp. 135–165). Kluwer Academic Publishers.

Harel, D., Kozen, D., & Tiuryn, J. (2000). *Dynamic logic.* Cambridge, MA, USA: MIT Press.

Kowalski, T. (2002). Pdl has interpolation. *Journal of Symbolic Logic*, *67*(3), 933–946.

Kowalski, T. (2004). Retraction note for "pdl has interpolation". *Journal of Symbolic Logic*, *69*(3), 935.

Lopez Pombo, C. G. (2007). *Fork algebras as a tool for reasoning across heterogeneous specifications* (Unpublished doctoral dissertation). Departamento de Computación, Facultad de Ciencias Exactas y Naturales, Universidad de Buenos Aires. (Promotor: Marcelo F. Frias)

Lopez Pombo, C. G., & Frias, M. F. (2006, July, 5–8). Fork algebras as a sufficiently rich universal institution. In M. Johnson & V. Vene (Eds.), *Proceedings of the 11th. international conference on algebraic methodology and software technology, AMAST 2006* (Vol. 4019, pp. 235–247). Kuressaare, Estonia: Springer-Verlag.

Maksimova, L. (1990). Temporal logics with "the next" operator do not have interpolation. *Sibirskii Matematicheskii Zhurnal*, *32*(6), 109–113.

Manna, Z., & Pnueli, A. (1995). *Temporal verification of reactive systems.* New York, NY, USA: Springer-Verlag.

Martini, A., & Wolter, U. (1997, June). A systematic study of mappings between institutions. In F. Parisi-Presicce (Ed.), *Proceedings of the 12th. international workshop on recent trends in algebraic development techniques WADT 1997* (Vol. 1376, pp. 300–315). Tarquinia, Italy: Springer-Verlag.

Martini, A., & Wolter, U. (1998, January). A single perspective on arrows between institutions. In A. M. Haeberer (Ed.), *Proceedings of the 7th. international conference on algebraic methodology and software technology – AMAST 1998* (Vol. 1548, pp. 486–501). Amazonia, Brasil: Springer-Verlag.

McLane, S. (1971). *Categories for working mathematician.* Berlin, Germany: Springer-Verlag.

Meseguer, J. (1989). General logics. In H.-D. Ebbinghaus, J. Fernandez-Prida, M. Garrido, D. Lascar, & M. R. Artalejo (Eds.), *Proceedings of the logic colloquium '87* (Vol. 129, pp. 275–329). Granada, Spain: North Holland.

Mossakowski, T., Maeder, C., & Luttich, K. (2007, April). The heterogeneous tool set, Hets. In O. Grumberg & M. Huth (Eds.), *Proceedings of the 13th. international conference on tools and algorithms for the construction and analysis of systems (TACAS 2007)* (Vol. 4424, pp. 519–522). Braga, Portugal: Springer-Verlag.

Mossakowski, T., & Tarlecki, A. (2009, June). Heterogeneous logical environments for distributed specifications. In A. Corradini & U. Montanari (Eds.), *Proceedings of 19th international workshop in algebraic development techniques* (Vol. 5486, p. 266-289). Pisa, Italy: Springer-Verlag.

Mossakowski, T., & Tarlecki, A. (2014). A relatively complete calculus for structured heterogeneous specifications. In A. Muscholl (Ed.), *Proceedings of 17th international conference on foundations of software science and computation structures (fossacs 2014), held as part of the european joint conferences on theory and practice of software* (Vol. 8412, pp. 441–456). Springer-Verlag.

Orlowska, E., & Golińska-Pilarek, J. (2011). *Dual tableaux: Foundations, methodology, case studies* (Vol. 33; R. Wójcicki, Ed.). Springer-Verlag.

Parisi-Presicce, F. (Ed.). (1997, June). *12th. international workshop on recent trends in algebraic development techniques (WADT'97)* (Vol. 1376). Tarquinia, Italy: Springer-Verlag.

Parnas, D. L. (1972). On the criteria to be used in decomposing systems into modules. *Communications of the ACM*, *15*(12), 1053–1058. (See also (Parnas, 2002))

Parnas, D. L. (1978, May). Designing software for ease of extension and contraction. In M. V. Wilkes, L. Belady, J. Su, H. Hayman, & P. Enslow (Eds.), *Proceedings of the 3rd. international conference on software engineering* (pp. 264–277). Atlanta, Georgia, USA: IEEE Computer Society. (See also (Parnas, 1979))

Parnas, D. L. (1979). Designing software for ease of extension and contraction. *IEEE Transactions on Software Engineering*, *5*(2), 128–138. (See also (Parnas, 1978))

Parnas, D. L. (2002). On the criteria to be used in decomposing systems into modules. In M. Broy & E. Denert (Eds.), *Software pioneers: contributions to software engineering* (pp. 411–427). New York: Springer-Verlag. (See also (Parnas, 1972))

Pnueli, A. (1977). The temporal logic of programs. In *Proceedings of 18th. annual ieee symposium on foundations of computer science* (pp. 46–57). Los Alamitos, CA, USA: IEEE Computer Society.

Pnueli, A. (1981). The temporal semantics of concurrent programs. *Theoretical Computer Science*, *13*(1), 45–60.

Reynolds, M. (2001). An axiomatization of full computational tree logic. *Journal of Symbolic Logic*, *66*(3), 1011–1057.

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

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

Sannella, D., & Tarlecki, A. (2012). *Foundations of algebraic specification and formal software development.* Springer-Verlag.

Sannella, D., & Tarlecki, A. (2014). Property-oriented semantics of structured specifications. *Mathematical Structures in Computer Science*, *24*(2).

Tarlecki, A. (1986). Bits and pieces of the theory of institutions. In D. H. Pitt, S. Abramsky, A. Poigné, & D. E. Rydeheard (Eds.), *Proceedings of the category theory and computer programming, tutorial and workshop* (Vol. 240, pp. 334–363). Springer-Verlag.

Tarlecki, A. (1996). Moving between logical systems. In M. Haveraaen, O. Owe, & O.-J. Dahl (Eds.), *Selected papers from the 11th workshop on specification of abstract data types joint with the 8th compass workshop on recent trends in data type specification* (Vol. 1130, pp. 478–502). Springer-Verlag.

Tarlecki, A. (2000). Towards heterogeneous specifications. In D. Gabbay & M. de Rijke (Eds.), *Frontiers of combining systems* (Vol. 2, pp. 337–360). Research Studies Press.

Tarlecki, A. (2003, August). Abstract specification theory: an overview. In M. Broy & M. Pizka (Eds.), *Proceedings of the nato advanced study institute on models, algebras and logic of engineering software* (pp. 43–79). Marktoberdorf, Germany: IOS Press.

Wirsing, M. (1991, July–August). Structured specifications: Syntax, semantics, and proof calculus. In F. L. Bauer, W. Brauer, & H. Schwichtenberg (Eds.), *Proceedings of the nato advanced study institute on logic and algebra of specifications* (pp. 411–442). Marktoberdorf, Germany: IOS Press.