Under consideration for publication in Math. Struct. in Comp. Science

Encoding Hybridised Institutions into First Order Logic

Răzvan Diaconescu
razvan.diaconescu@imar.ro
Simion Stoilow Institute of Mathematics of the Romanian Academy, Romania

Alexandre Madeira
madeira@di.uminho.pt
HASLab INESC TEC, DI Universidade do Minho, Portugal

Received December 2013; Revised May 2014

A 'hybridisation' of a logic, referred to as the *base logic*, consists of developing the characteristic features of hybrid logic on top of the respective base logic, both at the level of syntax (i.e. modalities, nominals, etc.) and of the semantics (i.e. possible worlds). By 'hybridised institutions' we mean the result of this process when logics are treated abstractly as *institutions* (in the sense of the institution theory of Goguen and Burstall). This work develops encodings of hybridised institutions into (many-sorted) first order logic (abbreviated FOL) as a 'hybridisation' process of abstract encodings of institutions into FOL, which may be seen as an abstraction of the well known standard translation of modal logic into first order logic. The concept of encoding employed by our work is that of *comorphism* from institution theory, which is a rather comprehensive concept of encoding as it features encodings both of the syntax and of the semantics of logics/institutions. Moreover we consider the so-called *theoroidal* version of comorphisms that encode signatures to theories, a feature that accommodates a wide range of concrete applications. Our theory is also general enough to accommodate various constraints on the possible worlds semantics as well a wide variety of quantifications. We also provide pragmatic sufficient conditions for the conservativity of the encodings to be preserved through the hybridisation process, which provides the possibility to shift a formal verification process from the hybridised institution to FOL.

1. Introduction

Hybrid logics (0) are a brand of modal logics that provides appropriate syntax for the possible worlds semantics in a simple and very natural way through the so-called *nominals*. This has several advantages from the point of views of logic and formal specification. For example it has been argued (0) that hybrid logics allow for a better more uniform proof theory than non-hybrid modal logics. Also in specifications of dynamic systems the possibility of explicit reference to specific states of the model is a very necessary feature.

Historically, hybrid logic was introduced in (0) and further developed in works such as (0; 0; 0) etc. Moreover recently hybrid logic has been developed (0) at an abstract institution theoretic level. Institution theory (0) is a categorical abstract model theory that arose about three decades ago within specification theory as a response to the explosion in the population of logics in use there, its original aim being to develop as much computing science as possible in a general uniform way independently of particular logical systems. This has now been achieved to an extent even greater than originally thought, as institution theory became the most fundamental mathematical theory underlying algebraic specification theory (in its wider meaning), also being increasingly used in other areas of computer science. Moreover, institution theory constitutes a major trend in the so-called 'universal logic' (in the sense envisaged by Jean-Yves Béziau (0; 0)) which is considered by many a true renaissance of mathematical logic.

The 'hybridisation' development in (0; 0), which extends the previous work on institution-independent possible worlds semantics of (0) to nominals and multi-modalities, abstracts away the details, both at the syntactic and semantic levels, that are independent of the very essence of the hybrid logic idea. This has several benefits. One is a general benefit of institution theoretic developments, namely that the theoretical development is not hindered by logical details that are often irrelevant. Another benefit is the applicability of the results to a wide variety of concrete instances. This hybridisation of institutions can be regarded as a generic and comprehensive (in the sense of addressing both the syntactic and the semantic levels) form of *hierarchical* logic combination, when the essential features of a logic are built on top of another logic. Besides of the work on modalisation of institutions (0; 0; 0) in the logic and specification theory literature there are other examples of such hierarchical logic combination, e.g. the 'temporalisation' method of (0) or the behavioural extension of preordered algebras (0). However while in the former case temporal logic is built on top of an abstract logic, in the latter behavioural logic is built on top of a concrete logic, namely preordered algebra. We should also emphasise that this form of hierarchical logic combination is very different in many ways from fibring (0) (which is the major general theory of logic combination in the mathematical logic literature), but a discussion comparing them is outside the scope of our paper.

Logic translations or encodings have a long tradition (recently discussed for example in (0)). This concept is especially important since in many cases it may provide a very convenient way to establish logical properties, by 'borrowing' or translating them via a respective encoding rather than by establishing them in a direct manner. A rather common target for such translations or encoding is first order logic (abbreviated FOL); this is because FOL is by far the most popular logical system, it is very well studied and understood, it has good semantic and proof theoretic properties (e.g. completeness, interpolation), and consequently it is supported by a wide variety of formal verification tools. The focus of our work is on extending the traditional translation of modal logic to FOL (0) (for the hybrid variant (0)) to encodings of abstract hybridised institutions into FOL. This may also be regarded as 'hybridisations' of encodings into FOL. While the possibility of such generalization of the standard encoding is hardly surprising, to distill a set of general abstract conditions making this generalization not only possible, but also widely applicable, is a highly non-trivial task. As precise mathematical notion for 'encoding', in this paper we employ the so-called 'theoroidal comorphisms' of (0; 0) which are just ordinary comorphisms but mapping signatures to theories.

Concerning practical applications of this work, our hybridisation method provides the foundations for a methodology for the formal specification and verification of reconfigurable systems (0), i.e., systems which behave differently in different modes of operation (often called configurations) and that shift between the modes in response to events. From a *configuration-as-local-models* perspective models of hybridised institutions are suitable structures to model reconfigurable systems. On the one hand, the relational part of the model represents the reconfigurability of the system - each configuration is represented by an world, each event is represented by a modality and each reconfiguration by a transition. On the other hand, the behaviour and the functionality of each particular configuration is modelled by the local model at that particular world. The ability to adopt a suitable (base) logic for the system in hand is a distinctive aspect of this approach. The encoding to FOL provides the foundation for the formal verification side of this methodology.

Contributions and structure of the paper.

The main target of this work are the formal specification experts and theoreticians. At least some familiarity with the spirit of institution theory would be an advantage to the reader.

The paper is structured as follows.

1 The first preliminary section of the paper is devoted to the brief review of institution theoretic notions that are needed by our work.
2 The second preliminary section recalls the process of hybridisation introduced in (0) and further refined in (0).
3 The main technical section of the paper develops the actual encoding of the hybridised institutions to FOL as a general lifting of abstract comorphisms $\mathcal{I} \to FOL^{\mathrm{pres}}$ (with FOL^{pres} denoting the institutions of the theories of $\mathcal{I}$) to comorphisms $\mathcal{HI}^C \to FOL^{\mathrm{pres}}$ (with $\mathcal{HI}^C$ denoting here a hybridisation of $\mathcal{I}$). This idea to 'hybridise comorphisms' has been sketched within a much restricted and rather preliminary form in (0), here we extend this in several directions: constrained models, theoroidal comorphisms (rather than plain comorphisms), and quantified sentences.
4 The next section studies how the conservativity of the base comorphism $\mathcal{I} \to FOL^{\mathrm{pres}}$ may be inherited by the lifting $\mathcal{HI}^C \to FOL^{\mathrm{pres}}$. Conservativity is a property of special importance, since it allows to transfer proof tasks from the source to the target logic: first translate across the comorphism, then perform them in the target logic, and finally return back the results to the source logic.
5 The final technical section develops a small case study that is meant to illustrate both the abstract developments of our work and the methodology for specification and verification of reconfigurable systems that has been mentioned above.

The abstract developments of this paper are illustrated by a series of concrete benchmark examples.

2. Institutions

In this section we present some concepts of institution theory that are needed by our work. Most of them are rather standard and may be found in many places in the literature, other constitute more recent developments, while a few of them (i.e. Dfn. 3.1, 2.6) are introduced here.

Institution theory is a categorical abstract model theory, hence it is heavily based upon category theory, though the level of category theory involved is rather elementary. We assume the reader is familiar with basic notions and standard notations from elementary category theory; e.g., see

(0) for an introduction to this subject. Here we recall very briefly some of them. By way of notation, $|\mathbb{C}|$ denotes the class of objects of a category $\mathbb{C}$, $\mathbb{C}(A, B)$ the set of arrows with domain A and codomain B, and composition is denoted by ";" and in diagrammatic order. The category of sets (as objects) and functions (as arrows) is denoted by $\mathbf{Set}$, and $\mathbf{CAT}$ is the category of all categories.† The opposite of a category $\mathbb{C}$ (obtained by reversing the arrows of $\mathbb{C}$) is denoted $\mathbb{C}^{op}$.

2.1. *Institutions*

Institutions have been defined by Goguen and Burstall in (0), the seminal paper (0) being printed after a delay of many years. Below we recall the concept of institution which formalises the intuitive notion of logical system, including syntax, semantics, and the satisfaction between them.

Definition 2.1 (Institution). An *institution* $\left(\mathrm{Sign}^{\mathcal{I}}, \mathrm{Sen}^{\mathcal{I}}, \mathrm{Mod}^{\mathcal{I}}, (\models^{\mathcal{I}}_{\Sigma})_{\Sigma\in|\mathrm{Sign}^{\mathcal{I}}|}\right)$ consists of
— a category $\mathrm{Sign}^{\mathcal{I}}$ whose objects are called *signatures*,
— a functor $\mathrm{Sen}^{\mathcal{I}} : \mathrm{Sign}^{\mathcal{I}} \to \mathbf{Set}$ giving for each signature a set whose elements are called *sentences* over that signature,
— a functor $\mathrm{Mod}^{\mathcal{I}} : (\mathrm{Sign}^{\mathcal{I}})^{op} \to \mathbf{CAT}$, giving for each signature Σ a category whose objects are called Σ-*models*, and whose arrows are called Σ-*(model) homomorphisms*, and
— a relation $\models^{\mathcal{I}}_{\Sigma} \subseteq |\mathrm{Mod}^{\mathcal{I}}(\Sigma)| \times \mathrm{Sen}^{\mathcal{I}}(\Sigma)$ for each $\Sigma \in |\mathrm{Sign}^{\mathcal{I}}|$, called the *satisfaction relation*,

such that for each morphism $\varphi : \Sigma \to \Sigma' \in \mathrm{Sign}^{\mathcal{I}}$, the satisfaction condition

$$M' \models^{\mathcal{I}}_{\Sigma'} \mathrm{Sen}^{\mathcal{I}}(\varphi)(\rho) \text{ if and only if } \mathrm{Mod}^{\mathcal{I}}(\varphi)(M') \models^{\mathcal{I}}_{\Sigma} \rho \tag{1}$$

holds for each $M' \in |\mathrm{Mod}^{\mathcal{I}}(\Sigma')|$ and $\rho \in \mathrm{Sen}^{\mathcal{I}}(\Sigma)$.

Notation 2.1. In any institution as above we use the following notations:
– for any $\mathcal{M} \subseteq |\mathrm{Mod}(\Sigma)|$, $\mathcal{M}^*$ denotes $\{\rho \in \mathrm{Sen}(\Sigma) \mid M \models_{\Sigma} \rho$ for each $M \in \mathcal{M}\}$.
– for any $E \subseteq \mathrm{Sen}(\Sigma)$, E^* denotes $\{M \in |\mathrm{Mod}(\Sigma)| \mid M \models_{\Sigma} \rho$ for each $\rho \in E\}$.
– for any $E, E' \subseteq \mathrm{Sen}(\Sigma)$, $E \models E'$ denotes $E^* \subseteq E'^*$ and $E \models\mid E'$ denotes $E^* = E'^*$.
– for any $E \subseteq \mathrm{Sen}(\Sigma)$, $\mathrm{Mod}(\Sigma, E)$ is the full subcategory of $\mathrm{Mod}(\Sigma)$ whose objects are in E^*.

The literature (e.g. (0; 0)) shows myriads of logical systems from computing or from mathematical logic captured as institutions. In fact, an informal thesis underlying institution theory is that any 'logic' may be captured by Dfn. 2.1. While this should be taken with a grain of salt, it certainly applies to any logical system based on satisfaction between sentences and models of any kind. Below we recall a few logics captured as institutions that will also be used in examples in our paper.

Example 2.1 (FOL, ALG, EQ, REL and PL). Let FOL be the institution of *first order logic with equality* in its many sorted form.

Its *signatures* are triples (S, F, P) consisting of
— a set of sort symbols S,
— a family $F = \{F_{\underline{\mathrm{ar}}\to s} \mid \underline{\mathrm{ar}} \in S^*, s \in S\}$ of sets of function symbols indexed by arities $\underline{\mathrm{ar}}$ (for the arguments) and sorts s (for the results), and

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

— a family $P = \{P_{\underline{\mathrm{ar}}} \mid \underline{\mathrm{ar}} \in S^*\}$ of sets of relation (predicate) symbols indexed by arities.

Signature morphisms map the three components in a compatible way. This means that a signature morphism $\varphi : (S, F, P) \to (S', F', P')$ consists of
— a function $\varphi^{\mathrm{st}} : S \to S'$,
— a family of functions $\varphi^{\mathrm{op}} = \{\varphi^{\mathrm{op}}_{\underline{\mathrm{ar}}\to s} : F_{\underline{\mathrm{ar}}\to s} \to F'_{\varphi^{\mathrm{st}}(\underline{\mathrm{ar}})\to\varphi^{\mathrm{st}}(s)} \mid \underline{\mathrm{ar}} \in S^*, s \in S\}$, and
— a family of functions $\varphi^{\mathrm{rl}} = \{\varphi^{\mathrm{rl}}_{\underline{\mathrm{ar}}\to s} : P_{\underline{\mathrm{ar}}} \to P'_{\varphi^{\mathrm{st}}(w)} \mid w \in S^*, s \in S\}$.

Models M for a signature (S, F, P) are first order structures interpreting each sort symbol s as a non-empty set M_s, each function symbol σ as a function M_σ from the product of the interpretations of the argument sorts to the interpretation of the result sort, and each relation symbol π as a subset M_π of the product of the interpretations of the argument sorts. By $|M|$ we denote $\{M_s \mid s \in S\}$ and we call it the *universe of* M or the *carrier set(s) of* M. A model homomorphism $h : M \to M'$ is an indexed family of functions $\{h_s : M_s \to M'_s \mid s \in S\}$ such that

— h is an (S, F)-algebra homomorphism $M \to M'$, i.e., $h_s(M_\sigma(m)) = M'_\sigma(h_{\underline{\mathrm{ar}}}(m))$ for each $\sigma \in F_{\underline{\mathrm{ar}}\to s}$ and each $m \in M_{\underline{\mathrm{ar}}}$, and
— $h_{\underline{\mathrm{ar}}}(m) \in M'_\pi$ if $m \in M_\pi$ (i.e. $h_{\underline{\mathrm{ar}}}(M_\pi) \subseteq M'_\pi$) for each relation $\pi \in P_{\underline{\mathrm{ar}}}$ and each $m \in M_{\underline{\mathrm{ar}}}$.

where $h_{\underline{\mathrm{ar}}} : M_{\underline{\mathrm{ar}}} \to M'_{\underline{\mathrm{ar}}}$ is the canonical component-wise extension of h, i.e. $h_{\underline{\mathrm{ar}}}(m_1, \dots, m_n) = (h_{s_1}(m_1), \dots, h_{s_n}(m_n))$ for $\underline{\mathrm{ar}} = s_1 \dots s_n$ and $m_i \in M_{s_i}$ for $1 \le i \le n$. A model homomorphism is *closed* when $M_\pi = h^{-1}_{\underline{\mathrm{ar}}}(M'_\pi)$ for each relation symbol $\pi \in P_{\underline{\mathrm{ar}}}$.

For each signature morphism φ, the *reduct* $M'\!\upharpoonright_\varphi$ of a model M' is defined by $(M'\!\upharpoonright_\varphi)_x = M'_{\varphi(x)}$ for each sort, function, or relation symbol x from the domain signature of φ.

Sentences are the usual first order sentences built from equational and relational atoms by iterative application of Boolean connectives and quantifiers. Sentence translations along signature morphisms just rename the sorts, function, and relation symbols according to the respective signature morphisms. They can be formally defined by induction on the structure of the sentences. While the induction step is straightforward for the case of the Boolean connectives it needs a bit of attention for the case of the quantifiers. For any signature morphism $\varphi : (S, F, P) \to (S', F', P')$,

$$\mathrm{Sen}^{FOL}(\varphi)((\forall X)\rho) = (\forall X^\varphi)\mathrm{Sen}^{FOL}(\varphi')(\rho)$$

for each finite *block* X of *variables for* (S, F, P). The variables need to be disjoint from the constants of the signature, also we have to ensure that Sen^{FOL} thus defined is functorial indeed and that there is no overloading of variables (which in certain situations would cause a failure of the Satisfaction Condition). These may be formally achieved by considering that a variable for (S, F, P) is a triple of the form $(x, s, (S, F, P))$ where x is the *name of the variable* and $s \in S$ is the *sort of the variable* and that two different variables in X have different names. We often abbreviate variables $(x, s, (S, F, P))$ by their name x or by their name and sort qualification like $(x : s)$. Then we let $(S, F+X, P)$ be the extension of (S, F, P) such that $(F+X)_{\underline{\mathrm{ar}}\to s} = F_{\underline{\mathrm{ar}}\to s}$ when $\underline{\mathrm{ar}}$ is non-empty and $(F + X)_{\to s} = F_{\to s} \cup \{(x, s, (S, F, P)) \mid (x, s, (S, F, P)) \in X\}$ and we let $\varphi' : (S, F + X, P) \to (S', F' + X^\varphi, P')$ be the canonical extension of φ that maps each variable $(x, s, (S, F, P))$ to $(x, \varphi(s), (S', F', P'))$.

As a matter of notation, instead of $(S, F + X, P)$ as above we may also write $(S, F, P) + X$

and when X is a singleton, i.e. $X = \{x\}$, we may simply write x instead of X. We may also extend these conventions to other institutions.

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

— $M \models_{(S,F,P)} t = t'$ when $M_t = M_{t'}$, where M_t denotes the interpretation of the (S,F)-term t in M defined recursively by $M_{\sigma(t_1,\ldots,t_n)} = M_\sigma(M_{t_1},\ldots,M_{t_n})$.

— $M \models_{(S,F,P)} \pi(t_1,\ldots,t_n)$ when $(M_{t_1},\ldots,M_{t_n}) \in M_\pi$, for each relational atom $\pi(t_1,\ldots,t_n)$.

— $M \models_{(S,F,P)} \rho_1 \wedge \rho_2$ when $M \models_{(S,F,P)} \rho_1$ and $M \models_{(S,F,P)} \rho_2$, and similarly for the other Boolean connectives $\vee$, $\Rightarrow$, $\neg$, etc.

— $M \models_{(S,F,P)} (\forall X)\rho$ when $M' \models_{(S,F+X,P)} \rho$ for any $(S, F+X, P)$-expansion M' of M, and similarly for $\exists$.

The institution ALG is obtained by FOL by discarding the relational symbols and the corresponding interpretations in models. The institution EQ is defined as the sub-institution of ALG where the sentences are just universally quantified equations $(\forall X)\, t = t'$. The institution REL is the sub-institution of single-sorted first-order logic with signatures having only constants and relational symbols.

The institution PL (of propositional logic) is the fragment of FOL determined by signatures with empty sets of sort symbols.

Example 2.2 (*PA*). Here we consider the institution PA of partial algebra as employed by the specification language CASL (0).

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

A partial algebra is just like an ordinary algebra (i.e. a FOL model without relations) but interpreting the function symbols of PF as partial rather than total functions. For any $\sigma \in PF_{\underline{\mathrm{ar}}\to s}$ we denote $\mathsf{dom}(A_\sigma) = \{a \in A_{\underline{\mathrm{ar}}} \mid A_\sigma(a) \text{ defined}\}$. A *partial algebra homomorphism* $h\colon\ A \to B$ is a family of (total) functions $\{h_s\colon\ A_s \to B_s \mid s \in S\}$ indexed by the set of sorts S of the signature such that $h_s(A_\sigma(a)) = B_\sigma(h_{\underline{\mathrm{ar}}}(a))$ for each function symbol $\sigma \in TF_{\underline{\mathrm{ar}}\to s} \cup PF_{\underline{\mathrm{ar}}\to s}$ and each string of arguments $a \in A_{\underline{\mathrm{ar}}}$ for which $A_\sigma(a)$ is defined.

The sentences have three kinds of atoms: *definedness* $\mathsf{df}(t)$, *strong* equality $t = t'$, and *existence* equality $t \stackrel{e}{=} t'$. The definedness $\mathsf{df}(t)$ of a term t holds in a partial algebra A when the interpretation A_t of t is defined. The strong equality $t \stackrel{s}{=} t'$ holds when both terms are undefined or both of them are defined and are equal. The existence equality $t \stackrel{e}{=} t'$ holds when both terms are defined and are equal.[‡] The sentences are formed from these atoms by Boolean connectives and quantifications over total variables (i.e variables that are always defined).

Recall from (0; 0):

Definition 2.2 (Internal logic). An institution $\mathcal{I}$ has *(semantic) conjunctions* when for each signature Σ and any Σ-sentences e_1 and e_2 there exists a Σ-sentence e such that $e^* = e_1^* \cap e_2^*$. Usually e is denoted by $e_1 \wedge e_2$.

[‡] Notice that $\mathsf{df}(t)$ is equivalent to $t \stackrel{e}{=} t$ and that $t \stackrel{s}{=} t'$ is equivalent to $(t \stackrel{e}{=} t') \vee (\neg\mathsf{df}(t) \wedge \neg\mathsf{df}(t'))$.

$\mathcal{I}$ has *(semantic) implications* when for each e_1 and e_2 as above there exists e such that $e^* = (\mathrm{Mod}(\Sigma) - e_1^*) \cup e_2^*$. Usually e is denoted $e_1 \Rightarrow e_2$.

$\mathcal{I}$ has *(semantic) existential $\mathcal{D}$-quantifications* for a class $\mathcal{D}$ of signature morphisms when for each $\chi : \Sigma \to \Sigma' \in \mathcal{D}$ when for each Σ'-sentence e' there exists a Σ-sentence e such that $e^* = \mathrm{Mod}(\chi)(e'^*)$. Usually e is denoted $(\exists\chi)e'$.

In the same style we may extend this list also to other semantic Boolean connectives disjunction ($\vee$), negation ($\neg$), equivalence ($\Leftrightarrow$) and to semantic universal quantifications ($(\forall\chi)e'$).

2.2. *Amalgamation and quantification spaces*

We recall the notions of *amalgamation* and *quantification space* that are crucial for what follows. The former is intensely used in institution theory, whereas the latter was introduced rather recently in (0). The respective definitions below represent a slight adaptation of the definitions from the literature to the needs of this paper; in this form Dfn. 2.3 and 2.5 have already appeared in (0; 0). Dfn. 2.4 was introduced in (0).

Definition 2.3 (Amalgamation property). A commuting square of functors

$$\begin{tikzcd} A & A_1 \arrow[l, "F_1"'] \\ A_2 \arrow[u, "F_2"] & A' \arrow[l, "G_2"] \arrow[u, "G_1"'] \end{tikzcd} \tag{2}$$

is a *weak amalgamation square* if and only if for each $M_1 \in |A_1|$ and $M_2 \in |A_2|$ such that $F_1(M_1) = F_2(M_2)$, there exists a $M' \in |A'|$ such that $G_1(M') = M_1$ and $G_2(M') = M_2$. When M' is required to be unique, the square is called *amalgamation square*. The object M' is called an *amalgamation of M_1 and M_2* and when it is unique it is denoted by $M_1 \otimes_{F_1,F_2} M_2$.

For any functor $\mathrm{Mod} : \mathrm{Sign}^{op} \to \mathbf{CAT}$ a commuting square of signature morphisms

$$\begin{tikzcd} \Sigma \arrow[r, "\varphi_1"] \arrow[d, "\varphi_2"'] & \Sigma_1 \arrow[d, "\theta_1"] \\ \Sigma_2 \arrow[r, "\theta_2"'] & \Sigma' \end{tikzcd} \tag{3}$$

is a *(weak) amalgamation square for* Mod when

$$\begin{tikzcd} \mathrm{Mod}(\Sigma) & \mathrm{Mod}(\Sigma_1) \arrow[l, "\mathrm{Mod}(\varphi_1)"'] \\ \mathrm{Mod}(\Sigma_2) \arrow[u, "\mathrm{Mod}(\varphi_2)"] & \mathrm{Mod}(\Sigma') \arrow[l, "\mathrm{Mod}(\theta_2)"] \arrow[u, "\mathrm{Mod}(\theta_1)"'] \end{tikzcd} \tag{4}$$

is a (weak) amalgamation square.

We say that an institution $\mathcal{I}$ has the (weak) amalgamation property when each pushout square of signature morphisms is a (weak) amalgamation square for the model functor $\mathrm{Mod}^{\mathcal{I}}$.

Most of the institutions formalising conventional or non-conventional logics have the amalgamation property (0; 0). These include our examples FOL, ALG, PL, REL, PA. Our concept of model amalgamation should not be confused with single signature and much harder one from conventional model theory (e.g. (0)) which refers to the amalgamation of elementary extensions.

The concept introduced by Dfn. 2.4 below will be used within the context of our abstract approach to constraining Kripke models.

Definition 2.4. A sub-functor $\mathrm{Mod}' \subseteq \mathrm{Mod} : \ \mathrm{Sign}^{op} \to \mathbf{CAT}$ *reflects (weak) amalgamation* for a class of pushout squares in Sign when each pushout square of that class that is a (weak) amalgamation square for Mod is a (weak) amalgamation square for Mod' too.

Definition 2.5 (Quantification space). For any category Sign a subclass of arrows $\mathcal{D} \subseteq \mathrm{Sign}$ is called a *quantification space* if, for any $(\chi : \ \Sigma \to \Sigma') \in \mathcal{D}$ and $\varphi : \ \Sigma \to \Sigma_1$, there is a designated pushout

$$\begin{tikzcd} \Sigma \arrow[r, "\varphi"] \arrow[d, "\chi"'] & \Sigma_1 \arrow[d, "\chi(\varphi)"] \\ \Sigma' \arrow[r, "\varphi[\chi]"'] & \Sigma'_1 \end{tikzcd}$$

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

$$\begin{tikzcd} \Sigma \arrow[r, "\varphi"] \arrow[d, "\chi"'] & \Sigma_1 \arrow[r, "\theta"] \arrow[d, "\chi(\varphi)"] & \Sigma_2 \arrow[d, "\chi(\varphi)(\theta)"] \\ \Sigma' \arrow[r, "\varphi[\chi]"'] & \Sigma'_1 \arrow[r, "\theta[\chi(\varphi)]"'] & \Sigma'_2 \end{tikzcd}$$

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

We say that a quantification space $\mathcal{D}$ for Sign is *adequate for* a functor $\mathrm{Mod} : \ \mathrm{Sign}^{op} \to \mathbf{CAT}$ when the designated pushouts mentioned above are weak amalgamation squares for Mod.

Our use of designated pushouts as in Dfn. 2.5 is required by the fact that quantified sentences ought to have a unique translation along a given signature morphism. The coherence property of the composition is required by the functoriality of the translations.

Example 2.3 ($\mathcal{D}^{FOL}$). Within the context of Ex. 2.1 above, the signature extensions $\chi : \ (S, F, P) \hookrightarrow (S, F + X, P)$, where X is a finite block of variables for (S, F, P) constitute a quantification space for Sign^{FOL} that is adequate for Mod^{FOL}. Let us denote it by $\mathcal{D}^{FOL}$. Given signature morphism $\varphi : \ (S, F, P) \to (S_1, F_1, P_1)$, then

- $\chi(\varphi) : \ (S_1, F_1, P_1) \hookrightarrow (S_1, F_1 + X^\varphi, P_1)$ where X^φ as defined in Ex. 2.1, and
- $\varphi[\chi]$ is the canonical extension of φ that maps each $(x, s, (S, F, P))$ to $(x, \varphi^{\mathrm{st}}(s), (S_1, F_1, P_1))$ (it corresponds to φ' of Ex. 2.1).

It is easy to note that these define pushout squares fulfilling the properties of Dfn. 2.5. The adequacy for Mod^{FOL} follows from the fact that Mod^{FOL} preserves all finite limits (see (0)).

Other quantification spaces for Sign^{FOL} that are also adequate for Mod^{FOL} may be obtained as follows:

1 In the example above we consider *infinite* blocks of variables instead of finite ones.
2 We consider blocks of second order variables of the form $(x, (w, s), (S, F, P))$ (function variables) or of the form $(x, w, (S, F, P))$ (relation variables) where $\underline{\text{ar}} \in S^*$ and $s \in S$. Then to any block X of second order variables it corresponds a signature extension $\chi : (S, F, P) \to (S, F + X^{\text{op}}, P + X^{\text{rl}})$ where X is split as $X^{\text{op}} \cup X^{\text{rl}}$ with X^{op} being the function variables and X^{rl} the relation variables, and where $F + X^{\text{op}}$ and $P + X^{\text{rl}}$ extend in the obvious way the definition of $F + X$ from Ex. 2.1.

Note that these definitions may also apply to REL and ALG. Similar definitions may also be developed in PA.

The property expressed by Dfn. 2.6 below will be used as a condition underlying the main result of this work.

Definition 2.6. For any functors $\text{Mod}_1, \text{Mod}_2 : \text{Sign} \to \mathbf{CAT}^{\text{op}}$ and any natural transformation $\beta : \text{Mod}_2 \to \text{Mod}_1$ we say that $(\chi : \Sigma \to \Sigma') \in \text{Sign}$ is *adequate for* β if and only if the following square is weak amalgamation square:

$$\begin{tikzcd}
\text{Mod}_1(\Sigma) & \text{Mod}_2(\Sigma) \arrow[l, "\beta_\Sigma"'] \\
\text{Mod}_1(\Sigma') \arrow[u, "\text{Mod}_1(\chi)"] & \text{Mod}_2(\Sigma') \arrow[l, "\beta_{\Sigma'}"] \arrow[u, "\text{Mod}_2(\chi)"']
\end{tikzcd}$$

When resorting to a Grothendieck construction it is possible to regard the adequacy for β of Dfn. 2.6 as a special case of the adequacy property of Dfn. 2.5; let us skip these rather technical details here.

2.3. *Comorphisms*

In the literature there are several concepts of structure preserving mappings between institutions. The original one, introduced by (0), is adequate for expressing a 'forgetful' operation from a 'more complex' institution to a structurally 'simpler' one. However, the institution mapping which is appropriate for our task here is that of institution comorphisms (0), previously know as 'plain map' in (0) or 'representation' in (0; 0). Institution comorphisms realize the intuition of 'embedding' a 'simpler' institution into a 'more complex' one, which is dual to the intuition realised by the institution morphisms.

Definition 2.7 (Comorphisms). An *institution comorphism* $(\Phi, \alpha, \beta) : \mathcal{I} \to \mathcal{I}'$ consists of

1 a functor $\Phi : \text{Sign} \to \text{Sign}'$,
2 a natural transformation $\alpha : \text{Sen} \Rightarrow \Phi; \text{Sen}'$, and
3 a natural transformation $\beta : \Phi^{\text{op}}; \text{Mod}' \Rightarrow \text{Mod}$

such that the following *satisfaction condition* holds

$$M' \models'_{\Phi(\Sigma)} \alpha_\Sigma(e) \text{ iff } \beta_\Sigma(M') \models_\Sigma e$$

for each signature $\Sigma \in |\text{Sign}|$, for each $\Phi(\Sigma)$-model M', and each Σ-sentence e.

The comorphism is *conservative* whenever, for each Σ-model M in $\mathcal{I}$, there exists a $\Phi(\Sigma)$-model M' in $\mathcal{I}'$ such that $M = \beta_\Sigma(M')$.

The following is a consequence of conservativity, with the important proof theoretic implication that we may prove things in the source institution by using the proof system of the target institution in a sound and complete way.

Fact 2.1. Given a conservative comorphism, for any set $\Gamma \subseteq \mathrm{Sen}(\Sigma)$ and sentence $\rho \in \mathrm{Sen}(\Sigma)$,

$$\Gamma \models_\Sigma \rho \text{ if and only if } \alpha_\Sigma(\Gamma) \models'_{\Phi(\Sigma)} \alpha_\Sigma(\rho).$$

2.4. *Presentations*

Although comorphisms generally express an embedding relationship between institutions, they can also be used for ‘encoding’ a ‘more complex’ institution $\mathcal{I}$ into a ‘simpler’ one $\mathcal{I}'$. The latter are especially useful for the borrowing methods; some references are (0; 0; 0; 0). In such encodings the structural complexity cost is shifted to the mapping Φ on the signatures, thus Φ maps signatures of $\mathcal{I}$ to theories of $\mathcal{I}'$ rather than signatures. In the literature these are sometimes (0; 0) called ‘theoroidal’ comorphisms. In the following we give a general construction which explains this concept as ordinary comorphism.

Definition 2.8 (Presentations). In any institution $\mathcal{I}$, a *presentation* is a pair (Σ, E) consisting of an $\mathcal{I}$-signature Σ and a set E of Σ-sentences. A *presentation morphism* $\varphi : (\Sigma, E) \to (\Sigma', E')$ is a signature morphism $\varphi : \Sigma \to \Sigma'$ such that $\varphi(E) \subseteq E'$.

Fact 2.2. Presentation morphisms are closed under the composition given by the composition of the signature morphisms.

This fact opens the door for the general construction given by the following definition.

Definition 2.9 (The institution of presentations). Let $\mathcal{I} = (\mathrm{Sign}, \mathrm{Sen}, \mathrm{Mod}, \models)$ be any institution. The institution of the presentations of $\mathcal{I}$, denoted by

$$\mathcal{I}^{\mathrm{pres}} = (\mathrm{Sign}^{\mathrm{pres}}, \mathrm{Sen}^{\mathrm{pres}}, \mathrm{Mod}^{\mathrm{pres}}, \models^{\mathrm{pres}})$$

is defined by

- $\mathrm{Sign}^{\mathrm{pres}}$ is the category Pres of presentations of $\mathcal{I}$,
- $\mathrm{Sen}^{\mathrm{pres}}(\Sigma, E) = \mathrm{Sen}(\Sigma)$,
- $\mathrm{Mod}^{\mathrm{pres}}(\Sigma, E)$ is the full subcategory of $\mathrm{Mod}(\Sigma)$ of those models which satisfy E, and
- for each (Σ, E)-model M and Σ-sentence e, $M \models^{\mathrm{pres}}_{(\Sigma,E)} e$ if and only if $M \models_\Sigma e$.

Fact 2.3. For any institution $\mathcal{I}$, $\mathcal{I}^{\mathrm{pres}}$ is indeed an institution.

Note that our definition of presentation morphism is slightly more restrictive than what is commonly defined in the literature (e.g. (0)) were the condition $\varphi(E) \subseteq E'$ is relaxed to $E' \models \varphi(E)$. Under that relaxation the (simple) theoroidal comorphisms of (0; 0) arise precisely as ordinary comorphisms $\mathcal{I} \to \mathcal{I}'^{\mathrm{pres}}$. The reason for our restriction is that in this way, later in the paper, we will avoid some technical difficulties, in the same time not sacrificing the applications since almost always the concrete institution encodings in form of theoroidal comorphisms fulfil rather naturally our restricted definition. Moreover if we consider infinite sets of sentences for the presentations we have the possibility to consider E and E' to be closed under semantical consequence, and in such case both versions are equivalent. However of course this may sacrifice the finitary character of the encodings. The literature abounds of examples of institution encodings that are presented as comorphisms $\mathcal{I} \to \mathcal{I}'^{\mathrm{pres}}$; many of them may be found in (0).

Example 2.4. Let us briefly recall the emblematic case of the encodings of PA into FOL. There are several such encodings as follows (details may be found in the literature, e.g. (0; 0; 0; 0)):

1 Perhaps the best known one encodes partial operations as total operations by adding for each sort an unary relation symbol standing for the defined values, the target presentations consisting of Horn sentences. This comorphism has the benefit of transfer of initial semantics.
2 The comorphism used in (0; 0) encodes partial operations as (functional) relations and while the target presentations also consist of Horn sentences, unlike in the previous case the translations of the sentences (α) is rather complex which meaning that Horn sentences may by translated to non-Horn sentences. However this comorphism has the benefit that the sentence translations are surjective, which allows the transfer of interpolation properties.
3 The encoding recently discovered in (0) adds a quasi-Boolean sort, like the first one preserves the Horn presentations, unlike the second one it is not surjective on the sentence translations but has the benefit of not involving any relation symbols.

3. Hybridised Institutions

In this section we present the institution-independent construction of hybrid logics that has already been introduced in (0; 0) as an extension of the previous work (0). Let us consider an institution $\mathcal{I} = (\mathrm{Sign}^{\mathcal{I}}, \mathrm{Sen}^{\mathcal{I}}, \mathrm{Mod}^{\mathcal{I}}, (\models^{\mathcal{I}}_{\Sigma})_{\Sigma \in |\mathrm{Sign}^{\mathcal{I}}|})$ with a designated quantification space $\mathcal{D}^{\mathcal{I}} \subseteq \mathrm{Sign}^{\mathcal{I}}$. This will be referred to as the *base institution*. Below we recall the method to enrich $\mathcal{I}$ with modalities and nominals, defining a suitable semantics for the enrichment. Moreover, it is shown that the outcome still defines a class of institutions, the so-called *hybridisations of* $\mathcal{I}$.

The category of $\mathcal{HI}$-signatures:

The category of *$\mathcal{I}$-hybrid signatures*, denoted by $\mathrm{Sign}^{\mathcal{HI}}$, is defined as the following direct (cartesian) product of categories:

$$\mathrm{Sign}^{\mathcal{HI}} = \mathrm{Sign}^{\mathcal{I}} \times \mathrm{Sign}^{REL}.$$

The REL-signatures are denoted by (Nom, Λ), where Nom is a set of constants called *nominals* and Λ is a set of relational symbols called *modalities*; Λ_n stands for the set of modalities of arity n. General category theory entails:

Proposition 3.1. The projection $\mathrm{Sign}^{\mathcal{HI}} \to \mathrm{Sign}^{\mathcal{I}}$ lifts small co-limits.

The existence of co-limits of signatures is one of the properties of institutions of key practical relevance for specification in-the-large (see (0)).

Corollary 3.1. $\mathrm{Sign}^{\mathcal{HI}}$ has all small co-limits.

$\mathcal{HI}$-sentences:

Let us fix a quantification space $\mathcal{D}^{\mathcal{HI}}$ for $\mathrm{Sign}^{\mathcal{HI}}$ such that for each $\chi \in \mathcal{D}^{\mathcal{HI}}$ its projection χ_{Sig} to $\mathrm{Sign}^{\mathcal{I}}$ belongs to $\mathcal{D}^{\mathcal{I}}$. The quantification space $\mathcal{D}^{\mathcal{HI}}$ is a parameter of the hybridisation process. Whenever $\mathcal{D}^{\mathcal{HI}}$ consists of identities we say the hybridisation is *quantifier-free*. Note that a quantifier-free hybridisation does not necessarily mean the absence of 'local' quantification, i.e. placed at the level of base institution $\mathcal{I}$.

Let $\Delta = (\Sigma, \mathrm{Nom}, \Lambda)$. The set of sentences $\mathrm{Sen}^{\mathcal{HI}}(\Delta)$ is the least set such that
— $\mathrm{Nom} \subseteq \mathrm{Sen}^{\mathcal{HI}}(\Delta)$;
— $\mathrm{Sen}^{\mathcal{I}}(\Sigma) \subseteq \mathrm{Sen}^{\mathcal{HI}}(\Delta)$;
— $\rho \star \rho' \in \mathrm{Sen}^{\mathcal{HI}}(\Delta)$ for any $\rho, \rho' \in \mathrm{Sen}^{\mathcal{HI}}(\Delta)$ and any $\star \in \{\vee, \wedge, \Rightarrow\}$,
— $\neg\rho \in \mathrm{Sen}^{\mathcal{HI}}(\Delta)$, for any $\rho \in \mathrm{Sen}^{\mathcal{HI}}(\Delta)$,
— $@_i\rho \in \mathrm{Sen}^{\mathcal{HI}}(\Delta)$ for any $\rho \in \mathrm{Sen}^{\mathcal{HI}}(\Delta)$ and $i \in \mathrm{Nom}$;
— $[\lambda](\rho_1, \dots, \rho_n), \langle\lambda\rangle(\rho_1, \dots, \rho_n) \in \mathrm{Sen}^{\mathcal{HI}}(\Delta)$, for any $\lambda \in \Lambda_{n+1}, \rho_i \in \mathrm{Sen}^{\mathcal{HI}}(\Delta)$, $i \in \{1, \dots, n\}$;
— $(\forall\chi)\rho, (\exists\chi)\rho \in \mathrm{Sen}^{\mathcal{HI}}(\Delta)$, for any $\rho \in \mathrm{Sen}^{\mathcal{HI}}(\Delta')$ and $\chi\colon\ \Delta \to \Delta' \in \mathcal{D}^{\mathcal{HI}}$;

When χ is a simple extension with variables we may abbreviatte it in the quantifications by the corresponding variables. For example when χ is an extension of $(\Sigma, \mathrm{Nom}, \Lambda)$ with a nominal variable i, instead of $(\forall\chi)\rho$ we may write $(\forall i)\rho$.

Our set of logical connectors follows mainstream hybrid logic literature (e.g. (0)). However we do not consider here the binder $\downarrow$ since it is known to be logically redundant.

Translations of $\mathcal{HI}$-sentences:

Let $\varphi = (\varphi_{\mathrm{Sig}}, \varphi_{\mathrm{Nom}}, \varphi_{\mathrm{MS}})\colon\ (\Sigma, \mathrm{Nom}, \Lambda) \to (\Sigma', \mathrm{Nom}', \Lambda')$ be a morphims of $\mathcal{HI}$-signatures.

The translation $\mathrm{Sen}^{\mathcal{HI}}(\varphi)$ is defined as follows:
— $\mathrm{Sen}^{\mathcal{HI}}(\varphi)(i) = \varphi_{\mathrm{Nom}}(i)$;
— $\mathrm{Sen}^{\mathcal{HI}}(\varphi)(\rho) = \mathrm{Sen}^{\mathcal{I}}(\varphi_{\mathrm{Sig}})(\rho)$ for any $\rho \in \mathrm{Sen}^{\mathcal{I}}(\Sigma)$;
— $\mathrm{Sen}^{\mathcal{HI}}(\varphi)(\rho \star \rho') = \mathrm{Sen}^{\mathcal{HI}}(\varphi)(\rho) \star \mathrm{Sen}^{\mathcal{HI}}(\varphi)(\rho')$, $\star \in \{\vee, \wedge, \Rightarrow\}$;
— $\mathrm{Sen}^{\mathcal{HI}}(\varphi)(\neg\rho) = \neg\mathrm{Sen}^{\mathcal{HI}}(\varphi)(\rho)$;
— $\mathrm{Sen}^{\mathcal{HI}}(\varphi)(@_i\rho) = @_{\varphi_{\mathrm{Nom}}(i)}\mathrm{Sen}^{\mathcal{HI}}(\rho)$;
— $\mathrm{Sen}^{\mathcal{HI}}(\varphi)([\lambda](\rho_1, \dots, \rho_n)) = [\varphi_{\mathrm{MS}}(\lambda)](\mathrm{Sen}^{\mathcal{HI}}(\rho_1), \dots, \mathrm{Sen}^{\mathcal{HI}}(\rho_n))$;
— $\mathrm{Sen}^{\mathcal{HI}}(\varphi)(\langle\lambda\rangle(\rho_1, \dots, \rho_n)) = \langle\varphi_{\mathrm{MS}}(\lambda)\rangle(\mathrm{Sen}^{\mathcal{HI}}(\rho_1), \dots, \mathrm{Sen}^{\mathcal{HI}}(\rho_n))$;
— $\mathrm{Sen}^{\mathcal{HI}}(\varphi)\big((\forall\chi)\rho\big) = (\forall\chi(\varphi))\mathrm{Sen}^{\mathcal{HI}}(\varphi[\chi])(\rho)$;
— $\mathrm{Sen}^{\mathcal{HI}}(\varphi)\big((\exists\chi)\rho\big) = (\exists\chi(\varphi))\mathrm{Sen}^{\mathcal{HI}}(\varphi[\chi])(\rho)$;

The following result may be obtained by recursion on the structure of the sentences by straightforward calculations (omitted here), the most interesting parts being those corresponding to the quantifiers $\forall$ and $\exists$; in those cases one relies crucially upon the properties expressed in Dfn. 2.5.

Proposition 3.2. $\mathrm{Sen}^{\mathcal{HI}}$ is a functor $\mathrm{Sign}^{\mathcal{HI}} \to \mathbf{Set}$.

$\mathcal{HI}$-models:

The $(\Sigma, \mathrm{Nom}, \Lambda)$-*models* are pairs (M, W) where
— W is a (Nom, Λ)-model in REL;
— M is a function $|W| \to |\mathrm{Mod}^{\mathcal{I}}(\Sigma)|$.

The carrier set $|W|$ forms *the set of the states of* (M, W); $\{W_n \mid n \in \mathrm{Nom}\}$ represents the interpretations of the *nominals* Nom, whereas relations $\{W_\lambda \mid \lambda \in \Lambda_n, n \in \omega\}$ represent the interpretation of the *modalities* Λ. We denote $M(w)$ simply by M_w.

A $(\Sigma, \mathrm{Nom}, \Lambda)$-*model homomorphism* $h : (M, W) \to (M', W')$ consists of a pair aggregating

— a (Nom, Λ)-model homomorphism in REL, $h_{st} : W \to W'$; i.e., a function $h_{st} : |W| \to |W'|$ such that for $i \in \mathrm{Nom}$, $W'_i = h_{st}(W_i)$; and, for any $w_1, \ldots, w_n \in |W|$, $\lambda \in \Lambda_n$, and $(w_1, \ldots, w_n) \in W_\lambda$, $(h_{st}(w_1), \ldots, h_{st}(w_n)) \in W'_\lambda$.

— a natural transformation $h_{mod} : M \Rightarrow M' \circ h_{st}$; note that h_{mod} is a $|W|$-indexed family of Σ-model homomorphisms $h_{mod} = \{(h_{mod})_w : M_w \to M'_{h_{st}(w)} \mid w \in |W|\}$. In the text sometimes we may abbreviate $(h_{mod})_w$ by h_w.

The composition of $\mathcal{HI}$-model homomorphisms is defined canonically as

$$h; h' = (h_{st}; h'_{st}, h_{mod}; (h'_{mod} \circ h_{st})).$$

Fact 3.1. Let Δ be any $\mathcal{HI}$-signature. Then Δ-models together with their homomorphisms constitute a category, denoted $\mathrm{Mod}^{\mathcal{HI}}(\Delta)$.

Reducts of $\mathcal{HI}$-models:

Let $\Delta = (\Sigma, \mathrm{Nom}, \Lambda)$ and $\Delta' = (\Sigma', \mathrm{Nom}', \Lambda')$ be two $\mathcal{HI}$-signatures, $\varphi = (\varphi_{\mathrm{Sig}}, \varphi_{\mathrm{Nom}}, \varphi_{\mathrm{MS}})$ a morphism between Δ and Δ' and (M', W') a Δ'-model. The *reduct* of (M', W') along φ, denoted by $\mathrm{Mod}^{\mathcal{HI}}(\varphi)(M', W')$, is the Δ-model (M, W) such that

— W is the $(\varphi_{\mathrm{Nom}}, \varphi_{\mathrm{MS}})$-reduct of W'; i.e.
  - $|W| = |W'|$;
  - for any $n \in \mathrm{Nom}$, $W_n = W'_{\varphi_{\mathrm{Nom}}(n)}$;
  - for any $\lambda \in \Lambda$, $W_\lambda = W'_{\varphi_{\mathrm{MS}}(\lambda)}$;

  and

— for any $w \in |W|$, $M_w = \mathrm{Mod}^{\mathcal{I}}(\varphi_{\mathrm{Sig}})(M'_w)$.

Theorem 3.1. (0) A pushout square of $\mathcal{HI}$-signature morphisms is a (weak) amalgamation square (for $\mathrm{Mod}^{\mathcal{HI}}$) if the underlying square of signature morphisms in $\mathcal{I}$ is a (weak) amalgamation square.

Corollary 3.2. (0) If $\mathcal{D}^{\mathcal{I}}$ is adequate for $\mathrm{Mod}^{\mathcal{I}}$ then $\mathcal{D}^{\mathcal{HI}}$ is adequate for $\mathrm{Mod}^{\mathcal{HI}}$.

Below we will se that the Satisfaction Condition for hybridised institutions relies upon the adequacy property from the conclusion of Cor. 3.2.

Constrained models:

Often the semantics of modal and hybrid logics may include various constraints on the models. A well known example is the uniform interpretation of the 'rigid' constants across the possible worlds, necessary for the most common form of quantification in first order modal logic. The following definition of (0) captures abstractly the model constraints.

Definition 3.1. A *constrained $\mathcal{HI}$-model functor* is a sub-functor $\mathrm{Mod}^C \subseteq \mathrm{Mod}^{\mathcal{HI}}$ such that it reflects weak amalgamation for the designated pushout squares corresponding to $\mathcal{D}^{\mathcal{I}}$. The models in Mod^C are called *constrained $\mathcal{HI}$-models*.

Informally, the meaning of the reflection condition of Dfn. 3.1 is that in the case of pushout squares of signature morphisms the amalgamation of constrained models yields a constrained model.

The following result, which is an immediate consequence of Cor. 3.2, Dfn. 3.1 and Dfn. 2.4, applies often in concrete situations, including all the examples in our paper.

Corollary 3.3. If $\mathcal{D}^{\mathcal{I}}$ is adequate for $\mathrm{Mod}^{\mathcal{I}}$ then $\mathcal{D}^{\mathcal{HI}}$ is adequate for any constrained $\mathcal{HI}$-model functor Mod^C.

The Satisfaction Relation:

Given a constrained model functor $\mathrm{Mod}^C \subseteq \mathrm{Mod}^{\mathcal{HI}}$, for any $(M, W) \in |\mathrm{Mod}^C(\Sigma, \mathrm{Nom}, \Lambda)|$ and for any $w \in |W|$ we define:

— $(M, W) \models^w i$ iff $W_i = w$; when $i \in \mathrm{Nom}$,
— $(M, W) \models^w \rho$ iff $M_w \models^{\mathcal{I}} \rho$; when $\rho \in \mathrm{Sen}^{\mathcal{I}}(\Sigma)$,
— $(M, W) \models^w \rho \vee \rho'$ iff $(M, W) \models^w \rho$ or $(M, W) \models^w \rho'$,
— $(M, W) \models^w \rho \wedge \rho'$ iff $(M, W) \models^w \rho$ and $(M, W) \models^w \rho'$,
— $(M, W) \models^w \rho \Rightarrow \rho'$ iff $(M, W) \models^w \rho$ implies that $(M, W) \models^w \rho'$,
— $(M, W) \models^w \neg\rho$ iff $(M, W) \not\models^w \rho$,
— $(M, W) \models^w @_j \rho$ iff $(M, W) \models^{W_j} \rho$,
— $(M, W) \models^w [\lambda](\xi_1, \ldots, \xi_n)$ iff for any $(w, w_1, \ldots, w_n) \in W_\lambda$ we have that $(M, W) \models^{w_i} \rho_i$ for some $1 \leq i \leq n$.
— $(M, W) \models^w \langle\lambda\rangle(\xi_1, \ldots, \xi_n)$ iff there exists $(w, w_1, \ldots, w_n) \in W_\lambda$ such that and $(M, W) \models^{w_i} \xi_i$ for any $1 \leq i \leq n$.
— $(M, W) \models^w (\forall\chi)\rho$ iff $(M', W') \models^w \rho$ for any (M', W') such that $\mathrm{Mod}^C(\chi)(M', W') = (M, W)$,
— $(M, W) \models^w (\exists\chi)\rho$ iff $(M', W') \models^w \rho$ for some (M', W') such that $\mathrm{Mod}^C(\chi)(M', W') = (M, W)$, and

We write $(M, W) \models \rho$ iff $(M, W) \models^w \rho$ for any $w \in |W|$.

Note that, as expected, we have the semantical equivalence between the sentences $\langle\lambda\rangle(\rho_1, \ldots, \rho_n)$ and $\neg[\lambda](\neg\rho_1, \ldots, \neg\rho_n)$. It is also interesting to note that if the quantification space allows quantifications with nominal variables, then the binder operator $\downarrow$ that appears in many works on hybrid logic, e.g. (0; 0) etc., is redundant since sentences of the form $(\downarrow i)\rho$ are semantically equivalent to $(\forall i)(i \Rightarrow \rho)$.

Our general semantics of quantifiers covers various concrete first order quantifications from the modal logic literature by letting χ be some concrete finite extensions of signatures with first order variables and by suitable choice of model constraints (Mod^C). For example the standard rigid quantification (e.g. (0)) is covered when the models are constrained such that its possible worlds share the same domain and the same interpretation of a designated set of constants that are marked as 'rigid' and when the first order variables considered are 'rigid'. Without such 'rigid' constraints we get to the situation when variables may be interpreted differently across different worlds, which amounts to the *world-line* semantics of (0). However in the applications the general technical conditions our main result exclude the latter situation.

The Satisfaction Condition:

Theorem 3.2 (Local Satisfaction Condition). (0) Assume $\mathcal{D}^{\mathcal{I}}$ is adequate for $\mathrm{Mod}^{\mathcal{I}}$. Let $\Delta = (\Sigma, \mathrm{Nom}, \Lambda)$ and $\Delta' = (\Sigma', \mathrm{Nom}', \Lambda')$ be two $\mathcal{HI}$-signatures and $\varphi : \Delta \to \Delta'$ a morphism of signatures. Given a constrained model functor $\mathrm{Mod}^C \subseteq \mathrm{Mod}^{\mathcal{HI}}$, for any $\rho \in \mathrm{Sen}^{\mathcal{HI}}(\Delta)$, $(M', W') \in |\mathrm{Mod}^C(\Delta')|$, and $w \in |W'|$

$$\mathrm{Mod}^C(\varphi)(M', W')(= \mathrm{Mod}^{\mathcal{HI}}(\varphi)(M', W')) \models^w \rho \text{ if and only if } (M', W') \models^w \mathrm{Sen}^{\mathcal{HI}}(\varphi)(\rho). \tag{5}$$

Note that in the quantifier-free situation, i.e. when $\mathcal{D}^{\mathcal{HI}}$ is trivial, then $\mathcal{D}^{\mathcal{I}}$ may also be considered trivial and hence the adequacy assumption of Thm. 3.2 holds trivially. Also in this case the constraint functor may be any sub-functor of $\mathrm{Mod}^{\mathcal{HI}}$ since the designated pushout squares corresponding to $\mathcal{D}^{\mathcal{I}}$ are trivial too.

Corollary 3.4 (Global Satisfaction Condition). (0) $(\mathrm{Sign}^{\mathcal{HI}}, \mathrm{Sen}^{\mathcal{HI}}, \mathrm{Mod}^C, \models)$ is an institution.

Let us call the institution $(\mathrm{Sign}^{\mathcal{HI}}, \mathrm{Sen}^{\mathcal{HI}}, \mathrm{Mod}^C, \models)$ *a hybridisation of* $\mathcal{I}$ and let us denote it by $\mathcal{HI}^C$. The hybridisation $(\mathrm{Sign}^{\mathcal{HI}}, \mathrm{Sen}^{\mathcal{HI}}, \mathrm{Mod}^{\mathcal{HI}}, \models)$, that does not constrains models, is denoted $\mathcal{HI}$ and is called the *free hybridisation* of $\mathcal{I}$. Note that in general, because of the quantifiers, the satisfaction relation $\models^{\mathcal{HI}^C}$ of a hybridisation $\mathcal{HI}^C$ with properly constrained models is *not* necessarily the restriction of $\models^{\mathcal{HI}}$, the satisfaction relation of $\mathcal{HI}$. Also hybridisations of institutions constitute an example of the general notion of *stratified institution* of (0).

Base logic versus hybrid logic:

In hybridised institutions, at the level of the sentences of the base institution we may have two sets of Boolean connectives, those of the hybridisation and those of the base institution (when the base institution has them). The following simple result allows us to ignore the distinction between the Boolean connectives of a hybridisation and those of the base institution. The result also states the general relationship between the quantification at the base and at the hybridised level.

Fact 3.2. For any hybridisation of $\mathcal{I}$, $(\mathrm{Sign}^{\mathcal{HI}}, \mathrm{Sen}^{\mathcal{HI}}, \mathrm{Mod}^C \subseteq \mathrm{Mod}^{\mathcal{HI}}, \models)$, let us denote the Boolean connectives and the quantifiers in the base institution $\mathcal{I}$ by $\circledwedge$, $\circledvee$, $\circledRightarrow$, $\circledneg$, and $\circledforall$, $\circledexists$, respectively. For any $(\Sigma, \mathrm{Nom}, \Lambda)$-model (M, W), any $w \in |W|$, and any sentences $\rho, \rho' \in \mathrm{Sen}^{\mathcal{I}}(\Sigma)$ of the base institution and for each $\chi \in \mathcal{D}^{\mathcal{HI}}$

- $(M, W) \models^w \rho \star \rho'$ iff $(M, W) \models^w \rho \circledast \rho'$ for $\star \in \{\wedge, \vee, \Rightarrow\}$,
- $(M, W) \models^w \neg\rho$ iff $(M, W) \models^w \circledneg\, \rho$,
- $(M, W) \models^w (\circledforall\, \chi)\rho$ implies $(M, W) \models^w (\forall(\chi, 1_{\mathrm{Nom}}, 1_\Lambda))\rho$, and
- $(M, W) \models^w (\exists(\chi, 1_{\mathrm{Nom}}, 1_\Lambda))\rho$ implies $(M, W) \models^w (\circledexists\, \chi)\rho$.

Embedding the base institution into its free hybridisation:

One may legitimately wonder about the existence of a canonical embedding of the base institution $\mathcal{I}$ into its hybridisation $\mathcal{HI}$ in the form of a comorphism $(\Phi, \alpha, \beta) : \mathcal{I} \to \mathcal{HI}$. The answer is as follows:

— $\Phi(\Sigma) = (\Sigma, \{i\}, \emptyset)$,

— $\alpha_\Sigma(\rho) = @_i \rho$, and

— $\beta_\Sigma(M, W) = M_{W_i}$.

It is easy to show that this is a conservative comorphism.

Examples:

A myriad of examples of hybridisation may be generated from our definition above by considering various instances for the three parameters of our hybridisation process: (1) the base institution $\mathcal{I}$, (2) the quantification space $\mathcal{D}^{\mathcal{HI}}$, and (3) the constrained models (Mod^C).

Example 3.1 (Hybrid propositional logic $\mathcal{H}'PL$). Applying the quantifier-free version of the hybridisation method described above to PL and fixing $\Lambda_2 = \{\lambda\}$ and $\Lambda_n = \emptyset$ for each $n \neq 2$, we obtain the institution $\mathcal{H}'PL$ of the "standard" *hybrid propositional logic* (without state quantifiers): the category of signatures is $\mathrm{Sign}^{\mathcal{H}'PL} = \mathbf{Set} \times \mathbf{Set}$ with objects denoted by (P, Nom) and morphisms by $(\varphi_{\mathrm{Sig}}, \varphi_{\mathrm{Nom}})$;[§] sentences are the usual hybrid propositional formulas, i.e., modal formulas closed by Boolean connectives, $[\lambda]$ denoted $\Box$, $\langle\lambda\rangle$ denoted $\Diamond$, and by the operator $@_i, i \in \mathrm{Nom}$; models consists of pairs (M, W) where W consists of a carrier set $|W|$, interpretations $W_i \in |W|$ for each $i \in \mathrm{Nom}$, and a binary relation $W_\lambda \subseteq |W| \times |W|$, and for each $w \in |W|$, M_s is a propositional model, i.e., a function $M_w : P \to \{0, 1\}$ which is equivalent to a subset $M_w \subseteq P$. Note that by virtue of Fact 3.2 we do not need to make a distinction between the Boolean connectives at the level of PL and at the level of $\mathcal{H}'PL$.

The T, $S4$, and $S5$ versions of hybrid propositional logic are obtained by constraining the models of $\mathcal{H}'PL$ to those models (M, W) for which W_λ is reflexive, preorder, and equivalence, respectively.

When we relax to arbitrary sets of modalities Λ rather than only λ, we obtain the "multi-modal hybrid propositional logic".

A challenging issue concerns finding suitable quantification spaces to capture versions of hybrid propositional logic. One choice is the quantifier-free version in which $\mathcal{D}^{\mathcal{H}'PL}$ would consist only of identities. However, it would be interesting, along the hybridisation process, to capture a quantifier such as $\mathbf{E}$, where $\mathbf{E}\rho$ means that "*ρ is true in some state of the model*" (0). Considering as a quantification space the extensions of signatures with nominal symbols, paves the way to express the following properties:

[§] Note that by fixing Λ to only one symbol of arity 2 means the restriction to a subcategory of the $\mathcal{H}PL$-signatures, i.e. $\mathrm{Sign}^{\mathcal{H}'PL} \subseteq \mathrm{Sign}^{\mathcal{H}PL}$. Then $\mathcal{H}'PL$ is the 'sub-institution' of $\mathcal{H}PL$ determined by this restriction of the signatures.

$(M,W) \models^w ((\forall i)i) \Leftrightarrow \rho$	iff	ρ is satisfied at w iff w is unique in (M,W)
$(M,W) \models^w (\exists i)@_i\rho$	iff	$(M,W) \models \mathbf{E}\rho$

Let us denote this modification of $\mathcal{H}'PL$ by $\mathcal{H}'PLn$. A *block of nominal variables* X for a $\mathcal{H}'PLn$ signature (P,Nom) is a finite set nominal variables of the form (x,P,Nom) (like in the case of FOL variables, x is the name and (P,Nom) the qualification of the variable) such that $(x,P,\mathrm{Nom}),(x',P,\mathrm{Nom}) \in X$ implies $x = x'$. Then $\mathcal{D}^{\mathcal{H}'PLn}$ may be defined as consisting of the signature extensions with blocks of nominal variables, i.e. $(P,\mathrm{Nom}) \hookrightarrow (P,\mathrm{Nom}\cup X)$. For any signature morphism $\varphi : (P,\mathrm{Nom}) \to (P',\mathrm{Nom}')$ and X block of nominal variables for (P,Nom) we define $X^\varphi = \{(x,P',\mathrm{Nom}') \mid (x,P,\mathrm{Nom}) \in X\}$. Then $\chi(\varphi)$ is the extension $(P',\mathrm{Nom}') \hookrightarrow (P',\mathrm{Nom}'\cup X^\varphi)$ and $\varphi[\chi]$ is the canonical extension of φ that maps each (x,P,Nom) to (x,P',Nom').

When we combine this quantification with the constraints T, S4, S5, etc., then we have to establish the adequacy condition for the constrained model sub-functor. However in this case this is almost trivial since we may consider $\mathcal{D}^{PL}$ (the quantification space at the level of the base institution) as being trivial and then reflection condition for the constrained models gets trivialised too.

Example 3.2 (Double layered hybridisation $\mathcal{H}'\mathcal{H}'PL$). An institution for specification of hierarchical state transition systems is achieved by a double layered hybridisation of PL (but it could also be extended to any other base institution instead of PL). This means a hybridisation of $\mathcal{H}'PL$ (or of any of its variants from Ex. 3.1). Let us denote by $\mathcal{H}'\mathcal{H}'PL$ the quantifier-free hybridisation of $\mathcal{H}'PL$. The models of this institution are “Kripke structures of Kripke structures”.

Thus the $\mathcal{H}'\mathcal{H}'PL$ signatures are triples $(P,\mathrm{Nom}^0,\mathrm{Nom}^1)$ with Nom^0 and Nom^1 denoting the nominals of the first and second layer of hybridisation, respectively. In order to prevent potential ambiguities, in general we tag the symbols of the respective layers of hybridisation by the superscripts 0 (for the first layer) and 1 (for the second layer). This convention should include nominals and connectors ($\Diamond$, $\wedge$, etc.); however by Fact 3.2 in the case of the Boolean connectors we may skip this. For instance, the expression $@_{j^1}k^0 \wedge \Box^1\rho$ is a sentence of $\mathcal{H}'\mathcal{H}'PL$ where the symbols k and j represent nominals of the first and second level of hybridisation and ρ a PL sentence. On the other hand, according to this tagging convention the expression $@_{j^0}k^1 \wedge \Box^1\rho$ would not parse. Our tagging convention is extended also to $\mathcal{H}'\mathcal{H}'PL$ models: a $(P,\mathrm{Nom}^0,\mathrm{Nom}^1)$-model is denoted by (M^1,W^1) where for any $w \in |W^1|$ the models M^1_w are denoted by (W^0_w,M^0_w).

We may consider also quantified versions of the double hybridisation of PL, and there are several variants of those depending on the existence of the quantifiers at each layer. Let us denote these generically by $\mathcal{H}'(\mathcal{H}'PLx)y$ where x is either empty or n and y is either empty, n^0, n^1, or n^{01}. The absence of x and/or y means the absence of quantification at the base and/or the upper level, respectively. The superscripts 0 and 1 tagged to n denote the existence of quantification at the upper level with nominals variables from the lower and upper level of hybridisation, respectively. For example $\mathcal{H}'(\mathcal{H}'PLn)n^{01}$ would have all possible quantifications, such as $(\forall^0 i^0)$, $(\forall^1 i^0)$, $(\forall^1 i^1)$ while $\mathcal{H}'(\mathcal{H}'PL)n^1$ would have only those of the form $(\forall^1 i^1)$.

Let $\mathcal{H}'\mathcal{H}'PL'$ denote the double hybridisation obtained by constraining the models (M^1,W^1)

to those such that for any $w, w' \in |W^1|$ and any $i^0 \in \mathrm{Nom}^0$ we have that $|W^0_w| = |W^0_{w'}|$ and $(W^0_w)_{i^0} = (W^0_{w'})_{i^0}$. The quantified variants of $\mathcal{H}'\mathcal{H}'PL'$ would require the reflection condition of Def. 3.1; however this follows easily.

The layered hybridisation construction (together with its associated notational conventions) may be iterated to higher layers of hybridisations, e.g. $\mathcal{H}^3PL$, $\mathcal{H}^4PL$, ... The convention on quantified versions carries also forward to such higher layers of hybridisation.

Example 3.3 (Hybrid first order logic $\mathcal{H}FOL$). Through the application of the hybridisation method to FOL by taking as a quantification space signature extensions both with FOL variables and variables over nominals, one captures the state-variables quantification of the multi-modal variant of *first-order hybrid logic* of (0). Like in the case of $\mathcal{H}'PL$, by virtue of the Fact 3.2 note that we do not need to make a distinction between the Boolean connectives at the level of FOL and those at the level of $\mathcal{H}FOL$. Moreover, because the carriers of the FOL models are non-empty we may easily show that in this case the implications of Fact 3.2 about quantifiers may be turned into equivalences, hence it is also not necessary to distiguish between quantifiers at the base FOL level and at the hybridised $\mathcal{H}FOL$ level.

Example 3.4 (Predefined sharing in $\mathcal{H}REL$). Let $\mathcal{H}REL'$ be the hybridisation of REL that constraints the models of $\mathcal{H}REL$ to those models (M, W) such that $\{M_i \mid i \in |W|\}$ share the same universe (underlying set) and the same interpretation of the constants. It is rather easy to note that the amalgamation of models preserves the sharing, hence the reflection condition of Dfn. 3.1 is fulfilled.

$\mathcal{D}^{\mathcal{H}REL'}$ consists of the signature extensions with FOL variables (for the states) and with nominal variables (in the style of $\mathcal{D}^{\mathcal{H}'PLn^1}$ of Ex. 3.1).

Note that like for $\mathcal{H}FOL$, in $\mathcal{H}REL'$ we also do not need to distinguish between the Boolean connectives and the quantifiers at the base and at the hybridised level.

Example 3.5 (User defined sharing in $\mathcal{H}FOL$). The above Ex. 3.4 may be considered an example of 'predefined' or 'default' sharing since the interpretation of *all* constants are shared. However in formal specification applications it is also important to consider 'user defined' sharing, in which one has the possibility to define at hand the entities to be shared. The first order modal logic institution $MFOL$ of (0) is such an example. Its hybrid version $\mathcal{H}FOLR'$ may be developed through the hybridisation process above as follows.

As the base institution of the hybridisation we consider the institution $FOLR$ defined as follows:

- Sign^{FOLR} is the category of the $MFOL$ signatures of (0): its objects are tuples (S, S_0, F, F_0, P, P_0) where (S_0, F_0, P_0) and (S, F, P) are FOL signatures such that (S_0, F_0, P_0) is a sub-signature of (S, F, P); the symbols of (S_0, F_0, P_0) are called 'rigid', and signature morphisms $\varphi : (S, S_0, F, F_0, P, P_0) \to (S', S'_0, F', F'_0, P', P'_0)$ are just FOL signature morphisms $(S, F, P) \to (S', F', P')$ that map rigid symbols to rigid symbols.
- $\mathrm{Sen}^{FOLR}(S, S_0, F, F_0, P, P_0)$ consists of those sentences in $\mathrm{Sen}^{FOL}(S, F, P)$ that contain only quantifiers over rigid variables,
- $\mathrm{Mod}^{FOLR}(S, S_0, F, F_0, P, P_0) = \mathrm{Mod}^{FOL}(S, F, P)$, and
- the satisfaction relation in $FOLR$ is induced canonically from FOL, i.e. $\models^{FOLR}_{(S,S_0,F,F_0,P,P_0)} = \models^{FOL}_{(S,F,P)}$.

We let $\mathcal{H}FOLR$ be the hybridisation of $FOLR$ with quantifications by nominal and *rigid* FOL variables.

For $\mathcal{H}FOLR'$ let us consider the constrained model sub-functor Mod^C such that $(M, W) \in |\mathrm{Mod}^C(\Sigma, \mathrm{Nom}, \Lambda)|$ if and only if for all $i, j \in |W|$ and each rigid symbol x in Σ, $(M_i)_x = (M_j)_x$. For any pushout square of signature morphisms in $\mathrm{Sign}^{\mathcal{H}FOLR}$ as below

$$\begin{tikzcd}
(\Sigma, \mathrm{Nom}, \Lambda) \arrow[r, "\varphi_1"] \arrow[d, "\varphi_2"'] & (\Sigma_1, \mathrm{Nom}_1, \Lambda_1) \arrow[d, "\theta_1"] \\
(\Sigma_2, \mathrm{Nom}_2, \Lambda_2) \arrow[r, "\theta_2"'] & (\Sigma', \mathrm{Nom}', \Lambda')
\end{tikzcd}$$

let us consider a constrained $(\Sigma_k, \mathrm{Nom}_k, \Lambda_k)$-model (M_k, W_k) for each $k \in \{1, 2\}$ such that $\mathrm{Mod}^C(\varphi_1)(M_1, W_1) = \mathrm{Mod}^C(\varphi_2)(M_2, W_2)$. We take the amalgamation $(M', W') = (M_1, W_1) \otimes (M_2, W_2)$ according to Thm. 3.1. Then we consider any rigid symbol x of Σ' and any $i, j \in |W'|$. By Prop. 3.1 we have that

$$\begin{tikzcd}
\Sigma \arrow[r, "(\varphi_1)_{\mathrm{Sig}}"] \arrow[d, "(\varphi_2)_{\mathrm{Sig}}"'] & \Sigma_1 \arrow[d, "(\theta_1)_{\mathrm{Sig}}"] \\
\Sigma_2 \arrow[r, "(\theta_2)_{\mathrm{Sig}}"'] & \Sigma'
\end{tikzcd}$$

is a pushout square of $FOLR$ signature morphisms. Note that the set of rigid symbols in Σ' is the the union of the translations of the rigid symbols from both Σ_1 and Σ_2 through $(\theta_1)_{\mathrm{Sig}}$ and $(\theta_2)_{\mathrm{Sig}}$. This means that there exists $k \in \{1, 2\}$ and x_k rigid symbol of Σ_k such that $x = \theta_k(x_k)$. It follows that $(M'_i)_x = ((M_k)_i)_{x_k}$ and $(M'_j)_x = ((M_k)_j)_{x_k}$, hence $(M'_i)_x = (M'_j)_x$ since $((M_k)_i)_{x_k} = ((M_k)_j)_{x_k}$ (because (M_k, W_k) is a constrained model). This proves that (M', W') is a constrained model, which gives the reflection condition of Dfn. 3.1.

The first order hybrid logic (e.g. (0)) appears as a fragment of $\mathcal{H}FOLR'$ when we discard the function symbols but constants and there are no rigid predicates.

Example 3.6 (User defined sharing in Hybrid Partial Algebra). Let PAR be a rigid version of the partial algebras institution of Ex. 2.2 that is defined similarly to $FOLR$, the rigid version of FOL from Ex. 3.5. This means we consider signatures of the form $(S, S_0, TF, TF_0, PF, PF_0)$ with (S_0, TF_0, PF_0) being a sub-signature of 'rigid symbols' for a PA signature (S, TF, PF), etc; we skip here the other details that replicate the corresponding details from the definition of $FOLR$. Let $\mathcal{H}PAR$ be the hybridisation of PAR with quantifications by nominals and PAR variables (i.e. rigid total variables); this means $\mathcal{D}^{\mathcal{H}PAR}$ consists of the signature extensions with *total rigid* (first-order) variables and with nominals variables. The amalgamation property of PA entails the adequacy of $\mathcal{D}^{PAR}$ for Mod^{PAR}. From Corollary 3.2 it follows that $\mathcal{D}^{\mathcal{H}PAR}$ is adequate for $\mathcal{H}PAR$.

We denote by $\mathcal{H}PAR'$ the hybridisation obtained by constraining the model sub-functor to Mod^C defined by $(M, W) \in |\mathrm{Mod}^C(\Sigma, \mathrm{Nom}, \Lambda)|$ if and only if the rigid sorts and total functions share the same interpretations in all the states and the rigid partial functions share the domains. This means that for all $i, j \in |W|$ and for each symbol x in S_0 or TF_0, $(M_i)_x = (M_j)_x$

and for any σ in PF_0 we have that $\mathsf{dom}((M_i)_\sigma) = \mathsf{dom}((M_j)_\sigma)$. The reflection condition for Mod^C is established in this case similarly to the corresponding reflection condition from Ex. 3.5.

A version of this example may require that the values of the rigid partial functions are also shared. Our choice of model constraints for $\mathcal{H}PAR'$ is on the one hand an illustration of the high flexibility given by the generality of our approach, and on the other hand constitutes the adequate choice for the logic platform of the case study of Sect. 6. In that case study we will consider models with two states, queues in one state and stacks in the other. The partial functions on queues and stacks are defined when these are non-empty (so the same definition domain), but they may give different values.

Example 3.7 (Temporalisation of logics). The general method of temporalisation of logics proposed in (0) is subsumed in a very simple way by our approach by considering the unconstrained hybridisation $\mathcal{HI}$ of an abstract institution $\mathcal{I}$ with the quantification space $\mathcal{D}^{\mathcal{HI}}$ consisting of finite extensions with nominal variables, and by restricting the signatures to those that have only one modality symbol of arity 2. A concrete example is $\mathcal{H}PL'n$ of Ex. 3.1. Then the generic modal operators 'since' and 'until' can be expressed by using the hybrid features. For example $Until(\rho_1, \rho_2)$ can be expressed (0) by

$$(\exists y)(\Diamond(y \wedge \rho_1) \wedge \Box(\Diamond y \Rightarrow \rho_2)).$$

The linearity of the 'time flow' (i.e. the binary relation associated to the modality symbol), which is necessary for many of developments in (0) can be captured in our framework as a model constraint. In this case the reflection condition is trivial since it is a condition only on the W part of the models and the reducts do not affect that.

The following table presents an overview of some of the examples discussed in this section.

hybridised institution	Base institution	Quantification space: first order	Quantification space: Nom	Quantification space: Λ	Model constraint
$\mathcal{H}'PL$	PL				
$\mathcal{H}'PL(T)$	PL				W reflexive
$\mathcal{H}'PLn$	PL		✓		
$\mathcal{H}'PLn(T)$	PL		✓		W reflexive
$\mathcal{H}'\mathcal{H}'PL$	$\mathcal{H}'PL$				
$\mathcal{H}'(\mathcal{H}'PL)n^1$	$\mathcal{H}'PL$		Nom^1		
$\mathcal{H}'(\mathcal{H}'PLn)$	$\mathcal{H}'PLn$				
$\mathcal{H}'\mathcal{H}'PL'$	$\mathcal{H}'PL$				$\lvert W^0_w\rvert = \lvert W^0_{w'}\rvert$ and $(W^0_w)_{i^0} = (W^0_{w'})_{i^0}$ for all $w, w' \in \lvert W^1\rvert$
$\mathcal{H}FOL$	FOL	✓	✓		
$\mathcal{H}REL$	REL	✓	✓		
$\mathcal{H}REL'$	REL	✓	✓		$\lvert M_i\rvert = \lvert M_j\rvert$ and $(M_i)_\sigma = (M_j)_\sigma$ for each constant σ
$\mathcal{H}FOLR$	$FOLR$	rigid	✓		
$\mathcal{H}FOLR'$	$FOLR$	rigid	✓		$(M_i)_x = (M_j)_x$ for each x in S_0, F_0, P_0
$\mathcal{H}PAR$	PAR	total & rigid	✓		
$\mathcal{H}PAR'$	PAR	total & rigid	✓		$(M_i)_x = (M_j)_x$ for each x in S_0, TF_0, and $\mathsf{dom}((M_i)_\sigma) = \mathsf{dom}((M_j)_\sigma)$ for each σ in PF_0

The following graph shows an expressiveness hierarchy for some of the examples in this section.

$$\begin{tikzcd}
 & & \mathcal{H}'PL \arrow[dl] \arrow[d] & & & \\
 & \mathcal{H}'\mathcal{H}'PL \arrow[dl] & \mathcal{H}'PLn \arrow[dl] \arrow[d] \arrow[drr] & & & \\
\mathcal{H}'\mathcal{H}'PLn^1 & \mathcal{H}'(\mathcal{H}'PLn) & \mathcal{H}FOL \arrow[dr] & & \mathcal{H}REL \arrow[dl] \arrow[d] & \\
 & & & \mathcal{H}FOLR \arrow[dr] & \mathcal{H}REL' \arrow[d] & \mathcal{H}PAR \arrow[d] \\
 & & & & \mathcal{H}FOLR' & \mathcal{H}PAR'
\end{tikzcd}$$

4. Encoding hybridised institutions into FOL

This is the main section of the paper and it is structured as follows:

1 We develop some technical preliminaries that will be used for developing the main result of our paper.
2 We develop the encoding of hybridised institutions into FOL at the general level; this may be regarded as a high generalisation of the standard translations of hybrid logics found in the literature (e.g. (0; 0)).
3 We instantiate the general encoding to a series of examples of concrete encodings.

4.1. *Technical preliminaries*

In order to ease the burden represented by the complexity of the general encoding of hybridised institutions into FOL we introduce now a series of notations and develop a technical lemma. All these concern only FOL, but they will be used immediately after.

Notation 4.1. For any FOL-signature (S, F, P) we denote by $([S], [F], [P])$ the following FOL-signature:

— $[S] = S \cup \{\mathrm{ST}\}$, where ST is a designated sort not in S,

— $[F]_{\underline{\mathrm{ar}} \to s} = \begin{cases} F_{\underline{\mathrm{ar}}' \to s} & \text{for any } s \in S, \underline{\mathrm{ar}}' \in S^* \text{ such that } \underline{\mathrm{ar}} = (\mathrm{ST})\underline{\mathrm{ar}}' \\ \emptyset & \text{for the other cases;} \end{cases}$

— $[P]_{\underline{\mathrm{ar}}} = \begin{cases} P_{\underline{\mathrm{ar}}'} & \text{for any } \underline{\mathrm{ar}}' \in S^* \text{ such that } \underline{\mathrm{ar}} = (\mathrm{ST})\underline{\mathrm{ar}}'; \\ \emptyset, & \text{for the other cases.} \end{cases}$

For any morphism of FOL signatures $\varphi : \ (S, F, P) \to (S', F', P')$ we let $[\varphi] : \ ([S], [F], [P]) \to ([S'], [F'], [P'])$ morphism of FOL signatures defined as follows:

— $[\varphi]^{\mathrm{st}}(\mathrm{ST}) = \mathrm{ST}$,

— $[\varphi]^{\mathrm{st}}(s) = \varphi^{\mathrm{st}}(s)$ for any $s \in S$,

— $[\varphi]^{\mathrm{op}}_{(\mathrm{ST})\underline{\mathrm{ar}}' \to s}(\sigma) = \varphi^{\mathrm{op}}_{\underline{\mathrm{ar}}' \to s}(\sigma)$ for any $\sigma \in F_{\underline{\mathrm{ar}}' \to s}$, and

— $[\varphi]^{\mathrm{rl}}_{(\mathrm{ST})\underline{\mathrm{ar}}'}(\pi) = \varphi^{\mathrm{rl}}_{\underline{\mathrm{ar}}'}(\pi)$ for any $\pi \in P_{\underline{\mathrm{ar}}'}$.

Definition 4.1. For any FOL-signature (S, F, P) and any new constant x of sort ST we define the following translation

$$[\text{-}]^x_{(S,F,P)} : \ \mathrm{Sen}^{FOL}(S, F, P) \to \mathrm{Sen}^{FOL}([S], [F] + x, [P])$$

defined by

— $[t = t']^x = ([t]^x = [t']^x)$ where $[\sigma(t_1, \ldots, t_n)]^x = \sigma(x, [t_1]^x, \ldots, [t_n]^x)$;
— $[\pi(t)]^x = \pi(x, [t]^x)$;
— $[\rho_1 \star \rho_2]^x = [\rho_1]^x \star [\rho_2]^x$, for $\star \in \{\vee, \wedge, \Rightarrow\}$;
— $[\neg\rho]^x = \neg[\rho]^x$;
— $[(\forall Y)\rho]^x = (\forall Y)([\rho]^x)_Y$ where $([\rho]^x)_Y$ is the result of replacing in $[\rho]^x$ all occurrences of $y(z)$ by y for each y in Y.¶

Definition 4.2. Let (S, F, P) be any FOL-signature.

— For any $s \in S$ let us denote by D_s a new designated relation symbol with arity $(\mathrm{ST})s$;
— For any $\sigma \in F_{s_1 \ldots s_n \to s}$, by D_σ we denote the Horn sentence

$$(\forall y)(\forall x_1, \ldots, x_n) \bigwedge_{1 \le i \le n} D_{s_i}(y, x_i) \Rightarrow D_s(y, \sigma(y, x_1, \ldots, x_n))$$

— $D_F = \{D_\sigma \mid \sigma \in F_{\underline{\mathrm{ar}} \to s}, \underline{\mathrm{ar}} \in S^*, s \in S\}$.

Definition 4.3. For any FOL-signature (S, F, P) and any $([S], [F], [P])$-model M' such that $M' \models D_F$, for any $w \in M'_{\mathrm{ST}}$ the (S, F, P)-model $M'|_w$ is defined as follows:

— for each $s \in S$, $(M'|_w)_s = \{m \in M'_s \mid (w, m) \in M'_{D_s}\}$;
— for each σ in F, $(M'|_w)_\sigma(m) = M'_\sigma(w, m)$;
— for each π in P, $m \in (M'|_w)_\pi$ iff $(w, m) \in M'_\pi$.

Let us note that the correctness of the definition of $M'|_w$, i.e. that for each $\sigma \in F_{\underline{\mathrm{ar}} \to s}$ and each $m \in (M'|_w)_{\underline{\mathrm{ar}}}$ we have $(M'|_w)_\sigma(m) \in (M'|_w)_s$, relies upon the condition that $M' \models D_F$.

Notation 4.2. For any (S, F, P)-sentence ρ, by $V(\rho)$ we denote the set of all sentences $(\forall x, y)D_s(x, y)$ for s any sort of a variable in a quantification that occurs in ρ. For any set E of sentences $V(E)$ denotes $\cup\{V(\rho) \mid \rho \in E\}$.

Lemma 1. For any FOL-signature (S, F, P), any $([S], [F], [P])$-model M' with $M' \models D_F$, any (S, F, P)-sentence ρ, and any $w \in M'_{\mathrm{ST}}$, if $M' \models V(\rho)$ then

$$M'|_w \models_{(S,F,P)} \rho \ \text{ if and only if } \ M'^w \models_{([S],[F]+x,[P])} [\rho]^x \tag{6}$$

where M'^w denotes the expansion of M' to $([S], [F] + x, [P])$ defined by $M'^w_x = w$.

Proof. The proof of the lemma is by induction on the structure of ρ as follows.

1 The proof for the case when ρ is $t = t'$ is an immediate consequence of the following relation

$$(M'|_w)_t = (M'^w)_{[t]^x}, \ \text{ for any term } \ t \tag{7}$$

which is proved by induction on the structure t as follows:

¶ Note that the signature of ρ contains y as constants, hence in $[\rho]^x$ each y of Y appears as a unary function $y(x)$ (since by definition $[y]^x = y(x)$). Then $([\rho]^x)_Y$ collapses back each $y(x)$ to the constant y.

$(M'|_w)_{\sigma(t_1,\ldots,t_n)}$
 = $(M'|_w)_\sigma((M'|_w)_{t_1},\ldots,(M'|_w)_{t_n})$ (definition of evaluation of terms)
 = $M'_\sigma(w,(M'|_w)_{t_1},\ldots,(M'|_w)_{t_n})$ (definition of $M'|_w$)
 = $M'_\sigma(w,M'^w_{[t_1]^x},\ldots,M'^w_{[t_n]^x})$ (induction hypothesis)
 = $M'^w_{\sigma(x,[t_1]^x,\ldots,[t_n]^x)}$ ($M'^w_x = w$ and definition of term evaluation)
 = $M'^w_{[\sigma(t_1,\ldots,t_n)]^x}$ (definition of $[\_]^x$).

2 If ρ is $\pi(t_1,\ldots,t_n)$:
$M'|_w \models \pi(t_1,\ldots,t_n)$
 iff $((M'|_w)_{t_1},\ldots,(M'|_w)_{t_n}) \in (M'|_w)_\pi$ (definition of FOL-satisfaction)
 iff $(w,M'^w_{[t_1]^x},\ldots,M'^w_{[t_n]^x}) \in M'_\pi$ (definition of $(M'|_w)_\pi$ and by (7))
 iff $M'^w \models \pi(x,[t_1]^x,\ldots,[t_n]^x)$ (because $M'^w_x = w$)
 iff $M'^w \models [\pi(t_1,\ldots,t_n)]^x$ (definition of $[\_]^x$).

3 When ρ is $\xi_1 \star \xi_2$ for $\star \in \{\wedge,\vee,\Rightarrow\}$ or ρ is $\neg\xi$, the proof reduces to a plain application of the induction hypothesis.

4 If ρ is $(\forall Y)\xi$:

$$M'|_w \models (\forall Y)\xi \text{ iff } M'' \models \xi \text{ for any } (S,F+Y,P)\text{-expansion } M'' \text{ of } M'|_w, \text{ and}$$

$$M'^w \models (\forall Y)([\xi]^x)_Y \text{ iff } N'^w \models ([\xi]^x)_Y \text{ for any } ([S],[F]{+}Y{+}x,[P])\text{-expansion } N'^w \text{ of } M'^w.$$

This case is solved if we proved the equivalence between the right hand sides of the above two equivalences. This follows by noting the following facts:

— There is a canonical bijection between the $([S],[F]{+}Y{+}x,[P])$-expansions N'^w of M'^w and the $(S,F+Y,P)$-expansions M'' of $M'|_w$ given by $M''_y = N'^w_y$ for each $y \in Y$. This relies upon the fact that $(M'|_w)_s = M'_s$ which follows from $M' \models V(\rho)$.
— Each N'^w as above determines an $([S],[F+Y]+x,[P])$-expansion N''^w of M'^w by $N''^w_y(m) = N'^w_y$ for all $m \in M'_{\mathrm{ST}}$ and each $y \in Y$. Furthermore

$$N''^w \models [\xi]^x \text{ if and only if } N'^w \models ([\xi]^x)_Y. \tag{8}$$

— Let N'' be the reduct of N''^w to $([S],[F+Y],[P])$. Then $M'' = N''|_w$.
— The induction hypothesis gives that $M'' \models \xi$ iff $N''^w \models [\xi]^x$. By (8) it follows that $M'' \models \xi$ iff $N'^w \models ([\xi]^x)_Y$.

□

4.2. *The definition of the encoding*

Thus, let $(\mathrm{Sign}^{\mathcal{HI}},\mathrm{Sen}^{\mathcal{HI}},\mathrm{Mod}^C,\models)$ be a hybridisation of an institution $\mathcal{I}$ such that for all $\chi \in \mathcal{D}^{\mathcal{HI}}$:
— χ_{Nom} are finite extensions, and
— χ_{MS} are identities.

Given any comorphism $(\Phi,\alpha,\beta): \mathcal{I} \to FOL^{\mathrm{pres}}$ such that for each $\varphi: \Sigma \to \Sigma'$ in $\mathcal{D}^{\mathcal{I}}$ we have that
— the underlying FOL signature morphism of $\Phi(\varphi)$ is in $\mathcal{D}^{FOL}$; and
— the difference between the presentations $\Phi(\Sigma')$ and $\Phi(\Sigma)$ consists of a finite set Γ_φ of sentences,

we define a comorphism $(\Phi'^C, \alpha', \beta'^C) : (\mathrm{Sign}^{\mathcal{HI}}, \mathrm{Sen}^{\mathcal{HI}}, \mathrm{Mod}^C, \models) \to FOL^{\mathrm{pres}}$ in two steps:

1 We define a functor $\Phi' : \mathrm{Sign}^{\mathcal{HI}} \to \mathrm{Sign}^{FOL^{\mathrm{pres}}}$ and natural transformations $\alpha' : \mathrm{Sen}^{\mathcal{HI}} \Rightarrow \Phi'; \mathrm{Sen}^{FOL^{\mathrm{pres}}}$ and $\beta' : \Phi'^{\mathrm{op}}; \mathrm{Mod}^{FOL^{\mathrm{pres}}} \Rightarrow \mathrm{Mod}^{\mathcal{HI}}$.

2 We extend the definitions of Φ' and β' to Φ'^C and β'^C respectively and prove the Satisfaction Condition for $(\Phi'^C, \alpha', \beta'^C)$.

Definition 4.4 (Translation of the signatures). For any $\mathcal{HI}$ signature $(\Sigma, \mathrm{Nom}, \Lambda)$, let

$$\Phi'(\Sigma, \mathrm{Nom}, \Lambda) = ([S_\Sigma], [F_\Sigma] + \overline{\mathrm{Nom}}, (D_s)_{s \in S_\Sigma} + [P_\Sigma] + \overline{\Lambda}, \overline{\Gamma_\Sigma} \cup D_{F_\Sigma})$$

where

— $\Phi(\Sigma) = ((S_\Sigma, F_\Sigma, P_\Sigma), \Gamma_\Sigma)$, where $(S_\Sigma, F_\Sigma, P_\Sigma)$ is a FOL-signature and Γ_Σ is a set of $(S_\Sigma, F_\Sigma, P_\Sigma)$-sentences;

— $(\overline{\mathrm{Nom}})_{\underline{\mathrm{ar}} \to s} = \begin{cases} \mathrm{Nom} & \text{when } \underline{\mathrm{ar}} = \emptyset, s = \mathrm{ST}, \\ \emptyset & \text{for the other cases;} \end{cases}$

— $(\overline{\Lambda})_{\underline{\mathrm{ar}}} = \begin{cases} \Lambda_n & \text{when } \underline{\mathrm{ar}} = (\mathrm{ST})^n, n \in \omega \\ \emptyset & \text{for the other cases;} \end{cases}$

— $\overline{\Gamma_\Sigma} = \{\forall x\, [\gamma]^x \mid \gamma \in \Gamma_\Sigma\} \cup V(\Gamma_\Sigma)$.

Definition 4.5 (Translation of the signature morphisms). For any $\mathcal{HI}$ signature morphism $\varphi = (\varphi_{\mathrm{Sig}}, \varphi_{\mathrm{Nom}}, \varphi_{\mathrm{MS}}) : (\Sigma_1, \mathrm{Nom}_1, \Lambda_1) \to (\Sigma_2, \mathrm{Nom}_2, \Lambda_2)$ the FOL^{pres} signature morphism $\Phi'(\varphi)$ is the extension of $[\Phi(\varphi_{\mathrm{Sig}})] : ([S_{\Sigma_1}], [F_{\Sigma_1}], [P_{\Sigma_1}]) \to ([S_{\Sigma_2}], [F_{\Sigma_2}], [P_{\Sigma_2}])$ defined by

— $\Phi'(\varphi)^{\mathrm{op}}(n) = \varphi_{\mathrm{Nom}}(n)$ for each $n \in \mathrm{Nom}_1$, and

— $\Phi'(\varphi)^{\mathrm{rl}}(D_s) = D_{\Phi(\varphi_{\mathrm{Sig}})^{\mathrm{st}}(s)}$ for each sort $s \in S_{\Sigma_1}$,

— $\Phi'(\varphi)^{\mathrm{rl}}(\lambda) = \varphi_{\mathrm{MS}}(\lambda)$ for each $\lambda \in \Lambda_1$.

Fact 4.1. $\Phi'(\varphi)$ of Dfn. 4.5 is a presentation morphism $\Phi'(\Sigma_1, \mathrm{Nom}_1, \Lambda_1) \to \Phi'(\Sigma_2, \mathrm{Nom}_2, \Lambda_2)$.

In quantified sentences part of the following definition we may assume without any loss of generality quantifications with only one nominal variable and only one first order variable symbol.

Definition 4.6 (Translation of the sentences). $\alpha'_{(\Sigma,\mathrm{Nom},\Lambda)}(\rho) = (\forall x)\alpha'^x_{(\Sigma,\mathrm{Nom},\Lambda)}(\rho)$, where $\alpha'^x_{(\Sigma,\mathrm{Nom},\Lambda)} : \mathrm{Sen}^{\mathcal{HI}}(\Sigma, \mathrm{Nom}, \Lambda) \to \mathrm{Sen}^{FOL}([S_\Sigma], [F_\Sigma] + \overline{\mathrm{Nom}} + x, (D_s)_{s \in S} + [P_\Sigma] + \overline{\Lambda})$ with x being a constant of sort ST, is defined by

— $\alpha'^x(i) = (i = x)$, $i \in \mathrm{Nom}$;

— for each $\rho \in \mathrm{Sen}^{\mathcal{I}}(\Sigma)$, $\alpha'^x(\rho) = [\alpha_\Sigma(\rho)]^x$

— $\alpha'^x(\rho_1 \star \rho_2) = \alpha'^x(\rho_1) \star \alpha'^x(\rho_2)$, $\star \in \{\vee, \wedge, \Rightarrow\}$;

— $\alpha'^x(\neg\rho) = \neg\alpha'^x(\rho)$;

— $\alpha'^x(@_i\rho) = \alpha'^i(\rho)$;

— $\alpha'^x([\lambda](\rho_1, \ldots, \rho_n)) = \forall y_1, \ldots, y_n \left(\lambda(x, y_1, \ldots, y_n) \Rightarrow \bigvee_{1 \le i \le n} \alpha'^{y_i}(\rho_i)\right)$;

— $\alpha'^x(\langle\lambda\rangle(\rho_1, \ldots, \rho_n)) = \exists y_1, \ldots, y_n \left(\lambda(x, y_1, \ldots, y_n) \wedge \bigwedge_{1 \le i \le n} \alpha'^{y_i}(\rho_i)\right)$;

— $\alpha'^x_{(\Sigma,\mathrm{Nom},\Lambda)}((\forall i)\rho) = (\forall i)\alpha'^x_{(\Sigma,\mathrm{Nom}+i,\Lambda)}(\rho)$ for $\rho \in \mathrm{Sen}^{\mathcal{HI}}(\Sigma, \mathrm{Nom} + i, \Lambda)$;

— $\alpha'^{x}_{\Delta}((\forall\chi)\rho) = (\forall y)\big(\overline{\Gamma_{\chi_{\rm Sig}}} \cup \{D_y\} \Rightarrow \alpha'^{x}_{\Delta'}(\rho)\big)_y$ for $\rho \in {\rm Sen}^{\mathcal{HI}}(\Delta')$ (where $\chi = (\chi_{\rm Sig}, 1_{\rm Nom}, 1_\Lambda) : \Delta = (\Sigma, {\rm Nom}, \Lambda) \to \Delta' = (\Sigma', {\rm Nom}, \Lambda)$ and $\Phi(\chi_{\rm Sig})$ extends the signature of $\Phi(\Sigma)$ with the variable y and the presentation $\Phi(\Sigma)$ with the finite set of sentences $\Gamma_{\chi_{\rm Sig}}$).

Note that in the definition of the translations of quantified sentences, for the sake of clarity, and without any loss of generality, we have treated quantification by nominals and by base institution signature morphisms separately, and we have considered single variables instead of finite blocks of variables. We have also omitted the case of the existential quantifications which get a translation that replicates that of the universal quantifications.

From the naturality of α it follows:

Fact 4.2. α' is natural transformation.

Definition 4.7 (Translation of the models). For any $\mathcal{HI}$ signature $(\Sigma, {\rm Nom}, \Lambda)$ and any $\Phi'(\Sigma, {\rm Nom}, \Lambda)$-model M' we define $\beta'_{(\Sigma,{\rm Nom},\Lambda)}(M') = (M, W)$ where

— W is the reduct $M'\!\upharpoonright_{(\{\rm ST\},\overline{\rm Nom},\overline{\Lambda})}$, i.e. $|W| = M'_{\rm ST}$, $W_i = M'_i$ for each $i \in {\rm Nom}$, and $W_\lambda = M'_\lambda$ for each λ in Λ, and

— $M : |W| \to |{\rm Mod}^{\mathcal{I}}(\Sigma)|$ is defined for each $w \in |W|$ by $M_w = \beta_\Sigma(M'|_w)$ where $M'|_w$ denotes here the abbreviation $(M'\!\upharpoonright_{([S_\Sigma],[F_\Sigma],[P_\Sigma])})|_w$.

Lemma 2. Dfn. 4.7 is correct, in the sense that for each $w \in |W|$, $M'|_w \models \Gamma_\Sigma$.

Proof. Since $M' \models V(\Gamma_\Sigma) \cup D_{F_\Sigma}$ we may apply the conclusion of Lemma 1 from the right to the left for each $\gamma \in \Gamma_\Sigma$. In order to do this we have just to note that because of $M' \models \overline{\Gamma_\Sigma}$, we have that $M' \models (\forall x)[\gamma]^x$ for each $\gamma \in \Gamma_\Sigma$, hence $M'^w \models [\gamma]^x$ for all w and for each $\gamma \in \Gamma_\Sigma$. □

Definition 4.8. A functor C is *matches* Φ' when the diagram below commutes

$$\begin{tikzcd}
{\rm Sign}^{\mathcal{HI}} \arrow[r, "\Phi'"] \arrow[d, "C"'] & {\rm Sign}^{FOL^{\rm pres}} \arrow[d, "U"] \\
{\rm Sign}^{FOL^{\rm pres}} \arrow[r, "U"'] & {\rm Sign}^{FOL}
\end{tikzcd}$$

where U denotes the forgetful functor.

For C matching Φ' we let

— Φ'^C denote the functor that represents the componentwise union of the corresponding presentations, i.e. $\Phi'^C(\Delta)$ is the union of $\Phi'(\Delta)$ and $C(\Delta)$, and

— $\beta'^C : \Phi'^C; {\rm Mod}^{FOL^{\rm pres}} \Rightarrow {\rm Mod}^{\mathcal{HI}}$ denotes the corresponding (componentwise) restriction of β'.

Theorem 4.1. Assume a functor C matching Φ' such that

1 For any $\mathcal{HI}$-signature $\Delta = (\Sigma, {\rm Nom}, \Lambda)$ and for any Σ-sentence ξ we have

$$\Phi'^C(\Delta) \models V(\alpha_\Sigma(\xi)). \tag{9}$$

2 Each signature morphism $(\chi : \Delta \to \Delta') \in \mathcal{D}^{\mathcal{HI}}$ with $\chi_{\rm Nom} = 1_{\rm Nom}$

— is adequate for β'^C; and

— satisfies

$$C(\Delta') \models\!\mid C(\Delta) \cup \{(\forall z_1, z_2) y(z_1) = y(z_2) \mid y \in Y\}. \tag{10}$$

(where the signature of $\Phi(\chi_{\mathrm{Sig}})$ adds the finite block of variables Y to the signature of $\Phi(\Sigma)$)

Then, for any $\Delta = (\Sigma, \mathrm{Nom}, \Lambda) \in |\mathrm{Sign}^{\mathcal{HI}}|$, $\rho \in \mathrm{Sen}^{\mathcal{HI}}(\Delta)$, $M' \in |\mathrm{Mod}^{FOL^{\mathrm{pres}}}(\Phi'^C(\Delta))|$ and $w \in M'_{\mathrm{ST}}$,

$$\beta'^C_\Delta(M') \models^w_\Delta \rho \text{ if and only if } M'^w \models_{\Phi'(\Delta)+x} \alpha'^x_\Delta(\rho), \tag{11}$$

where (like in Lemma 1) M'^w denotes the expansion of M' to $\Phi'(\Delta) + x$ defined by $M'^w_x = w$.

Proof. The proof is by induction on the structure of ρ. Let us denote $\beta'^C_\Delta(M')$ by (M, W).

1 If $\rho = i$ for some $i \in \mathrm{Nom}$:
$(M, W) \models^w_\Delta i$
iff $W_i = w$ (definition of $\models^w_\Delta$)
iff $M'_i (= M'^w_i) = M'^w_x$ (by definition of β' and of M'^w)
iff $M'^w \models_{\Phi'(\Delta)+x} i = x$ (definition of FOL^{pres}-satisfaction)
iff $M'^w \models_{\Phi'(\Delta)+x} \alpha'^x(i)$ (definition of α'^x).

2 If $\rho \in \mathrm{Sen}^{\mathcal{I}}(\Sigma)$:
$(M, W) \models^w_\Delta \rho$
iff $M_w \models^{\mathcal{I}} \rho$ (definition of $\models^w_\Delta$)
iff $\beta_\Sigma(M'|_w) \models_\Sigma \rho$ (definition of β')
iff $M'|_w \models_{\Phi(\Sigma)} \alpha_\Sigma(\rho)$ (by the satisfaction condition of (Φ, α, β))
iff $M'^w \models_{\Phi'(\Delta)+x} [\alpha_\Sigma(\rho)]^x$ (by (9) and Lemma 1)
iff $M'^w \models_{\Phi'(\Delta)+x} \alpha'^x(\rho)$ (by definition of α'^x).

3 If $\rho = \xi \vee \xi'$:
$(M, W) \models^w_\Delta \xi \vee \xi'$
iff $(M, W) \models^w_\Delta \xi$ or $(M, W) \models^w_\Delta \xi'$ (definition of $\models_\Delta$)
iff $M'^w \models_{\Phi'(\Delta)+x} \alpha'^x(\xi)$ or $M'^w \models_{\Phi(\Delta)+x} \alpha'^x(\xi')$ (by induction hypothesis)
iff $M'^w \models_{\Phi'(\Delta)+x} \alpha'^x(\xi) \vee \alpha'^x(\xi')$ (definition of $\models^{FOL^{\mathrm{pres}}}$)
iff $M'^w \models_{\Phi'(\Delta)+x} \alpha'^x(\xi \vee \xi')$ (by definition of α'^x).
The proofs for the cases when $\rho = \xi \wedge \xi'$, $\rho = \xi \Rightarrow \xi'$, $\rho = \neg\xi$, etc. are analogous.

4 If $\rho = @_i\xi$:
$(M, W) \models^w_\Delta \rho$
iff $(M, W) \models^{W_i}_\Delta \xi$ (by definition of $\models_\Delta$)
iff ${M'}^{W_i} \models_{\Phi'(\Delta)+x} \alpha'^x(\xi)$ (by induction hypothesis)
iff $M' \models_{\Phi'(\Delta)} \alpha'^i(\xi)$ (because $M'^{W_i}_x = W_i = M'_i$)
iff $M'^w \models_{\Phi'(\Delta)+x} \alpha'^i(\xi)$ (by the satisfaction condition in FOL)
iff $M'^w \models_{\Phi'(\Delta)+x} \alpha'^x(@_i\xi)$ (by definition of α'^x).

5 If $\rho = [\lambda](\xi_1, \ldots, \xi_n)$ with $\lambda \in \Lambda_{n+1}$:

$(M,W) \models^{w}_{\Delta} [\lambda](\xi_1,\ldots,\xi_n)$
iff for any $(w,w_1,\ldots,w_n) \in W_\lambda$ there exists $1 \le i \le n$ such that $(M,W) \models^{w_i} \xi_i$
(by definition of $\models_\Delta$)
iff for any $(w,w_1,\ldots,w_n) \in W_\lambda$ there exists $1 \le i \le n$ such that $M'^{w_i} \models_{\Phi'(\Delta)+y_i} \alpha'^{y_i}(\xi_i)$
(by induction hypothesis)
iff $M'^{ww_1\ldots w_n} \models_{\Phi'(\Delta)+x+y_1+\cdots+y_n} \lambda(x,y_1,\ldots,y_n) \Rightarrow \bigvee_{1\le i\le n} \alpha'^{y_i}(\xi_i)$ for all $w_1,\ldots,w_n$
iff $M'^{w} \models_{\Phi'(\Delta)+x} \forall y_1,\ldots,y_n\, \lambda(x,y_1,\ldots,y_n) \Rightarrow \bigvee_{1\le i\le n} \alpha'^{y_i}(\xi_i)$
(by the Rule of Generalization in FOL)
iff $M'^{w} \models_{\Phi'(\Delta)+x} \alpha'^{x}([\lambda](\xi_1,\ldots,\xi_n))$
(by definition of α'^{x}).

6 If $\rho = (\forall i)\xi$:
$(M,W) \models^{w}_{\Delta} (\forall i)\xi$
iff $(M,W') \models^{w}_{\Delta+i} \xi$ for each $(\mathrm{Nom}+i,\Lambda)$-expansion W' of W
iff $N^{w} \models_{\Phi'(\Delta)+i+x} \alpha'^{x}_{\Delta+i}(\xi)$ for each $(\Phi'(\Delta)+i)$-expansion N of M'
(by the induction hypothesis)
iff $M'^{w} \models_{\Phi'(\Delta)+x} (\forall i)\alpha'^{x}_{\Delta+i}(\xi)$
iff $M'^{w} \models_{\Phi'(\Delta)+x} \alpha'^{x}_{\Delta}((\forall i)\xi)$ (by the definition of α'^{x}_{Δ}).

7 If $\rho = (\forall\chi)\xi$:
Let $\chi :\ \Delta = (\Sigma,\mathrm{Nom},\Lambda) \to \Delta' = (\Sigma',\mathrm{Nom},\Lambda)$. We have to prove that

$$(M,W) \models^{w}_{\Delta} (\forall\chi)\xi \text{ if and only if } M'^{w} \models_{\Phi'(\Delta)+x} (\forall y)\big(\overline{\Gamma_{\chi_{\mathrm{Sig}}}} \cup \{D_y\} \Rightarrow \alpha'^{x}_{\Delta'}(\xi)\big)_y.$$

$\boxed{\Rightarrow}$ Let N' be any $(\Phi'(\Delta)+x)$-expansion of of M'. We have to prove that

$$N'^{w} \models_{\Phi'(\Delta)+x+y} \big(\overline{\Gamma_{\chi_{\mathrm{Sig}}}} \cup \{D_y\} \Rightarrow \alpha'^{x}_{\Delta'}(\xi)\big)_y$$

which is equivalent to proving that

$$N''^{w} \models_{\Phi'(\Delta')+x} \overline{\Gamma_{\chi_{\mathrm{Sig}}}} \cup \{D_y\} \Rightarrow \alpha'^{x}_{\Delta'}(\xi)$$

where N'' is the $\Phi'(\Delta')$-expansion of M' defined by $N''_y(z) = N'_y$. Let us assume that $N''^{w} \models \overline{\Gamma_{\chi_{\mathrm{Sig}}}} \cup \{D_y\}$. We have the following:
— $N'' \models \overline{\Gamma_{\chi_{\mathrm{Sig}}}} \cup \{D_y\}$ because $\overline{\Gamma_\chi}$ and D_y are $\Phi'(\Delta')$-sentences;
— $N'' \models \Phi'^{C}(\Delta)$ because N'' is an expansion of M' and $M' \models \Phi'^{C}(\Delta)$; and
— $N'' \models (\forall z_1,z_2) y(z_1) = y(z_2)$.

From these three satisfactions and from (10) it follows that $N'' \models \Phi'^{C}(\Delta')$. Consequently

$$\beta'^{C}_{\Delta'}(N'')\!\restriction_\chi = \beta'^{C}_{\Delta}(N''\!\restriction_{\Phi'(\chi)}) = \beta'^{C}_{\Delta}(M') = (M,W)$$

hence $\beta'^{C}_{\Delta'}(N'') \models \xi$. By the induction hypothesis it follows that $N''^{w} \models \alpha'^{x}_{\Delta'}(\xi)$.

$\boxed{\Leftarrow}$ Let (N,W) be a χ-expansion of (M,W). We have to prove that $(N,W) \models^{w}_{\Delta'} \xi$. By the adequacy hypothesis there exists a $\Phi'^{C}(\Delta')$-model N'' such that $(N,W) = \beta'^{C}_{\Delta'}(N')$ and $M' = N''\!\restriction_{\Phi'^{C}(\chi)}$. By the induction hypothesis is suffices to prove that $N''^{w} \models_{\Phi'(\Delta')+x} \alpha'^{x}_{\Delta'}(\xi)$.
We let N' be the $(\Phi'(\Delta)+y)$-expansion of M' defined by $N'_y = N''_y(z)$. This definition is

correct because $N' \models C(\Delta')$ which by (10) implies that $N''_y(z)$ is invariant with respect to z. Since N'^w is an expansion of M'^w, by hypothesis it follows that

$$N'^w \models_{\Phi'(\Delta)+y+x} (\overline{\Gamma_{\chi \mathrm{Sig}}} \cup \{D_y\} \Rightarrow \alpha'^x_{\Delta'}(\xi))_y.$$

which implies

$$N''^w \models_{\Phi'(\Delta')+x} \overline{\Gamma_{\chi \mathrm{Sig}}} \cup \{D_y\} \Rightarrow \alpha'^x_{\Delta'}(\xi).$$

The desired conclusion follows now from the fact that N'', as a $\Phi'^C(\Delta')$-model, satisfies the condition of the implication above.

□

Corollary 4.1 (Satisfaction condition for $(\Phi'^C, \alpha', \beta'^C)$). If in addition to the conditions of Thm. 4.1 above we also have that

$$\beta'_\Delta(M') \in |\mathrm{Mod}^C(\Delta)| \text{ for each } \mathcal{HI}\text{-signature } \Delta \text{ and each } M' \in |\mathrm{Mod}^{FOL^{\mathrm{pres}}}(\Phi'^C(\Delta))|$$

then $(\Phi'^C, \alpha', \beta'^C)$ is comorphism $(\mathrm{Sign}^{\mathcal{HI}}, \mathrm{Sen}^{\mathcal{HI}}, \mathrm{Mod}^C, \models) \to FOL^{\mathrm{pres}}$, i.e. for any $\Delta \in |\mathrm{Sign}^{\mathcal{HI}}|$, $\rho \in \mathrm{Sen}^{\mathcal{HI}}(\Delta)$ and $M' \in |\mathrm{Mod}^{FOL^{\mathrm{pres}}}((\Phi'^C)(\Delta))|$,

$$\beta'^C_\Delta(M') \models_\Delta \rho \text{ if and only if } M' \models_{\Phi'^C(\Delta)} \alpha'_\Delta(\rho).$$

4.3. *Examples*

Example 4.1. Let us consider the case of T hybrid propositional logic ($\mathcal{H}'PLn(T)$, see Ex. 3.1). The base comorphism (Φ, α, β) is the canonical embedding of PL into FOL determined by embedding of the PL signatures as FOL signatures. This means the D's and the Γ's are empty. The quantification space for the hybridisation consists of extensions with nominal variables. The functor C is such that each $C(P, \mathrm{Nom})$ is the presentation containing the sentence $(\forall x)\lambda(x, x)$. Note that Φ'^C maps any signature (P, Nom) to the FOL-presentation $((\{\mathrm{ST}\}, \overline{\mathrm{Nom}}, [P] + \lambda), (\forall x)\lambda(x, x))$.

The conditions (9) and (10) of Thm. 4.1 are vacuously satisfied, and so is also the adequacy condition for β'^C (of the same theorem).

Example 4.2. In the case of the encoding of $\mathcal{H}'\mathcal{H}'PL$ (from Ex. 3.2) the base comorphism is the embedding of the free hybridisation of PL into FOL; hence (see Ex. 4.1) we have

$$\Phi(P, \mathrm{Nom}^0) = (\{\mathrm{ST}^0\}, \overline{\mathrm{Nom}^0}, [P] + \lambda^0)$$

(we use λ^0 and λ^1 to distinguish the relations underlying $\Box^0$ and $\Box^1$ respectively). Thus

$$\Phi'(P, \mathrm{Nom}^0, \mathrm{Nom}^1) = (\{\mathrm{ST}^0, \mathrm{ST}^1\}, [\overline{\mathrm{Nom}^0}] + \overline{\mathrm{Nom}^1}, D_{\mathrm{ST}^0} + [[P]] + \lambda^0 + \lambda^1, D_{\overline{\mathrm{Nom}^0}}).$$

As expected we have now a sort of states for each level of hybridisation, i.e. $[\{\mathrm{ST}^0\}] = \{\mathrm{ST}^0, \mathrm{ST}^1\}$. The predicate $D_{\mathrm{ST}^0} : \ (\mathrm{ST}^1)(\mathrm{ST}^0)$ plays the role of a "sub-state-relation". A nominal i^0 of Nom^0 is interpreted as an operation $i^0 : \ \mathrm{ST}^1 \to \mathrm{ST}^0$ (of $[\overline{\mathrm{Nom}^0}]$) and similarly the base modality λ^0 by a predicate $\lambda^0 : \ (\mathrm{ST}^1)(\mathrm{ST}^0)(\mathrm{ST}^0)$. Also $D_{\overline{\mathrm{Nom}^0}} = \{(\forall y) D_{\mathrm{ST}^0}(y, i^0(y)) \mid i^0 \in \mathrm{Nom}^0\}$. In order to get the condition (9) of Thm. 4.1 fulfilled, since for any $\rho \in \mathrm{Sen}^{\mathcal{H}'PL}(P, \mathrm{Nom}^0)$ the sentence $\alpha_{(P, \mathrm{Nom}^0)}(\rho)$ is ST^0-quantified, we take

$$C(P, \mathrm{Nom}^0, \mathrm{Nom}^1) = \{(\forall x, y) D_{\mathrm{ST}^0}(x, y)\}. \tag{12}$$

Note that since $C(P, \mathrm{Nom}^0, \mathrm{Nom}^1) \models D_{\overline{[\mathrm{Nom}^0]}}$ we may define
$\Phi'^C(P, \mathrm{Nom}^0, \mathrm{Nom}^1) =$
$(\{\mathrm{ST}^0, \mathrm{ST}^1\}, \overline{[\mathrm{Nom}^0]} + \overline{\mathrm{Nom}^1}, D_{\mathrm{ST}^0} + [[P]] + \lambda^0 + \lambda^1, \{(\forall x, y)D_{\mathrm{ST}^0}(x, y)\})$.
Because of the absence of quantifications, the adequacy condition on β'^C and the condition (10) of Thm. 4.1 hold trivially.

With respect to the quantified versions of $\mathcal{H}'\mathcal{H}'PL$ the situation is as follows.

— The condition (10) of Thm. 4.1 holds trivially for the quantifications $(\forall^0 n^0)$ and $(\forall^1 n^1)$. (In the former case χ is identity and in the latter case χ_{Sig} is identity.)

— In the presence of quantifications $(\forall^1 n^0)$, χ_{Sig} adds n^0 to the $\mathcal{H}PL$ signature and consequently $\Phi(\chi_{\mathrm{Sig}})$ adds n^0 as a new constant of sort ST^0. The condition (10) of Thm. 4.1 can be fulfilled only for the constrained versions from the $\mathcal{H}'(\mathcal{H}'PL'x)y$ family. In these cases C has to reflect the sharing of sub-states domains by the sentences $\{(\forall x, y, z)D_{\mathrm{ST}^0}(x, z) \Leftrightarrow D_{\mathrm{ST}^0}(y, z)\}$ and the sharing of the interpretations of the base nominals by the sentences $\{(\forall x, y)i(x) = i(y) \mid i \in \overline{[\mathrm{Nom}^0]}\}$. Since the former are consequences of (12), we have that

$$C(P, \mathrm{Nom}^0, \mathrm{Nom}^1) = \{(\forall x, y)D_{\mathrm{ST}^0}(x, y)\} \cup \{(\forall x, y)i(x) = i(y) \mid i \in \overline{[\mathrm{Nom}^0]}\}).$$

Example 4.3. Let us consider the free hybridisation of FOL only with quantification over nominal variables ($\mathcal{H}FOL$ of Ex. 3.3). The base comorphism (Φ, α, β) is identity, hence the Γ's are empty. Hence we have that

$$\Phi'((S, F, P), \mathrm{Nom}, \Lambda) = ([S], [F] + \overline{\mathrm{Nom}}, (D_s)_{s \in S} + [P] + \overline{\Lambda}, D_F).$$

In order to get the condition (9) of Thm. 4.1 fulfilled we define

$$C((S, F, P), \mathrm{Nom}, \Lambda) = \{(\forall x, y)D_s(x, y) \mid s \in S\}.$$

Note that $C((S, F, P), \mathrm{Nom}, \Lambda) \models D_F$ hence we may write

$$\Phi'^C((S, F, P), \mathrm{Nom}, \Lambda) = ([S], [F] + \overline{\mathrm{Nom}}, (D_s)_{s \in S} + [P] + \overline{\Lambda}, \{(\forall x, y)D_s(x, y) \mid s \in S\}).$$

Because in this case we allow only quantifications with nominal variables the condition (10) of Thm. 4.1 is vacuously fulfilled and so is also the adequacy condition for β'^C (of the same theorem).

The variant of this example when the base institution is quantifier-free fragment of FOL rather that the whole of FOL, has the C's empty, and hence $\Phi'^C = \Phi'$.

The variant of the above variant that considers quantification with first order variables at the level of the hybridisation, in order to get the condition (10) of Thm. 4.1 fulfilled, requires

$$C((S, F, P), \mathrm{Nom}, \Lambda) \models \{(\forall z_1, z_2)y(z_1) = y(z_2) \mid y \in F_{\to s}, s \in S\}.$$

However because the hybridisation is free (in particular because constants are not interpreted uniformly across possible worlds) there is no way to get the adequacy condition for β'^C, hence in this case we cannot build the encoding comorphism.

Example 4.4. When encoding $\mathcal{H}REL'$ (of Ex. 3.4) the base comorphism (Φ, α, β) is the canonical embedding of REL into FOL determined by embedding of the REL signatures as FOL signatures. Hence the Γ's are empty. Thus:

$$\Phi'((C, P), \mathrm{Nom}, \Lambda) = ((\{\mathrm{ST}, \star\}, [C] + \overline{\mathrm{Nom}}, \{D_\star\} + [P] + \overline{\Lambda}), D_C).$$

The sharing of the underlying universe requires that the C's contain $(\forall x,y,z)D_\star(x,z) \Leftrightarrow D_\star(y,z)$. However in order to get the condition (9) of Thm. 4.1 fulfilled $(\forall x,y)D_\star(x,y)$ is also needed. Since the latter sentence implies the former and also implies D_C, we can do only with $(\forall x,y)D_\star(x,y)$. Finally, the sharing of the interpretations of the constants requires $\{(\forall x,y)\sigma(x)=\sigma(y) \mid \sigma \in C\}$. This also meets the requirement of condition (10) of Thm. 4.1. Hence:

$\Phi'^C((C,P),\mathrm{Nom},\Lambda) =$
$((\{\mathrm{ST},\star\},[C]+\overline{\mathrm{Nom}},\{D_\star\}+[P]+\overline{\Lambda}),\{(\forall x,y)D_\star(x,y)\}\cup\{(\forall x,y)\sigma(x)=\sigma(y)\mid\sigma\in C\})$.

It remains to check the adequacy condition for β'^C, which is a very easy enterprise. Let Δ denote the $\mathcal{H}REL$ signature $((C,P),\mathrm{Nom},\Lambda)$. For any block Y of variables for the REL signature (C,P), for any $\mathcal{H}REL'$-model (N,W) for $\Delta+Y$, and any $\Phi'^C(\Delta)$-model M' such that $(N,W)\upharpoonright_\Delta = \beta'(M')$,

$$\begin{tikzcd}
\mathrm{Mod}^{\mathcal{H}REL'}(\Delta) & \mathrm{Mod}^{FOL^{\mathrm{pres}}}(\Phi'^C(\Delta)) \arrow[l, "\beta'^C_\Delta"'] \\
\mathrm{Mod}^{\mathcal{H}REL'}(\Delta+Y) \arrow[u] & \mathrm{Mod}^{FOL^{\mathrm{pres}}}(\Phi'^C(\Delta+Y)) \arrow[l, "\beta'^C_{\Delta+Y}"] \arrow[u]
\end{tikzcd}$$

the amalgamation of M' and (N,W) is the $\Phi'^C(\Delta+Y)$-expansion N' of M' defined by $N'_y(z) = (N_s)_y \in M_s$ for any $z \in M'_{\mathrm{ST}} = |W|$ and any $s \in |W|$. This definition does not depend on s because the underlying universe and the interpretation of the constants are shared. Note also that N' satisfies indeed the sentences of $\Phi'^C(\Delta+Y)$ since by the satisfaction condition in FOL it satisfies the sentences of $\Phi'^C(\Delta)$ and it also satisfies $(\forall z_1,z_2)y(z_1)=y(z_2)$ for each $y \in Y$.

Example 4.5. In the case of the encoding of $\mathcal{H}FOLR'$ (see Ex. 3.5) the quantification space $\mathcal{D}^{\mathcal{H}FOLR}$ consists of extensions with nominal variables and rigid first-order variables. The base comorphism (Φ,α,β) is defined as follows:

1 Φ is the forgetful functor $\mathrm{Sign}^{FOLR} \to \mathrm{Sign}^{FOL}$ that maps a signature (S,S_0,F,F_0,P,P_0) to (S,F,P),
2 $\alpha_{(S,S_0,F,F_0,P,P_0)}$ is the inclusion $\mathrm{Sen}^{FOLR}(S,S_0,F,F_0,P,P_0) \subseteq \mathrm{Sen}^{FOL}(S,F,P)$ (the difference is given by the quantification which in $FOLR$ is restricted to the rigid symbols), and
3 $\beta_{(S,S_0,F,F_0,P,P_0)}$ is the identity on $\mathrm{Mod}^{FOL}(S,F,P)$.

This is a comorphism mapping signatures to signatures, hence the Γ's are empty. Thus

$$\Phi'\big((S,S_0,F,F_0,P,P_0),\mathrm{Nom},\Lambda\big) = ([S],[F]+\overline{\mathrm{Nom}},D+[P]+\overline{\Lambda},D_F)$$

The specification of the model constraints requires that $C\big((S,S_0,F,F_0,P,P_0)\big)$ contains the following sentences:

1 for each $s \in S_0$, $(\forall x,y,z)D_s(x,z) \Leftrightarrow D_s(y,z)$,
2 for each σ in F_0, $(\forall x,y,Z)\sigma(x,Z)=\sigma(y,Z)$, and
3 for each π in P_0, $(\forall x,y,Z)\pi(x,Z) \Leftrightarrow \pi(y,Z)$.

Note that these already cover the condition (10) of Thm. 4.1. For the condition (9) of Thm. 4.1 we have to add also the sentences $(\forall x,y)D_s(x,y)$ for each $s \in S_0$, which are stronger than $(\forall x,y,z)D_s(x,z) \Leftrightarrow D_s(y,z)$. All these together define the functor C that specifies the constraints. Finally, the adequacy condition for β'^{C} may be checked easily in the same way as in Ex. 4.4; therefore we omit this here.

Example 4.6. In the case of the encoding of $\mathcal{H}PAR'$ (see Ex. 3.6) the base comorphism (Φ, α, β) extends canonically the first encoding comorphism $PA \to FOL^{\mathrm{pres}}$ mentioned in Ex. 2.4 to a comorphism $PAR \to FOL^{\mathrm{pres}}$ as follows:

1 Φ maps each signature $(S, S_0, TF, TF_0, PF, PF_0)$ to the presentation

$$\big((S, TF + PF, (\mathsf{Def}_s)_{s\in S}), \Gamma_{(S,TF,PF)}\big)$$

where $\Gamma_{(S,TF,PF)}$ axiomatizes the definability of terms through the new predicates $(\mathsf{Def}_s)_{s\in S}$ as follows:

$$\begin{aligned}\Gamma_{(S,TF,PF)} = \quad & \{(\forall X)\mathsf{Def}_s(\sigma(X)) \Rightarrow \mathsf{Def}_{\underline{\mathrm{ar}}}(X) \mid \sigma \in (TF + PF)_{\underline{\mathrm{ar}}\to s}, \underline{\mathrm{ar}} \in S^*, s \in S\} \cup \\ & \{(\forall X)\mathsf{Def}_{\underline{\mathrm{ar}}}(X) \Rightarrow \mathsf{Def}_s(\sigma(X)) \mid \sigma \in TF_{\underline{\mathrm{ar}}\to s}, \underline{\mathrm{ar}} \in S^*, s \in S\}\end{aligned}$$

(where $\mathsf{Def}_{\underline{\mathrm{ar}}}(X)$ denotes $\bigwedge_{(x\,:\,s)\in X}(\mathsf{Def}_s(x))$).

2 $\alpha_{(S,S_0,TF,TF_0,PF,PF_0)}$ is recursively defined as follows:
— $\alpha(t \stackrel{\mathrm{e}}{=} t') = \mathsf{Def}_s(t) \wedge (t = t')$;
— $\alpha((\forall X)\rho) = (\forall X)(\mathsf{Def}_{\underline{\mathrm{ar}}}(X) \Rightarrow \alpha(\rho))$;
— α commutes with boolean connectives $\wedge$, $\vee$, $\Rightarrow$, etc.

3 $\beta_{(S,S_0,TF,TF_0,PF,PF_0)}$ maps any $\big((S, TF+PF, (\mathsf{Def}_s)_{s\in S}), \Gamma_{(S,TF,PF)}\big)$ model M to the partial algebra $\beta(M)$ where:
— for any $s \in S$, $\beta(M)_s = M_{\mathsf{Def}_s}$;
— for any $\sigma \in TF_{\underline{\mathrm{ar}}\to s}$, $\beta(M)_\sigma(m) = M_\sigma(m)$;
— for any $\sigma \in PF_{\underline{\mathrm{ar}}\to s}$, $\beta(M)_\sigma$ consists of the restriction of M_σ to $M_{\mathsf{Def}_{\underline{\mathrm{ar}}}}$ such that $dom(\beta(M)_\sigma) = \{x \in M_{\mathsf{Def}_{\underline{\mathrm{ar}}}} \mid M_\sigma(x) \in M_{\mathsf{Def}_s}\}$.‖

The encoding to FOL^{pres} obtained as instance of the general encoding presented above yields

$$\begin{aligned}&\Phi'\big((S, S_0, TF, TF_0, PF, PF_0), \mathrm{Nom}, \Lambda\big) = \\ &\qquad \big(([S], [TF + PF] + \overline{\mathrm{Nom}}, (D_s)_{s\in S} + [(\mathsf{Def}_s)_{s\in S}] + \overline{\Lambda}), D_{TF+PF} \cup \overline{\Gamma_{(S,TF,PF)}}\big).\end{aligned}$$

For any $\mathcal{H}PAR$ signature $((S, S_0, TF, TF_0, PF, PF_0), \mathrm{Nom}, \Lambda)$,

$$\begin{aligned}&C((S, S_0, TF, TF_0, PF, PF_0), \mathrm{Nom}, \Lambda) = \\ &\qquad \{(\forall x, z)D_s(x, z) \mid s \in S_0\} \cup \\ &\qquad \{(\forall x, y, Z)\sigma(x, Z) = \sigma(y, Z) \mid \sigma \text{ in } TF_0\} \cup \\ &\qquad \{(\forall x, y, Z)\mathsf{Def}_s(x, (\sigma(x, Z)) \Leftrightarrow \mathsf{Def}_s(y, (\sigma(y, Z)) \mid \sigma \in (PF_0)_{\underline{\mathrm{ar}}\to s}, \underline{\mathrm{ar}} \in S^*, s \in S\}\end{aligned}$$

Note that the first component in the definition of C covers both the condition (9) of Thm. 4.1 and the condition on the interpretation of the rigid sorts while the condition (10) of Thm 4.1 is entailed by the second component of C. Finally, the adequacy condition for β'^{C} may be checked easily in the same way as in Ex. 4.4; therefore we omit this here.

‖ Here $M_{\mathsf{Def}_{\underline{\mathrm{ar}}}}$ denotes $M_{\mathsf{Def}_{s_1}} \times \cdots \times M_{\mathsf{Def}_{s_n}}$ for $\underline{\mathrm{ar}} = s_1 \ldots s_n$.

5. Conservativeness

In this section we give a general method to lift the conservativity property from the base comorphism $(\Phi, \alpha, \beta) : \ \mathcal{I} \to FOL^{\mathrm{pres}}$ to the comorphism $(\Phi'^{C}, \alpha', \beta') : \ \mathcal{HI}^{C} \to FOL^{\mathrm{pres}}$. For this we assume the conditions and the notations of Thm. 4.1 above.

Proposition 5.1. Let us assume for each $\mathcal{I}$-signature Σ a mapping

$$\delta_\Sigma : \ |\mathrm{Mod}^{\mathcal{I}}(\Sigma)| \to |\mathrm{Mod}^{FOL^{\mathrm{pres}}}(\Phi(\Sigma))|$$

such that for each Σ-model A, $\beta_\Sigma(\delta_\Sigma(A)) = A$. For each $\mathcal{HI}$ signature $\Delta = (\Sigma, \mathrm{Nom}, \Lambda)$ and each model $(M, W) \in |\mathrm{Mod}^{\mathcal{HI}}(\Delta)|$ if a sort s of some variable that occurs in some quantification of some sentence in $\Phi(\Sigma)$, for any $w, w' \in |W|$, we have that

$$\delta_\Sigma(M_w)_s = \delta_\Sigma(M_{w'})_s \tag{13}$$

then there exists a $\Phi'(\Delta)$-model $\delta'_\Delta(M, W)$ such that $\beta'_\Delta(\delta'_\Delta(M, W)) = (M, W)$.

Proof. Let $\Phi(\Sigma) = ((S_\Sigma, F_\Sigma, P_\Sigma), \Gamma_\Sigma)$. We define the $\Phi'(\Delta)$-model $\delta'_\Delta(M, W) = M'$ as follows:

- $M'_{\mathrm{ST}} = |W|$,
- $M'_i = W_i$ for each $i \in \mathrm{Nom}$,
- $M'_\lambda = W_\lambda$ for each modality symbol λ in Λ,
- for each $s \in S_\Sigma$ we define $M'_s = \bigcup_{w \in |W|} \delta_\Sigma(M_w)_s$ and $M'_{D_s} = \{(w, m) \mid m \in \delta_\Sigma(M_w)_s\}$,
- for each $\sigma \in (F_\Sigma)_{\underline{\mathrm{ar}} \to v}$ we define $M'_\sigma(w, m) = \begin{cases} \delta_\Sigma(M_w)_\sigma(m), & \text{when } m \in \delta_\Sigma(M_w)_{\underline{\mathrm{ar}}}; \\ \text{any } y \in \delta_\Sigma(M_w)_v, & \text{otherwise.} \end{cases}$
  Note that the correctness of this definition relies upon our basic hypothesis that the FOL-models have non-empty carriers.
- for each π in P_Σ we define $M'_\pi = \{(w, m) \mid m \in \delta_\Sigma(M_w)_\pi\}$.

Now we have to prove that M' satisfies the sentences of $\Phi'(\Delta)$. That $M' \models D_{F_\Sigma}$ follows immediately from the definitions of M'_{D_s} and of M'_σ. Also from the hypothesis (13) we have that $M' \models V(\Gamma_\Sigma)$. For each $w \in |W|$ we let $M'|_w$ be defined like in Dfn. 4.7 and Lemma 1. Note that

$$\text{for each } w \in |W|, \ M'|_w = \delta_\Sigma(M_w). \tag{14}$$

Since $\delta_\Sigma(M_w) \models \Gamma_\Sigma$, from (14) and Lemma 1 it follows that $M'^w \models \{[\gamma]^x \mid \gamma \in \Gamma_\Sigma\}$, (where M'^w denotes the expansion of M' to the signature extended with the constant x such that $M'^w_x = w$). From the latter relation we deduce that $M' \models \overline{\Gamma_\Sigma}$.

That $\beta'(M') = (M, W)$ may be noted immediately with the help of the relation (14). $\square$

Corollary 5.1. Within the framework of Prop. 5.1, any comorphism like in Cor. 4.1 such that for each constraint model $(M, W) \in |\mathrm{Mod}^C(\Delta)|$

1 (M, W) satisfies the condition (13), and
2 $\delta'_\Delta(M, W) \models C(\Delta)$

is conservative.

Example 5.1. The encoding of T hybrid propositional logic of Ex. 4.1 is conservative according to Cor. 5.1 as follows:

- δ_Σ are identities,

– the condition (13) of Prop. 5.1 is vacuously fulfilled (the Γ's are empty), and
– obviously for each T $\mathcal{H}PL'$ model (M, W), $\delta'_\Delta(M, W) \models (\forall x)\lambda(x, x)$.

Example 5.2. Let as characterise the conservativity of encodings discussed in Ex. 4.2. For that, let us rename to δ^0 the output δ' of Prop. 5.1 applied to the encoding $\mathcal{H}'PL \to FOL$ (see Ex. 5.1)

$$\delta^0_{(P,\mathrm{Nom}^0)} : \ \mathrm{Mod}^{\mathcal{H}'PL}(P, \mathrm{Nom}^0) \to \mathrm{Mod}^{FOL}(\{\mathrm{ST}^0\}, \overline{\mathrm{Nom}^0}, [P] + \lambda^0).$$

Then by applying again Prop. 5.1 with δ^0 in the role of δ we obtain δ^1 in the role of δ':

$$\begin{array}{ll} \delta^1_{(P,\mathrm{Nom}^0,\mathrm{Nom}^1)} : & \mathrm{Mod}^{\mathcal{H}'\mathcal{H}'PL}(P, \mathrm{Nom}^0, \mathrm{Nom}^1) \to \\ & \mathrm{Mod}^{FOL}((\{\mathrm{ST}^0, \mathrm{ST}^1\}, [\overline{\mathrm{Nom}^0}] + \overline{\mathrm{Nom}^1}, D_{\mathrm{ST}^0} + [[P]] + \lambda^0 + \lambda^1)). \end{array}$$

Note however that there are models (M, W) such that $\delta'^1_\Delta(M, W) \not\models (\forall x, y)D_{\mathrm{ST}^0}(x, y)$, hence for the encoding of $\mathcal{H}'\mathcal{H}'PL$ Prop. 5.1 does not get the conservativity property. But if we consider the constrained cases $\mathcal{H}'(\mathcal{H}'PL'x)y$ with the sharing of the substates universes and of the base nominals then we get the conservativity property through Prop. 5.1 and Cor. 5.1.

Example 5.3. The encoding of the quantifier free hybridisation of FOL of Ex. 4.3 is *not* conservative. Although the condition (13) of Prop. 5.1 is vacuously fulfilled (the Γ's are empty) the example fails on the condition introduced by Cor. 5.1 since there are models (M, W) such that $\delta'_\Delta(M, W) \not\models (\forall x, y)D_s(x, y)$.

However the variant of the example that considers the quantifier free fragment of FOL as base institution is conservative because in this case the C's are empty (see Ex. 4.3) and thus the condition introduced by Cor. 5.1 is vacuously fulfilled.

Example 5.4. The encoding of the hybridisation $\mathcal{H}REL'$ of REL of Ex. 4.4 is conservative according to Cor. 5.1 as follows:
– δ_Σ are identities,
– the condition (13) of Prop. 5.1 is vacuously fulfilled (the Γ's are empty), and
– for each $\mathcal{H}REL'$ model (M, W), $\delta'_\Delta(M, W)$ satisfies $C(\Delta)$ since for all $w, w' \in |W|$ we have that

$$\delta_\Sigma(M_w)_x = (M_w)_x = (M_{w'})_x = \delta_\Sigma(M_{w'})_x$$

for each sort symbol or constant x.

Example 5.5. The encoding of the hybridisation of $FOLR$ of Ex. 4.5 is conservative according to Cor. 5.1 by arguments similar to those presented in Ex. 5.4 above.

Example 5.6. Let us show how the encoding of the hybridisation $\mathcal{H}PAR'$ of Ex. 4.6 is conservative according to Cor. 5.1. For each $\Sigma = (S_0, S, TF_0, TF, PF_0, PF)$-model M, $\delta_\Sigma(M)$ is defined as follows:
– for any $s \in S$, $\delta_\Sigma(M)_s = M_s \cup \{\perp\}$ where $\perp$ is a new element; and $\delta_\Sigma(M)_{\mathsf{Def}_s} = M_s$
– for any $\sigma \in (TF + PF)_{\underline{\mathrm{ar}} \to s}$,

$$\delta_\Sigma(M)_\sigma(m) = \begin{cases} M_\sigma(m), & \text{if } m \in \delta_\Sigma(M)_{\mathsf{Def}_{\underline{\mathrm{ar}}}} \text{ and } M_\sigma(m) \text{ is defined} \\ \perp, & \text{otherwise.} \end{cases}$$

It is easy to check that $\delta_\Sigma(M) \models \Gamma_{(S,TF,PF)}$ and that $\beta_\Sigma(\delta_\Sigma(M)) = M$.

The condition (13) of Prop. 5.1 is satisfied as follows. For each $(M, W) \in |\mathrm{Mod}^{\mathcal{H}PAR'}(\Delta)|$ and each rigid sort s (since all quantifications with first order variables are taken over rigid sorts) and any $w, w' \in |W|$ we have

$$\begin{aligned}
\delta_\Sigma(M_w)_s = &\ (M_w)_s \cup \{\perp\} && \text{(definition of } \delta) \\
= &\ (M_{w'})_s \cup \{\perp\} && ((M_w)_s = (M_{w'})_s \text{ because } s \text{ is rigid)} \\
= &\ \delta_\Sigma(M_{w'})_s && \text{(definition of } \delta).
\end{aligned}$$

The justification that for each $\mathcal{H}PAR'$ model (M, W), $\delta'_\Delta(M, W) \models C(\Delta)$ goes as follows:

1 for each $s \in S_0$, $\delta'_\Delta(M, W) \models (\forall x, z) D_s(x, z)$ means that for each $w \in |W|$ and each $m \in \delta'_\Delta(M, W)_s$ we have that $(w, m) \in \delta'_\Delta(M, W)_{D_s}$ which according the definition of $\delta'_\Delta(M, W)_{D_s}$ from the proof of Prop. 5.1 means $m \in \delta_\Sigma(M_w)_s$. But $\delta'_\Delta(M, W)_s = \delta_\Sigma(M_w)_s$ because s is rigid (which according to an argument above implies that for all $w, w' \in |W|$, $\delta_\Sigma(M_w)_s = \delta_\Sigma(M_{w'})_s$).

2 for each σ in TF_0, $\delta'_\Delta(M, W) \models (\forall x, y, Z)\sigma(x, Z) = \sigma(y, Z)$ holds because of the following facts:
- for each rigid sort s and each $w \in |W|$, $\delta'_\Delta(M, W)_s = \delta_\Sigma(M_w)_s$;
- since σ is rigid and total, for each $w, w' \in |W|$, $(M_w)_\sigma = (M_{w'})_\sigma$;
- for each $w \in |W|$, $\delta'_\Delta(M, W)_\sigma(w, m) = \delta_\Sigma(M_w)_\sigma(m)$ because $\delta'_\Delta(M, W)_s = \delta_\Sigma(M_w)_s$.

3 For all $\underline{\mathrm{ar}} \in S^*$, $s \in S$ and $\sigma \in (PF_0)_{\underline{\mathrm{ar}} \to s}$, $\delta'_\Delta(M, W) \models (\forall x, y, Z)\mathsf{Def}_s(x, (\sigma(x, Z)) \Leftrightarrow \mathsf{Def}_s(y, (\sigma(y, Z))$ means that for all $w, w' \in |W|$, $\delta'_\Delta(M, W)_\sigma(w, m) \in \delta_\Sigma(M_w)_{\mathsf{Def}_s} = (M_w)_s$ if and only if $\delta'_\Delta(M, W)_\sigma(w', m) \in \delta_\Sigma(M_{w'})_{\mathsf{Def}_s} = (M_{w'})_s$. But $(M_w)_s = (M_{w'})_s$ and $\delta'_\Delta(M, W)_\sigma(w, m) = \delta_\Sigma(M_w)_\sigma(m)$ and $\delta'_\Delta(M, W)_\sigma(w', m) = \delta_\Sigma(M_{w'})_\sigma(m)$. Thus the property is equivalent to the fact that $(M_w)_\sigma(m)$ is defined if and only if $(M_{w'})_\sigma(m)$ is defined, which holds by the rigidity of σ, i.e. $(M_w)_\sigma$ and $(M_{w'})_\sigma$ have the same domain.

6. A case study

In this section we present an example of a $\mathcal{H}PAR'$ (see Ex. 3.6) specification and a formal verification using the encoding of Ex. 4.6.

6.1. *A plastic buffer specification*

For our $\mathcal{H}PAR'$ specification, as notation, we use an extension of the language CASL (0) as follows:

— The fields **nom** and **modal** are used for the definition of the constants (denoting the nominal symbols) and predicates (denoting the modalities symbols), respectively, of the REL-part of the hybrid signature. The arities of the predicates is given by natural numbers.

— That a symbol is rigid is marked by **R** at the end of its declaration.

The case study that we adress consists of a specification of a reconfigurable data structure that may be very briefly described as follows.

A 'plastic' buffer has two distinct modes of execution: in one of them it behaves as a stack; in the other as a queue. The alternation of configurations is triggered by an event 'shift'.

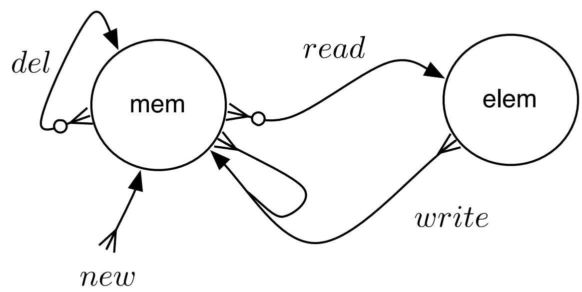

Fig. 1. Plastic Buffer PAR signature

The system has two different modes of execution denoted by the nominals *fifo* (for the queue mode) and *lifo* (for the stack mode), respectively. The modes reconfiguration is denoted by the modality symbol $shift$. These symbols make up the REL component of the hybrid signature and support the expression of the specification of the dynamics of our hybrid models.

The local behaviors of the system is specified through a PAR (which is the so-called base institution in this case; see Ex. 3.6) signature with mem denoting the sort for the stacks/queues and $elem$ for the elements of those. A total operation $write$ denotes the 'push/enqueue' operation while $read$ denotes the 'top/front' operations on stack/queues. A partial operation del denotes the 'pop' operation. Since we intend to use the same elements in both modes and moreover the buffer should contain the same data in both modes, $elem$ and mem are declared rigid. The operation $write$ is also rigid and since it is total it means it has the same effect in both modes. The operations $read$ and del play different roles in each mode of the system. However, they are also declared as rigid because they are partial and in $\mathcal{H}PAR$ this means that while their interpretation might differ according to the actual mode, their domains should not vary according to the mode. In this case both $read$ and del are defined on non-empty stacks/queues.

The PAR part of the signature is presented in Figure 6.1 in a ADJ-style diagram style, where the partiality and rigidity of operations is marked by a circle and a forked source, respectively.

nom *fifo*
lifo
modal *shift* : 1
sorts *mem* **R**
elem **R**
ops *new* : $\to$ *mem* **R**
write : *mem* $\times$ *elem* $\to$ *mem* **R**
del : *mem* $\to ?$ *mem* **R**
read : *mem* $\to ?$ *elem* **R**

The operators @ are used to express the properties that should be satisfied just in particular states of the system by considering the standard PA axiomatization of stacks and queues tagged by the respective nominals:

Ⓥ *e* : *elem*; Ⓥ *m* : *mem*;

%(stack properties)%

- $@_{lifo} del(write(m, e)) \stackrel{e}{=} m$
- $@_{lifo} read(write(m, e)) \stackrel{e}{=} e$

%(queue properties)%

- $@_{fifo} read(write(new, e)) \stackrel{e}{=} e$
- $@_{fifo} del(write(new, e)) \stackrel{e}{=} new$
- $@_{fifo}\ read(m) \stackrel{e}{=} read(m) \Rightarrow read(write(m, e)) \stackrel{e}{=} read(m)$
- $@_{fifo}\ read(m) \stackrel{e}{=} read(m) \Rightarrow del(write(m, e)) \stackrel{e}{=} write(del(m), e)$

Note that the quantification of the sentences of the specification is *local* which means that this happens at the level of the base institution PAR (hence $\textcircled{\scriptsize\forall}$; see Fact 3.2). For example the first sentence of the specification reads as $@_{lifo}(\textcircled{\scriptsize\forall}\, e)(\textcircled{\scriptsize\forall}\, m) del(write(m, e)) \stackrel{e}{=} m$.

The following axiom expresses a 'no-junk' condition, that there are no other possible worlds apart of those denoted by the nominals:

- $lifo \vee fifo$

Finally, we have to specify the dynamics of the hybrid model, i.e., the transitions:

- $@_{fifo}\langle shift\rangle lifo \wedge @_{lifo}\langle shift\rangle fifo$

A model In the following we provide an example of a model (M, W) for the above specification. The REL part of the signature is interpreted as follows:
- $|W| = \{\text{sfifo}, \text{slifo}\}$;
- $W_{lifo} = $ slifo and $W_{fifo} = $ sfifo;
- $W_{shift} = \{(\text{sfifo}, \text{slifo}), (\text{slifo}, \text{sfifo})\}$.

The buffers are lists (strings) of elements of a fixed set A, with concatenation denoted $\_.\_$. The set of the lists over A is denoted A^* and the empty list is denoted ϵ.
- $(M_{\text{sfifo}})_{elem} = (M_{\text{slifo}})_{elem} = A$;
- $(M_{\text{sfifo}})_{mem} = (M_{\text{slifo}})_{mem} = A^*$;
- $(M_{\text{slifo}})_{new} = (M_{\text{slifo}})_{new} = \epsilon$;
- $(M_{\text{sfifo}})_{write}(L, a) = (M_{\text{slifo}})_{write}(L, a) = L.a$;
- $(M_{\text{slifo}})_{del}(L) = J$ if $L = J.a$ for $J \in A^*$, $a \in A$ and is undefined otherwise;
- $(M_{\text{sfifo}})_{del}(L) = J$ if $L = a.J$ for $J \in A^*$, $a \in A$ and is undefined otherwise;
- $(M_{\text{slifo}})_{read}(L) = a$ if $L = J.a$, for $J \in A^*$, $a \in A$ and is undefined otherwise;
- $(M_{\text{sfifo}})_{read}(L) = a$ if $L = a.J$, for $J \in A^*$, $a \in A$ and is undefined otherwise;

It is not difficult to prove that (M, W) is *initial* in the class of the models of the specification that interpret the elements by A.

The following are properties of the model (M, W):

$$(\forall e, m, m')\ [shift](m' = write(e, m)) \Leftrightarrow (m' = write(e, m)). \tag{15}$$

$$(\forall e, m, m')\ \langle shift\rangle(m' = write(e, m)) \Leftrightarrow (m' = write(e, m)). \tag{16}$$

$$(\forall m, m')\big((\exists m_1)(m' = del(m_1)) \wedge [shift](m_1 = del(m))\big) \tag{17}$$

$$\Leftrightarrow$$

$$\big((\exists m_2)([shift](m' = del(m_2)) \wedge (m_2 = del(m))\big).$$

6.2. *Encoding the example in FOL*

Now we proceed with the encoding of our $\mathcal{H}PAR'$ specification into the respective FOL specification according to Ex. 4.6. For that, let start with the definition of the signature:

logic CASL.FOL

spec PLASTICBUFFERFOL =
  sorts *ST*;
  mem;
  elem
  ops *fifo* : *ST*;
  lifo : *ST*;
  new : *ST* $\rightarrow$ *mem*;
  write : *ST* $\times$ *mem* $\times$ *elem* $\rightarrow$ *mem*;
  read : *ST* $\times$ *mem* $\rightarrow$ *elem*;
  del : *ST* $\times$ *mem* $\rightarrow$ *mem*
  preds *shift* : *ST* $\times$ *ST*;
  Def_mem : *ST* $\times$ *mem*;
  Def_elem : *ST* $\times$ *elem*;
  D_mem : *ST* $\times$ *mem*;
  D_elem : *ST* $\times$ *elem*

Note that *Def* encode the partiality (coming from the encoding of PA into FOL) while *D* encode the domains of the different worlds (coming from our general encoding).

The $\overline{\Gamma_{(S,TF,PF)}}$:

$\forall$ *e* : *elem*; *w* : *ST*; *m* : *mem*
- *Def_mem*(*w*, *new*(*w*))
- *Def_mem*(*w*, *m*) $\wedge$ *Def_elem*(*w*, *e*) $\Leftrightarrow$ *Def_mem*(*w*, *write*(*w*, *m*, *e*))
- *Def_mem*(*w*, *del*(*w*, *m*)) $\Rightarrow$ *Def_mem*(*w*, *m*)
- *Def_elem*(*w*, *read*(*w*, *m*)) $\Rightarrow$ *Def_mem*(*w*, *m*)
- *D_mem*(*w*, *m*)
- *D_elem*(*w*, *e*)

The specification of the D_{TF+PF}-sentences is redundant as they are all consequences of the $V(\Gamma)$ (i.e. the previous two sentences). So, let us skip this. The same happens for the first two sentences determined by the constraint functor C. The other sentences determined by the constraint functor C are as follows:

$\forall$ *e* : *elem*; *w*, *v* : *ST*; *m* : *mem*
- *new*(*w*) = *new*(*v*)
- *write*(*w*, *m*, *e*) = *write*(*v*, *m*, *e*)
- *Def_mem*(*w*, *del*(*w*, *m*)) $\Leftrightarrow$ *Def_mem*(*v*, *del*(*v*, *m*))
- *Def_elem*(*w*, *read*(*w*, *m*)) $\Leftrightarrow$ *Def_elem*(*v*, *read*(*v*, *m*))

and finally, the translation of the specification:

$\forall$ *e* : *elem*; *m* : *mem*
- *Def_mem*(*lifo*, *m*) $\wedge$ *Def_elem*(*lifo*, *e*)

$\Rightarrow Def\_mem(lifo, m) \wedge del(lifo, write(lifo, m, e)) = m$
$\bullet\ Def\_mem(lifo, m) \wedge Def\_elem(lifo, e)$
$\Rightarrow Def\_elem(lifo, e) \wedge read(lifo, write(lifo, m, e)) = e$
$\bullet\ Def\_elem(fifo, e)$
$\Rightarrow Def\_elem(fifo, e) \wedge read(fifo, write(fifo, new(fifo), e)) = e$
$\bullet\ Def\_elem(fifo, e)$
$\Rightarrow Def\_mem(fifo, new(fifo)) \wedge del(fifo, write(fifo, new(fifo), e)) = new(fifo)$
$\bullet\ Def\_mem(fifo, m) \wedge Def\_elem(fifo, e)$
$\Rightarrow Def\_elem(fifo, read(fifo, m)) \wedge read(fifo, m) = read(fifo, m)$
$\Rightarrow Def\_elem(fifo, read(fifo, m)) \wedge read(fifo, write(fifo, m, e)) = read(fifo, m)$
$\bullet\ Def\_mem(fifo, m) \wedge Def\_elem(fifo, e)$
$\Rightarrow Def\_elem(fifo, read(fifo, m)) \wedge read(fifo, m) = read(fifo, m)$
$\Rightarrow Def\_mem(fifo, write(fifo, del(fifo, m), e)) \wedge$
$del(fifo, write(fifo, m, e)) = write(fifo, del(fifo, m), e)$
$\forall\, w : ST \bullet w = fifo \vee w = lifo$ %(no_junk)%

$\exists\, y, z : ST \bullet (shift(fifo, y) \wedge y = lifo) \wedge (shift(lifo, z) \wedge z = fifo)$

end

6.3. *Formal verification*

The formal verification of properties (15,16,17) is performed as follows:

1 We translate the properties by using the $\mathcal{H}PAR'$ instance of our general encoding (as given in Ex. 4.6);
2 By the conservativity property of the encoding of $\mathcal{H}PAR'$ (see Ex. 5.6), according to the general result of Fact 2.1 it is enough to prove that the translations of the properties are a consequence of the translation of the specification. The corresponding proofs are performed by using SPASS (0) automatic first order logic prover through the Hets system (0).

Unlike the translation of the specification, the translation of the considered properties involves the full complexity of the general translation of Dfn. 4.6 because the properties contain quantifications at the level of the hybridisation, and because the base encoding is a proper theoroidal comorphism. Moreover the equalities involved in these properties are strong rather than existence equalities, which adds a further complexity to the result of the translations. Hence the results of the translations look rather complex when compared with the inputs. For example the translation of (15) is as follows:

$\forall\, e : elem; m, m' : mem; x : ST$
$\bullet\ (\forall\, z : ST \bullet Def\_elem(z, e) \wedge Def\_mem(z, m) \wedge Def\_mem(z, m')$
$\wedge D\_elem(z, e) \wedge D\_mem(z, m) \wedge D\_mem(z, m'))$
$\Rightarrow ((\forall\, y : ST \bullet shift(x, y) \Rightarrow (Def\_mem(y, m') \wedge m' = write(y, m, e))$
$\vee (\neg Def\_mem(y, m') \wedge \neg Def\_mem(y, write(y, m, e))))$
$\Leftrightarrow (Def\_mem(x, m') \wedge m' = write(x, m, e))$
$\vee (\neg Def\_mem(x, m') \wedge \neg Def\_mem(x, write(x, m, e))))$

7. Conclusions

In this paper we have developed a hybridisation process for abstract institutions encodings into FOL expressed as theoroidal comorphisms. This provides a generic encoding of hybridised institutions into FOL, with the hybridised institutions being considered rather generally through abstract treatments of the base logic, of the constraints on the possible worlds, of the quantifiers. Moreover we have provided sufficient and pragmatic conditions for these encodings of hybridised institutions into FOL be to conservative, which implies preservation and reflection of the semantic deduction relation. Consequently formal verifications may be shifted from the level of concrete hybridised institutions (which may constitute appropriate specification logics for various kinds of dynamic systems) to FOL, with the benefit of using the rather powerful and rich theorem proving tool support available for FOL. We have illustrated this with a small case study.

This work opens up two main avenues for further research. One consists of investigations of the possibility to 'borrow' logical properties from FOL to hybridised institutions through the encoding comorphisms developed here, in the style of works such as (0; 0) etc. Important target properties would be interpolation and initial semantics, both of them relevant within the formal specification and verification contexts. The other further research avenue consists of developing tool support for formal verifications of system specifications based on hybridised institutions, especially through integration within the Hets environment (0). First important steps have already been undertaken in (0) where a hybridisation of CASL has been integrated into Hets and a generic parser (parameterised by the base institution parser) has been implemented.

Acknowledgements

We thank both Till Mossakowski and Andrzej Tarlecki for the technical suggestion of using the predicates D. The work of the first author has been supported by a grant of the Romanian National Authority for Scientific Research, CNCS-UEFISCDI, project number PN-II-ID-PCE-2011-3-0439. The work of the second author was funded by the European Regional Development Fund through the COMPETE Programme, and by the Portuguese Foundation for Science and Technology through the projects FCOMP-01-0124-FEDER-028923 and NORTE-01-0124-FEDER-000060.

References

Marc Aiguier and Răzvan Diaconescu. Stratified institutions and elementary homomorphisms. *Information Processing Letters*, 103(1):5–13, 2007.

Carlos Areces, Patrick Blackburn, and Samuel R. Delany. Bringing them all together. *Journal of Logic and Computation*, 11:657–669, 2001.

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

Jean-Yves Béziau. 13 questions about universal logic. *Bulletin of the Section of Logic*, 35(2/3):133–150, 2006.

Jean-Yves Béziau, editor. *Universal Logic: an Anthology*. Studies in Universal Logic. Springer Basel, 2012.

Patrick Blackburn. Representation, reasoning, and relational structures: a hybrid logic manifesto. *Logic Journal of IGPL*, 8(3):339–365, 2000.

Patrick Blackburn and Maarten Marx. Tableaux for quantified hybrid logic. In Uwe Egly and Christian G. Fermüller, editors, *TABLEAUX 2002*, volume 2381 of *Lecture Notes in Computer Science*, pages 38–52. Springer, 2002.

Patrick Blackburn and Jerry Seligman. Hybrid languages. *Journal of Logic, Language and Information*, 4(3):251–272, 1995.

Patrick Blackburn and Miroslava Tzakova. Hybrid languages and temporal logic. *Logic journal of IGPL*, 7(1):27–54, 1999.

Torben Braüner. Natural deduction for first-order hybrid logic. *Journal of Logic, Language and Information*, 14:173, 2005.

Torben Braüner. *Hybrid Logic and its Proof-Theory*, volume 37 of *Applied Logic Series*. Springer, 2011.

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

Walter Carnielli, Marcelo Coniglio, Dov M. Gabbay, Paula Gouveia, and Cristina Sernadas. *Analysis and Synthesis of Logics How to Cut and Paste Reasoning Systems*, volume 35 of *Applied Logic —Series*. Springer, 2008.

Maura Cerioli and José Meseguer. May I borrow your logic? (transporting logical structures along maps). *Theoretical Computer Science*, 173:311–347, 1997.

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

Răzvan Diaconescu. An encoding of partial algebras as total algebras. *Information Processing Letters*, 109(23–24):1245–1251, 2009.

Răzvan Diaconescu. Quasi-boolean encodings and conditionals in algebraic specification. *Journal of Logic and Algebraic Programming*, 79(2):174–188, 2010.

Răzvan Diaconescu. Coinduction for preordered algebras. *Information and Computation*, 209(2):108–117, 2011.

Răzvan Diaconescu. Borrowing interpolation. *Journal of Logic and Computation*, 22(3):561–586, 2012.

Răzvan Diaconescu. Interpolation for predefined types. *Mathematical Structures in Computer Science*, 22(1):1–24, 2012.

Răzvan Diaconescu. Quasi-varieties and initial semantics in hybridized institutions. *Journal of Logic and Computation*, DOI:10.1093/logcom/ext016.

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

Răzvan Diaconescu and Petros Stefaneas. Ultraproducts and possible worlds semantics in institutions. *Theoretical Computer Science*, 379(1):210–230, 2007.

Marcelo Finger and Dov M. Gabbay. Adding a temporal dimension to a logic system. *Journal of Logic, Language and Information*, 1(3):203–233, 1992.

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

Joseph Goguen and Grigore Roşu. Institution morphisms. *Formal Aspects of Computing*, 13:274–307, 2002.

Wilfrid Hodges. *A shorter Model Theory*. Cambridge University Press, 1997.

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

Alexandre Madeira. *Foundations and techniques for software reconfigurability*. PhD thesis, Universidades do Minho, Aveiro and Porto (Joint MAP-i Doctoral Programme), 2013.

Manuel A. Martins, Alexandre Madeira, Răzvan Diaconescu, and Luis Barbosa. Hybridization of institutions. In Andrea Corradini, Bartek Klin, and Corina Cîrstea, editors, *Algebra and Coalgebra in Computer Science*, volume 6859 of *Lecture Notes in Computer Science*, pages 283–297. Springer, 2011.

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

T. Mossakowski, C. Maeder, and K. Lütich. The heterogeneous tool set. In *Lecture Notes in Computer Science*, volume 4424, pages 519–522. 2007.

Till Mossakowski. Different types of arrow between logical frameworks. In F. Meyer auf der Heide and B. Monien, editors, *Proc. ICALP 96*, volume 1099 of *Lecture Notes in Computer Science*, pages 158–169. Springer Verlag, 1996.

Till Mossakowski, Răzvan Diaconescu, and Andrzej Tarlecki. What is a logic translation? *Logica Universalis*, 3(1):59–94, 2009.

Renato Neves, Alexandre Madeira, Manuel A. Martins, and Luís Soares Barbosa. Hybridisation at work. In *Algebra and Coalgebra in Computer Science*, volume 8089 of *Lecture Notes in Computer Science*, pages 340–345, 2013.

Solomon Passy and Tinko Tinchev. An essay in combinatory dynamic logic. *Information and Computation*, 93(2):263–332, 1991.

Marius Petria and Răzvan Diaconescu. Abstract Beth definability in institutions. *Journal of Symbolic Logic*, 71(3):1002–1028, 2006.

Arthur N. Prior. *Past, Present and Future*. Oxford University Press, 1967.

Donald Sannella and Andrzej Tarlecki. *Foundations of Algebraic Specifications and Formal Software Development*. Springer, 2012.

Gerhard Schurz. Combinations and completeness transfer for quantified modal logics. *Logic Journal of the IGPL*, 19(4):598–616, 2011.

Andrzej Tarlecki. Bits and pieces of the theory of institutions. In David Pitt, Samson Abramsky, Axel Poigné, and David Rydeheard, editors, *Proceedings, Summer Workshop on Category Theory and Computer Programming*, volume 240 of *Lecture Notes in Computer Science*, pages 334–360. Springer, 1986.

Andrzej Tarlecki. Moving between logical systems. In Magne Haveraaen, Olaf Owe, and Ole-Johan Dahl, editors, *Recent Trends in Data Type Specification*, volume 1130 of *Lecture Notes in Computer Science*, pages 478–502. Springer, 1996.

Andrzej Tarlecki. Towards heterogeneous specifications. In D. Gabbay and M. van Rijke, editors, *Proceedings, International Conference on Frontiers of Combining Systems (FroCoS'98)*, pages 337–360. Research Studies Press, 2000.

Johan van Bentham. *Modal Logic and Classical Logic*. Humanities Press, 1988.

Christoph Weidenbach, Uwe Brahm, Thomas Hillenbrand, Enno Keen, Christian Theobald, and Dalibor Topic. Spass version 2.0. In *Proceedings of the 18th International Conference on Automated Deduction*, CADE-18, pages 275–279, London, UK, 2002. Springer-Verlag.