

# Completeness and decidability results for hybrid(ised) logics<sup>\*</sup>

Renato Neves<sup>1</sup>, Manuel A. Martins<sup>2</sup>, and Luís S. Barbosa<sup>1</sup>

<sup>1</sup> HASLab INESC TEC & Univ. Minho, Portugal

`renato.j.neves@inesctec.pt` `luís.s.barbosa@inesctec.pt`

<sup>2</sup> CIDMA - Dep. Mathematics, Univ. Aveiro, Portugal  
`martins@ua.pt`

**Abstract.** Adding to the modal description of transition structures the ability to refer to specific states, hybrid(ised) logics provide an interesting framework for the specification of reconfigurable systems. The qualifier ‘hybrid(ised)’ refers to a generic method of developing, on top of whatever specification logic is used to model software configurations, the elements of an hybrid language, including nominals and modalities. In such a context, this paper shows how a calculus for a hybrid(ised) logic can be generated from a calculus of the base logic and that, moreover, it preserves soundness and completeness. A second contribution establishes that hybridising a decidable logic also gives rise to a decidable hybrid(ised) one. These results pave the way to the development of dedicated proof tools for such logics used in the design of reconfigurable systems.

**Keywords:** institutions, hybrid logic, decidability, completeness

## 1 Introduction

### 1.1 Motivation

The need to master ubiquitous and increasingly complex software systems, often of a safety-critical nature, has brought proof and verification to a central place in Computer Science and Software Engineering. Logics, as formal reasoning frameworks, provide tools for a rigorous specification (and analysis) of software systems, as opposed to more conventional practices in software development which are often pre-scientific and unable to prove the absence of error designs.

Ideally, the working software engineer seeks for logics that can effectively provide “yes-or-no” answers to queries regarding properties of the system (*i.e.*

---

<sup>\*</sup> Accepted authors’ manuscript published as *Completeness and Decidability Results for Hybrid(ised) Logics*, Renato Neves, Manuel A. Martins, Luís S. Barbosa, in *Lecture Notes in Computer Science*, Volume 8941, 146–161, edited by Christiano Braga, Narciso Martí-Oliet, Springer International Publishing, 2015. [DOI: 10.1007/978-3-319-15075-8\_10]. The final publication is available at Springer via [http://link.springer.com/chapter/10.1007%2F978-3-319-15075-8\\_10](http://link.springer.com/chapter/10.1007%2F978-3-319-15075-8_10).

*decidable* logics), as well as logics with a calculus providing enough syntactic rules to derive falsehood from any false statement (*i.e.* a *complete* calculus). The engineer also looks for logics with the right expressive power to specify the system at hand, a job made difficult by the complex and heterogeneous nature of current software systems which typically require a number of different logics to be suitably specified. For example, some form of equational logic may be used for data type specifications, while transitional behaviour may resort to a modal or temporal logic and fuzzy requirements may become in order to express contextual constraints. Actually, this justifies the quest for methodologies in which a specification framework can be tailored by combining whichever logics are found suitable to deal with the different nature of the requirements in presence. As Goguen and Meseguer put it in a landmark paper [11],

*“The right way to combine various programming paradigms is to discover their underlying logics, combine them, and then base a language upon the combined logic.”*

This line of research has been particularly active for the last twenty years. Finger and Gabbay, for example, showed in [9] how to add a temporal dimension to an arbitrary logic, and proved that decidability and completeness is preserved along this process. Baltazar [2] did similar work but with respect to adding a probabilistic dimension. Other, similar results include *e.g.* [6], [7], as well as a *hybridisation* method [14], in whose development the current authors have been involved, and constitutes the starting point of the work reported in the sequel.

## 1.2 Context

Essentially hybridisation turns a given logic, defined as an *institution*, into a hybrid logic, a brand of modal logics that adds to the modal description of transition structures the ability to refer to specific states (*cf.* [1, 3]). This paves the way to an expressive framework, proposed in [13], for the specification of *reconfigurable* systems, *i.e.*, systems which may evolve through different execution modes, or configurations, along their lifetime. Specification proceeds in two steps:

- *globally* the system’s dynamics is represented by a transition structure described in a hybrid language, whose states correspond to possible configurations;
- *locally* each state is endowed with a structure modelling the specification of the associated configuration.

The logic used locally, *i.e.* the one to be hybridised, depends on the application requirements. Typical candidates are equational, partial algebra or first-order logic (FOL), but one may equally resort to multivalued logics or even to hybrid logic itself equipping, in the last case, each state with another (local) transition system. Verification resorts to a parametrised translation to FOL (developed in [14] and [15]), but at the cost of losing decidability and adding extra complexity.

The generic character of this hybridisation process is achieved through its rendering in the context of institution theory [10]. Such a theory formalises the essence of what a logical system actually is, by encompassing syntax, semantics and satisfaction. However, its classical definition, the one in which the hybridisation method is based, does not include an abstract structure to represent a logic calculus. The problem was addressed in [8] with the introduction of  $\pi$ -institutions, and, more recently, in [5] with the notion of an *institution with proofs*, a more general version of the previous work.

### 1.3 Contributions and roadmap

This paper starts by recasting the hybridisation method in the theory of institutions with proofs, which makes possible the systematic generation of a calculus when hybridising a given logic.

Then, we prove that, under certain conditions, this method preserves decidability, and furthermore that the generated calculus is sound and complete whenever the one corresponding to the base logic is. Those are the paper's main contributions. Besides their theoretical relevance, from a pragmatic point of view they pave the way to the development of effective verification algorithms.

The paper is organised as follows. Institutions with proofs are briefly reviewed in Section 2. Then, Section 3 introduces the generation of an hybrid calculus from a base one. Section 4 establishes decidability and completeness. Finally, Section 5 concludes the paper and hints at future lines of research.

## 2 Background

We first recall the notion of an institution [10]. As already mentioned, it formalises the essence of a logical system, encompassing syntax, semantics and satisfaction. Put forward by J. Goguen and R. Burstall in the late seventies, its original aim was to develop as much as Computer 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, with the theory of institutions becoming the most fundamental mathematical theory underlying algebraic specification methods, and also increasingly used in other areas of Computer Science. Formally,

**Definition 1.** *An institution is a tuple  $(\text{Sign}^I, \text{Sen}^I, \text{Mod}^I, (\models_\Sigma^I)_{\Sigma \in |\text{Sign}^I|})$ , where:*

- $\text{Sign}^I$  is a category whose objects are signatures and arrows signature morphisms,
- $\text{Sen}^I : \text{Sign}^I \rightarrow \text{Set}$ , is a functor that, for each signature  $\Sigma \in |\text{Sign}^I|$ , returns a set of sentences over  $\Sigma$ ,
- $\text{Mod}^I : (\text{Sign}^I)^{\text{op}} \rightarrow \text{Cat}$ , is a functor that, for each signature  $\Sigma \in |\text{Sign}^I|$ , returns a category whose objects are models over  $\Sigma$ ,

- $\models_{\Sigma}^I \subseteq |Mod^I(\Sigma)| \times Sen^I(\Sigma)$ , or simply  $\models$ , if the context is clear, is a satisfaction relation such that, for each signature morphism  $\varphi : \Sigma \rightarrow \Sigma'$ ,

$Mod^I(\varphi)(M') \models_{\Sigma}^I \rho$  iff  $M' \models_{\Sigma'}^I Sen^I(\varphi)(\rho)$ , for any  $M' \in |Mod^I(\Sigma')|$  and  $\rho \in Sen^I(\Sigma)$ . Graphically,

$$\begin{array}{ccccc}
 \Sigma & & Mod^I(\Sigma) & \xrightarrow{\models_{\Sigma}^I} & Sen^I(\Sigma) \\
 \downarrow \varphi & & \uparrow Mod^I(\varphi) & & \downarrow Sen^I(\varphi) \\
 \Sigma' & & Mod^I(\Sigma') & \xrightarrow{\models_{\Sigma'}^I} & Sen^I(\Sigma')
 \end{array}$$

Intuitively, this property means that satisfaction is preserved under change of notation.

**Definition 2.** Consider an institution  $I$  and signature  $\Sigma \in |Sign^I|$ . We say that a sentence  $\rho \in Sen^I(\Sigma)$  is  $\Sigma$ -valid (or simply, valid) if for each model  $M \in |Mod^I(\Sigma)|$ ,  $M \models_{\Sigma}^I \rho$ . Usually we prefix such sentences by  $\models_{\Sigma}^I$  or, simply by  $\models^I$  or just  $\models$ .

**Definition 3.** An institution  $I$  has the negation property if, for any signature  $\Sigma \in |Sign^I|$  and sentence  $\rho \in Sen^I(\Sigma)$ , there is a sentence,  $\neg\rho \in Sen^I(\Sigma)$ , such that for any model  $M \in |Mod^I(\Sigma)|$ ,  $M \models_{\Sigma}^I \rho$  iff  $M \not\models_{\Sigma}^I \neg\rho$ .

If this property holds, satisfiability of sentences may be rephrased as follows,

**Definition 4.** Consider institution  $I$  with the negation property and a signature  $\Sigma \in |Sign^I|$ . For any sentence  $\rho \in Sen^I(\Sigma)$ ,

$\rho$  is  $\Sigma$ -unsatisfiable iff  $\neg\rho$  is  $\Sigma$ -valid.

Similarly,

**Definition 5.** An institution  $I$  has the explicit satisfaction property, if for any signature  $\Sigma \in |Sign^I|$  and sentence  $\rho \in Sen^I(\Sigma)$ , satisfiability of  $\rho$  entails the existence of a model  $M \in |Mod^I(\Sigma)|$  such that  $M \models_{\Sigma}^I \rho$ .

Note that this last property holds in the most common logics used in specification, e.g., propositional, fuzzy, equational, partial and first-order.

**Definition 6.** An institution  $I$  has the conjunction property if, for any signature  $\Sigma \in |Sign^I|$  and sentences  $\rho, \rho' \in Sen^I(\Sigma)$ , there is sentence  $\rho \wedge \rho' \in Sen^I(\Sigma)$ , such that for any model  $M \in |Mod^I(\Sigma)|$ ,  $M \models_{\Sigma}^I \rho \wedge \rho'$  iff  $M \models_{\Sigma}^I \rho$  and  $M \models_{\Sigma}^I \rho'$

Note that with the conjunction property we are able to define a sentence  $(\rho \wedge \neg\rho) \in Sen^I(\Sigma)$ , denoted by  $\perp$ , that is not satisfied by any model of  $|Mod^I(\Sigma)|$ .

An institution for which both the negation and conjunction properties hold, is said to have the typical boolean connectives.

In order to better grasp this rather abstract concept of an institution let us analyse some typical examples.

*Example 1.* Many sorted first order logic (FOL)

- SIGNATURES.  $Sign^{FOL}$  is a category whose objects are triples  $(S, F, P)$ , consisting of a set of sort symbols  $S$ , a family,  $F = (F_{w \rightarrow s})_{w \in S^*, s \in S}$ , of function symbols indexed by their arity, and a family,  $P = (P_w)_{w \in S^*}$ , of relational symbols also indexed by their arity.  
A signature morphism in this category is a triple  $(\varphi_{st}, \varphi_{op}, \varphi_{rl}) : (S, F, P) \rightarrow (S', F', P')$  such that if  $\sigma \in F_{w \rightarrow s}$ , then  $\varphi_{op}(\sigma) \in F'_{\varphi_{st}(w) \rightarrow \varphi_{st}(s)}$ , and if  $\pi \in P_w$  then  $\varphi_{rl}(\pi) \in P'_{\varphi_{st}(w)}$ .
- SENTENCES. For each signature object  $(S, F, P) \in |Sign^{FOL}|$ ,  $Sen^{FOL}(S, F, P)$  is the smallest set generated by:

$$\rho \ni \neg \rho \mid \rho \wedge \rho \mid t = t \mid \pi(X) \mid \forall x : s . \rho'$$

where  $t$  is a term of sorts with the syntactic structure  $\sigma(X)$  for  $\sigma \in F_{w \rightarrow s}$  and  $X$  a list of terms compatible with the arity of  $\sigma$ .  $\pi \in P_w$  and  $X$  is a list of terms compatible with the arity of  $\pi$ . Finally,  $\rho' \in Sen^{FOL}(S, F \uplus \{x\}_{\rightarrow s}, P)$ .  $Sen^I(\varphi)$ , for  $\varphi$  a signature morphism, is a function that, given a sentence  $\rho \in Sen^I(S, F, P)$ , replaces the signature symbols in  $\rho$  under the mapping corresponding to  $\varphi$ .

- MODELS. For each signature  $(S, F, P) \in |Sign^{FOL}|$ ,  $Mod^{FOL}(S, F, P)$  is the category with only identity arrows and whose objects are models with a carrier set  $|M_s|$ , for each  $s \in S$ ; a function  $M_\sigma : |M_w| \rightarrow |M_s|$ , for each  $\sigma_{w \rightarrow s} \in F_{w \rightarrow s}$ ; a relation  $M_\pi \subseteq |M_w|$ , for each  $\pi \in P_w$ .
- SATISFACTION. Satisfaction of sentences by models is the usual Tarskian satisfaction.

*Example 2.* Equational logic (EQ)

The institution EQ is the sub-institution of FOL in which sentences are restricted to those of the type  $\forall \vec{x} : \vec{s} . t = t'$

*Example 3.* Propositional logic (PL)

Institution PL is the sub-institution of FOL in which signatures with no empty set of sorts are discarded.

As seen above, no notion of a proof system is considered in the definition of an institution. This is a limitation if one is interested in logical systems with calculi, as is the case in this paper which aims at introducing the systematic generation of calculi for hybridised logics. To overcome this we resort to the following extended definition of an institution with proofs [5].

**Definition 7.** *An institution with proofs adds to the original definition a functor  $Prf^I : Sign^I \rightarrow Cat$  such that, for each  $\Sigma \in |Sign^I|$ ,  $Prf(\Sigma)$  (called the category of  $\Sigma$ -proofs) has subsets of  $Sen^I(\Sigma)$  (i.e.,  $|Prf(\Sigma)| = |\mathcal{P}(Sen^I(\Sigma))|$ ) as objects, and the corresponding proofs as arrows. The latter are preserved along signature morphisms. In addition, for  $A, B \in |Prf^I(\Sigma)|$ , if  $A \subseteq B$  then there is an arrow  $B \longrightarrow A$ ; if  $A \cap B = \emptyset$  and there is  $\Gamma \in |Prf^I(\Sigma)|$  such that*

$p : \Gamma \longrightarrow A$  and  $q : \Gamma \longrightarrow B$ , then there is a unique proof  $\langle p, q \rangle$  making the following diagram to commute

$$\begin{array}{ccccc} & & A & \xleftarrow{i_1} & (A \uplus B) & \xrightarrow{i_2} & B \\ & & \swarrow p & & \uparrow \langle p, q \rangle & & \searrow q \\ & & \Gamma & & & & \end{array}$$

For the sake of simplicity, when a singleton set of sentences is presented in a proof arrow, we may drop the curly brackets. Note that the restrictions imposed to the proof arrows oblige  $Prf^I$  to follow the basic properties of a proof system. In particular, we have

1. *Reflexivity* (if  $A \in \Gamma$ , then  $\Gamma \vdash A$ ) follows from the fact that  $\{A\} \subseteq \Gamma$  and therefore  $\Gamma \longrightarrow A$ .
2. *Monotonicity* (if  $\Gamma \vdash A$  and  $\Gamma \subseteq \Delta$  then  $\Delta \vdash A$ ), follows from composition of proofs, where  $\Delta \longrightarrow \Gamma$  is given by inclusion and  $\Gamma \longrightarrow A$  by the assumption.
3. *Transitivity* (if  $\Gamma \vdash A$  and  $\{\Delta, A\} \vdash B$  then  $\Gamma \cup \Delta \vdash B$ ), follows from the product of disjoint sets, reflexivity and monotonicity,

$$\begin{array}{ccccccc} & & \Gamma & \longrightarrow & A & \longrightarrow & A' \\ & \nearrow & & & & & \uparrow \\ (\Gamma \cup \Delta) & \cdots \longrightarrow & \Delta \uplus A' & \longrightarrow & (\Delta \cup A) & \longrightarrow & B \\ & \searrow & & & \downarrow & & \\ & & \Delta & & & & \end{array}$$

where  $A' = A - (A \cap \Delta)$  ( $A' \subseteq A$  and  $(\Delta \cup A) \subseteq (\Delta \cup A')$ ).

Note that functor  $Prf^I$  distinguishes different proofs between the same pair of objects, as opposed to entailment systems<sup>3</sup>. In this work, however, we restrict ourselves to entailment systems in which  $Prf^I(\Sigma)$  has at most one arrow for each pair of objects, *i.e.* that  $Prf^I(\Sigma)$  is *thin*. Such restriction makes showing the uniqueness of  $\langle p, q \rangle$  trivial.

**Definition 8.** Let  $I$  be an institution with proof system  $Prf^I$ . We say that  $Prf^I$  is sound if, for any signature  $\Sigma \in |Sign^I|$  and sentence  $\rho \in Sen^I(\Sigma)$ ,

$$\text{if arrow } \emptyset \longrightarrow \rho \text{ is in } Prf^I(\Sigma) \text{ then } \models^I \rho.$$

**Definition 9.** Let  $I$  be an institution with proof system  $Prf^I$ . We say that  $Prf^I$  is complete if, for any signature  $\Sigma \in |Sign^I|$  and sentence  $\rho \in Sen^I(\Sigma)$ ,

$$\text{if } \models^I \rho \text{ then arrow } \emptyset \longrightarrow \rho \text{ is in } Prf^I(\Sigma)$$

---

<sup>3</sup> Typically, in an entailment system  $\Gamma \vdash A$  means that  $\Gamma$  derives (or entails)  $A$ .

Hence, soundness and completeness of  $Prf^I$  entails the equivalence, for any signature  $\Sigma \in |Sign^I|$  and sentence  $\rho \in Sen^I(\Sigma)$ ,

$$\models^I \rho \text{ iff } \emptyset \longrightarrow \rho \text{ is in } Prf^I(\Sigma)$$

We can now show that

**Theorem 1.** *If an institution  $I$  has classical boolean connectives, and a sound and complete calculus  $Prf^I$ , with the reductio ad absurdum property, then, for any signature,  $\Sigma \in |Sign^I|$ , and sentence,  $\rho \in Sen^I(\Sigma)$ ,*

$$\rho \text{ is satisfiable iff } \rho \longrightarrow \perp \text{ is not in } Prf^I(\Sigma)$$

*Proof.*

$$\begin{aligned} & \models^I \rho \text{ iff } \emptyset \longrightarrow \rho \text{ is in } Prf^I(\Sigma) \\ \Leftrightarrow & \quad \{ \text{ defn. of satisfiability } \} \\ & \neg \rho \text{ is unsat iff } \emptyset \longrightarrow \rho \text{ is in } Prf^I(\Sigma) \\ \Leftrightarrow & \quad \{ \text{ soundness, completeness of } Prf^I(\Sigma) \text{ and } r.a.a \} \\ & \neg \rho \text{ is unsat iff } \neg \rho \longrightarrow \perp \text{ is in } Prf^I(\Sigma) \\ \Leftrightarrow & \quad \{ \text{ defn. of negation } \} \\ & \rho \text{ is unsat iff } \rho \longrightarrow \perp \text{ is in } Prf^I(\Sigma) \\ \Leftrightarrow & \quad \{ \text{ de Morgan's law } \} \\ & \rho \text{ is sat iff } \rho \longrightarrow \perp \text{ is not in } Prf^I(\Sigma) \end{aligned}$$

**Corollary 1.** *In the context of theorem 1, if  $I$  has the explicit satisfaction property, then*

$$\begin{aligned} & \rho \text{ is sat iff } \rho \longrightarrow \perp \text{ is not in } Prf^I(\Sigma) \\ \Leftrightarrow & \quad \{ \text{ explicit satisfaction property } \} \\ & \rho \text{ has a model iff } \rho \longrightarrow \perp \text{ is not in } Prf^I(\Sigma) \end{aligned}$$

This last result will be essential in the sequel for proving completeness of hybridised logics.

### 3 Hybridisation of logics and their calculi

As mentioned before, the existence of software products that are built and maintained with respect to requirements of different nature calls for techniques that

favour combination of logics. Hybridisation [14] was born in this context. It aims at providing a framework to specify reconfigurable systems, whose execution modes are described by whatever logic the engineer finds suitable, whereas the transition structure is expressed in a hybrid language.

From a point of view of verification, however, the engineer is not only interested in having a hybridised logic, but also, in a very pragmatic way, in its calculus. This section addresses such issue. It starts by revisiting hybridisation and then, through the notion of institutions with proofs, it shows how to lift the calculus in the base logic to its hybridised counterpart.

### 3.1 Hybridisation revisited

**Definition 10.** *The category  $Sign^{\mathcal{H}}$  is the category  $Set \times Set$  whose objects are pairs  $(Nom, \Lambda)$ , with  $Nom$  denoting a set of nominal symbols and  $\Lambda$  a set of modality symbols.*

**Definition 11.** *Given an institution  $I = (Sign^I, Sen^I, Mod^I, \models^I)$  its hybridised version  $\mathcal{H}I = (Sign^{\mathcal{H}I}, Sen^{\mathcal{H}I}, Mod^{\mathcal{H}I}, \models^{\mathcal{H}I})$  is defined as follows,*

- $Sign^{\mathcal{H}I} = Sign^{\mathcal{H}} \times Sign^{\mathcal{I}}$ ,
- given a signature  $(\Delta, \Sigma) \in |Sign^{\mathcal{H}I}|$ ,  $Sen^{\mathcal{H}I}(\Delta, \Sigma)$  is the least set generated by

$$\rho \ni \neg \rho \mid \rho \mathbb{A} \rho \mid i \mid @_i \rho \mid \langle \lambda \rangle \rho \mid \forall x \rho' \mid \psi \mid A \rho$$

for  $i$  a nominal,  $\lambda$  a modality,  $\psi \in Sen^{\mathcal{I}}(\Sigma)$  and  $\rho' \in Sen^{\mathcal{H}I}(\Delta \uplus \{x\}, \Sigma)$  where  $x$  is a nominal. We use non standard boolean connectives  $(\neg, \mathbb{A})^4$  in order to distinguish them from the boolean connectives that the base logic may have.

- given a signature  $(\Delta, \Sigma) \in |Sign^{\mathcal{H}I}|$ , a model  $M \in |Mod^{\mathcal{H}I}(\Delta, \Sigma)|$  is a triple  $(W, R, m)$  such that,
  - $W$  is a non-empty set of worlds,
  - $R$  is a family of relational symbols indexed by the modality symbols, such that for each  $\lambda \in \Lambda$  ( where  $\Delta = (-, \Lambda)$  ),  $R_\lambda \subseteq W \times W$ ,
  - and  $m : W \rightarrow |Mod^I(\Sigma)|$ ,
and for each  $i \in Nom$ ,  $(W, R, m)_i$  is interpreted as a world in  $W$ .
- given a signature  $(\Delta, \Sigma) \in |Sign^I|$ , a model  $M = (W, R, m) \in |Mod^I(\Delta, \Sigma)|$  and a sentence  $\rho \in Sen^{\mathcal{H}I}(\Delta, \Sigma)$ , the satisfaction relation is defined as,

$$M \models_{(\Delta, \Sigma)}^{\mathcal{H}I} \rho \text{ iff } M \models^w \rho, \text{ for all } w \in W$$

where,

---

<sup>4</sup> implication ( $\Rightarrow$ ) and biimplication ( $\Leftrightarrow$ ) are built in the usual way.



$$\begin{aligned}
M \models^w \neg \rho & \text{ iff } M \not\models^w \rho \\
M \models^w \rho \mathbin{\wedge} \rho' & \text{ iff } M \models^w \rho \text{ and } M \models^w \rho' \\
M \models^w i & \text{ iff } M_i = w \\
M \models^w @_i \rho & \text{ iff } M \models^{M_i} \rho \\
M \models^w \langle \lambda \rangle \rho & \text{ iff there is some } w' \in W \text{ such that } (w, w') \in R_\lambda \text{ and } M \models^{w'} \rho \\
M \models^w A \rho & \text{ iff } M \models^w \forall x @_x \rho \\
M \models^w \psi & \text{ iff } m(w) \models_\Sigma^I \psi \\
M \models^w \forall x \rho & \text{ iff for all } M', M' \models \rho
\end{aligned}$$

for  $(W, R, m) = M' \in |Mod^{\mathcal{H}I}(\Delta \uplus \{x\}, \Sigma)|$  a model expansion of  $M$ , with the only difference between them being the interpretation of nominal  $x$ : while it is defined in  $M'$ , in  $M$  it is not.

Note that sentence  $\rho$  being satisfiable means that there is a model  $(W, R, m) = M \in |Mod^{\mathcal{H}I}(\Delta, \Sigma)|$  such that  $M \models^w \rho$  for some  $w \in W$ . Hence, hybridised logics do not have the explicit satisfaction property. One can, however, redefine the satisfaction relation in the hybridisation method to,

$$M \models_{(\Delta, \Sigma)}^{\mathcal{H}I} \rho \text{ iff } M \models^w \rho, \text{ for some } w \in W$$

which then provides to logics hybridised in this alternative way the explicit satisfaction property.

A weak hybridisation of an institution  $I$ , denoted by  $\mathcal{H}I$ , is obtained as  $\mathcal{H}I$ , but the omission of syntax constructor  $\forall x \rho$ . The following decidability results are formulated with respect to weak hybridisation.

### 3.2 Hybridising a calculus

We now present the hybridisation of calculi in the context of institutions with proofs. Let us assume that  $I$  has a proof system, *i.e.*, that  $Prf^I$  is well defined, and that, in particular, it is an entailment system, *i.e.*,  $Prf^I$  only defines thin categories. Then we define  $Prf^{\mathcal{H}I}$  as follows:

For any  $((Nom, A), \Sigma) \in |Sign^{\mathcal{H}I}|$ ,

1. for any  $\rho \in Sen^I(\Sigma)$ , if  $\emptyset \longrightarrow \rho$  is in  $Prf^I(\Sigma)$  then  $\emptyset \longrightarrow \rho$  is in  $Prf^{\mathcal{H}I}((Nom, A), \Sigma)$ ,
2. for any nominal  $i, j \in Nom$ , modality  $\lambda \in A$ ,  $\rho, \rho' \in Sen^{\mathcal{H}I}((Nom, A), \Sigma)$ , proof arrows in Table 1 are in  $Prf^{\mathcal{H}I}((Nom, A), \Sigma)$
3. finally,  $Prf^{\mathcal{H}I}((Nom, A), \Sigma)$  has all the inclusion proof arrows and for each  $A, B, \Gamma \in |Prf^{\mathcal{H}I}((Nom, A), \Sigma)|$  if  $\Gamma \longrightarrow A$ ,  $\Gamma \longrightarrow B$  then  $\Gamma \longrightarrow A \cup B$ .

$Prf^{\mathcal{H}I}$  is maintained thin in its construction process in order to have it as an entailment system.

### Axioms

- (CT) All substitution instances of classical tautologies
- (Dist)  $\emptyset \longrightarrow @_i(\rho \Rightarrow \rho') \Leftrightarrow (@_i\rho \Rightarrow @_i\rho')$
- ( $\perp$ )  $\emptyset \longrightarrow @_i\perp \Rightarrow \perp$
- (Scope)  $\emptyset \longrightarrow @_i@_j\rho \Rightarrow @_j\rho$
- (Ref)  $\emptyset \longrightarrow @_i i$
- (Intro)  $\emptyset \longrightarrow (i \mathbin{\&\&} \rho) \Rightarrow @_i\rho$
- ( $\Box E$ )  $\emptyset \longrightarrow ([\lambda]\rho \mathbin{\&\&} \langle \lambda \rangle i) \Rightarrow @_i\rho$
- ( $\forall E$ )  $\emptyset \longrightarrow \forall x \rho \Rightarrow \rho[i/x]$

### Rules

- (MP) if  $\emptyset \longrightarrow \rho$  and  $\rho \longrightarrow \rho'$  then  $\emptyset \longrightarrow \rho'$
- ( $N_{@}$ ) if  $\emptyset \longrightarrow \rho$  then  $\emptyset \longrightarrow @_i\rho$
- (Name) if  $i$  does not occur free in  $\rho$  and  $\emptyset \longrightarrow @_i\rho$  then  $\emptyset \longrightarrow \rho$
- ( $\Box I$ ) if  $i$  does not occur free in  $\rho, \rho'$  and  $\emptyset \longrightarrow (\rho \wedge \langle \lambda \rangle i) \Rightarrow @_i\rho'$   
then  $\emptyset \longrightarrow \rho \Rightarrow [\lambda]\rho'$
- ( $\forall I$ ) if  $i$  does not occur free in  $\forall x \rho', \rho$  and  $\emptyset \longrightarrow \rho \Rightarrow \rho'[i/x]$   
then  $\emptyset \longrightarrow \rho \Rightarrow \forall x \rho'$

**Table 1.** Axioms and rules for  $Prf^{\mathcal{H}I}$  from [3]

## 4 Decidability and completeness of hybridised logics

Decidability and completeness are properties that one usually looks for when defining a new logic. From a Computer Science perspective, they are essential as a basis for tool-supported proofs. Formally,

**Definition 12.** *Decidability of an institution  $I$  means that, for each signature  $\Sigma \in |\text{Sign}^I|$  and sentence  $\rho \in \text{Sen}^I(\Sigma)$ , there is an effective algorithm able to decide whether  $\rho$  is valid.*

After some preliminary work, we address first this definition in the context of hybridised logics.

### 4.1 Preliminaries

Recall that in the sequel we assume that the base institution  $I$  has the classical boolean connectives and the explicit satisfaction property. Furthermore, it has a calculus,  $Prf^I$ , is sound, complete and has the *reductio ad absurdum* property.

**Notation 1.** *Consider  $(\Delta, \Sigma) \in |\text{Sign}^{\mathcal{H}I}|$  and  $\rho \in \text{Sen}^{\mathcal{H}I}(\Delta, \Sigma)$ . Let  $B_\rho = \{\psi_1, \dots, \psi_n\}$  to denote the set of all maximal sentences,  $\psi_i \in \text{Sen}^I(\Sigma)$ , occurring in  $\rho$ . Then, the set of base sentences,  $\Omega_\rho$ , denotes the least set such that for each  $a \in 2^{B_\rho}$ ,*

$$(\chi_1 \wedge \dots \wedge \chi_n) \in \Omega_\rho \subseteq \text{Sen}^I(\Sigma)$$

where

$$\chi_i = \begin{cases} \psi_i & \text{if } \psi_i \in a \\ \neg\psi_i & \text{if } \psi_i \notin a \end{cases}$$

Whenever suitable we abbreviate  $(\chi_1 \wedge \dots \wedge \chi_n)$  to  $\chi$ , and refer to components of  $\chi$  as  $\chi_i$ .

**Lemma 1.** *For any model  $M \in |\text{Mod}^I(\Sigma)|$ ,  $M$  satisfies exactly one of the sentences in  $\Omega_\rho$ .*

*Proof.* Suppose that  $M$  fails to satisfy a sentence  $\chi \in \Omega_\rho$ . This only happens when at least one member of  $\chi$  is not satisfied by  $M$ . By definition of  $\Omega_\rho$  we know that  $\Omega_\rho$  has another sentence  $\chi'$  which negates all the failed components in  $\chi$  and therefore  $M$  must satisfy  $\chi'$ .

Suppose that  $M$  satisfies a sentence  $\chi \in \Omega_\rho$ . Clearly, by the definition of  $\Omega_\rho$  any other sentence  $\chi' \in \Omega_\rho$  must negate at least one of the components of  $\chi$ . Since  $M$  cannot satisfy a component and its negation,  $\chi'$  cannot be satisfied by  $M$ .

**Notation 2.** *If  $\Omega_\rho$  is not empty, Lemma 1 allows the use of notation  $\Omega_\rho^M$  to denote the sentence in  $\Omega_\rho$  which is satisfied by a model  $M \in |\text{Mod}^I(\Sigma)|$ .*

Next, in order to take advantage of the well known decidability and completeness results for hybrid propositional logic,  $\mathcal{HPL}$ , we define a function between  $\mathcal{HI}$  and  $\mathcal{HPL}$  sentences,

**Definition 13.** *Consider a signature  $(\Delta, \Sigma) \in |\text{Sen}^{\mathcal{HI}}|$ , a sentence  $\rho \in \text{Sen}^{\mathcal{HI}}(\Delta, \Sigma)$ , and a PL signature  $\text{Prop}$  that, for each  $\psi_i \in \text{Sen}^I(\Sigma)$ , has a propositional symbol  $\pi_{\psi_i}$ . Then a function  $\sigma : \text{Sen}^{\mathcal{HI}}(\Delta, \Sigma) \rightarrow \text{Sen}^{\mathcal{HPL}}(\Delta, \text{Prop})$  is defined to replace the base sentences that occur in  $\rho$  and  $B_\rho$  by propositions from  $\text{Prop}$ . Formally,*

$$\begin{aligned} \sigma(\neg \rho) &= \neg \sigma(\rho) \\ \sigma(\rho \mathbin{\&\&} \rho') &= \sigma(\rho) \mathbin{\&\&} \sigma(\rho') \\ \sigma(i) &= i \\ \sigma(@_i \rho) &= @_i \sigma(\rho) \\ \sigma(\langle \lambda \rangle \rho) &= \langle \lambda \rangle \sigma(\rho) \\ \sigma(\forall x \rho) &= \forall x \sigma(\rho) \\ \sigma(A \rho) &= A \sigma(\rho) \\ \sigma(\psi_i) &= \pi_{\psi_i} \end{aligned}$$

**Definition 14.** *For each  $\chi \in \Omega_\rho$  we define function  $\sigma' : \chi \rightarrow \text{Sen}^{\text{PL}}(\text{Prop})$  such that,*

$$\sigma'(\chi_i) = \begin{cases} \neg \pi_{\psi_i} & \text{if } \chi_i = \neg \psi_i \\ \pi_{\psi_i} & \text{if } \chi_i = \psi_i \end{cases}$$

*and denote by  $\sigma'[\chi]$  the result of applying  $\sigma'$  to each member of  $\chi$ .*

Note that both  $\sigma$  and  $\sigma'$  are injective.

## 4.2 Decidability

**Lemma 2.** *Consider a signature  $(\Delta, \Sigma) \in |\text{Sign}^{\mathcal{HI}}|$  and  $\rho \in \text{Sen}^{\mathcal{HI}}(\Delta, \Sigma)$ . For any  $\chi \in \Omega_\rho$ , if  $\chi$  is satisfiable  $\sigma'[\chi]$  is also satisfiable.*

*Proof.* Unsatisfaction of  $\sigma'[\chi]$  may only come from the following cases:

1. A component of  $\sigma'[\chi]$  is unsatisfiable,
2. two components of  $\sigma'[\chi]$  contradict each other.

A component in  $\sigma'[\chi]$  is  $\pi_{\psi_i}$  or  $\neg\pi_{\psi_i}$ , hence the first case never happens. If two elements contradict each other, that is, if one is  $\pi_{\psi_i}$  and the other  $\neg\pi_{\psi_i}$  then surely  $\chi$  has elements  $\psi_i$  and  $\neg\psi_i$ , which renders it unsatisfiable.

**Theorem 2.** *Consider signature  $(\Delta, \Sigma) \in |\text{Sign}^{\mathcal{HI}}|$  and  $\rho \in \text{Sen}^{\mathcal{HI}}(\Delta, \Sigma)$ . If  $\rho$  is satisfiable,  $\sigma(\rho)$  is also satisfiable.*

*Proof.* If  $\rho$  is satisfiable we have a model  $M = (W, R, m) \in |\text{Mod}^{\mathcal{HI}}(\Delta, \Sigma)|$  such that  $M \models^w \rho$  for some  $w \in W$ . Through this assumption and Lemma 2, we define a model  $(W, R, m') \in |\text{Mod}^{\mathcal{HPL}}(\Delta, \Sigma)|$  as follows: for any  $w \in W$ ,  $m'(w)$  is a model satisfying  $\sigma'[\Omega_\rho^{m(w)}]$  (recall that Lemma 2 proves that  $\sigma'[\Omega_\rho^{m(w)}]$  is satisfiable).

It remains to show that  $(W, R, m') \models^w \sigma(\rho)$ , for some  $w \in W$ . Since models  $(W, R, m)$  and  $(W, R, m')$  have the same Kripke structure and  $\rho, \sigma(\rho)$  only differ in the base sentences, we just need to check that for all  $\chi \in \Omega_\rho$ ,  $m(w) \models \chi$  entails that  $m'(w) \models \sigma'[\chi]$  for any  $w \in W$ . Actually, this is a direct consequence of condition,  $m(w) \models \Omega_\rho^{m(w)}$  entails that  $m'(w) \models \sigma'[\Omega_\rho^{m(w)}]$  for all  $w \in W$ , which is freely given by the definition of  $(W, R, m')$ .

Now, we want to show the converse of Theorem 2. For this we need yet another definition to cater for the “preservation” of information with respect to satisfiability of the base sentences; information that is “lost” by  $\sigma(\rho)$ . Thus,

**Definition 15.** *Let  $\text{Sat}^{\mathcal{I}}$  be an effective decision procedure of  $\mathcal{I}$ , and  $\vee$  denote the disjunction operator, built from  $\mathbb{A}, \neg$ . Then define*

$$\eta(\rho) = \begin{cases} \vee \{ \chi \in \Omega_\rho \mid \text{Sat}^{\mathcal{I}}(\chi) \text{ is “unsat” } \}, & \text{if } B_\rho \neq \emptyset \\ \perp, & \text{otherwise} \end{cases}$$

**Corollary 2.** *Clearly, satisfiability of  $\rho$  entails satisfiability of  $\rho \mathbb{A} \neg \eta(\rho)$ , which in turn, by Theorem 2, entails satisfiability of  $\sigma(\rho \mathbb{A} \neg \eta(\rho))$ .*

**Lemma 3.** *Consider a model  $(W, R, m) \in |\text{Mod}^{\mathcal{HPL}}(\Delta, \text{Prop})|$  such that  $(W, R, m) \models \sigma(\rho \mathbb{A} \neg \eta(\rho))$ . For any  $\chi \in \Omega_\rho$ , if  $\sigma'[\chi]$  is satisfied by a model in  $\text{img}(m)$ ,  $\chi$  is satisfiable.*

*Proof.* If  $\chi$  is unsatisfiable then, by definition of  $\eta$ , occurs as one of the literals in  $\eta(\rho)$ , hence no model in  $\text{img}(m)$  may satisfy it.

**Theorem 3.** Consider signature  $(\Delta, \Sigma) \in |\text{Sign}^{\mathcal{HI}}|$  and  $\rho \in \text{Sen}^{\mathcal{HI}}(\Delta, \Sigma)$ . If  $\sigma(\rho \mathbin{\&\&} A \neg \eta(\rho))$  is satisfiable, then  $\rho$  is satisfiable.

*Proof.* If  $\sigma(\rho \mathbin{\&\&} A \neg \eta(\rho))$  is satisfiable we have a model  $M = (W, R, m) \in |\text{Mod}^{\mathcal{HPL}}(\Delta, \text{Prop})|$  such that  $M \models^w \sigma(\rho \mathbin{\&\&} A \neg \eta(\rho))$  for some  $w \in W$ . Through this assumption, and by Lemma 3, we define a model  $(W, R, m') \in |\text{Mod}^{\mathcal{HI}}(\Delta, \Sigma)|$  as follows: for any  $w \in W$ ,  $m'(w)$  is a model satisfying  $\chi$  where  $\sigma'[\chi] = \sigma'[\Omega_\rho^{m(w)}]$

It remains to show that  $(W, R, m') \models^w \rho$  for some  $w \in W$ . Since models  $(W, R, m)$  and  $(W, R, m')$  have the same Kripke structure satisfied by the sentences  $\rho, \sigma(\rho \mathbin{\&\&} A \neg \eta(\rho))$ , we just have to show that for all  $\chi \in \Omega_\rho$ ,  $m(w) \models \sigma'[\chi]$  entails that  $m'(w) \models \chi$  for any  $w \in W$ . Actually, this is a direct consequence of condition,  $m(w) \models \sigma'[\Omega_\rho^{m(w)}]$  entails  $m'(w) \models \Omega_\rho^{m(w)}$ , for all  $w \in W$ , which is given by the definition of  $(W, R, m')$ .

**Corollary 3.** From Corollary 2 and Theorem 3 we have that

$$\rho \text{ is satisfiable iff } \sigma(\rho \mathbin{\&\&} A \neg \eta(\rho)) \text{ is satisfiable.}$$

Then, since  $\mathcal{H}'PL$  was already proved to be decidable [12], we may use an effective decision procedure of  $\mathcal{H}'PL$  to check for satisfiability of sentences written in  $\mathcal{H}'I$ . This leads to the expected result

**Corollary 4.** If  $I$  is decidable then  $\mathcal{H}'I$  is also decidable.

Note that the proof of Theorem 3 paves the way for an example decision algorithm, that is, an algorithm able not only to answer “yes” or “no” to the question “Is  $\rho$  satisfiable?”, but also to build a model that satisfies sentence  $\rho$ , if it exists. Technically, to construct such an algorithm one also needs to have example decision algorithms for both  $I$  and  $\mathcal{H}'PL$ . The latter has at least one prover that meets this requirement [12]. Then, as indicated in the proof, through a  $\mathcal{H}'PL$ ’s decision procedure, one extracts a Kripke frame for  $\rho$  in which suitable models of  $I$  are “attached” given its example decision algorithm for  $I$ .

Finally, note that the decision algorithm for  $\mathcal{H}'I$ , conceptualised in Theorem 3, may be computationally hard. Indeed, in order to define  $\eta(\rho)$  the decision algorithm for  $I$  must be executed  $2^n$  times where  $n = |B_\rho|$ .

In addition, if we want the algorithm to give example models, the decision procedure for  $I$  must also be executed a number of times that can reach the number of worlds in the model built by the decision procedure for  $\mathcal{H}'PL$ .

### 4.3 Soundness and completeness

In this section we focus on the entailment system for  $\mathcal{HI}$ , i.e., on functor  $\text{Prf}^{\mathcal{HI}}$ , to show that the rules in  $\text{Prf}^{\mathcal{HI}}$  are both sound and complete. Recall that hybridised logics have the typical boolean connectives and that  $\text{Prf}^{\mathcal{HI}}$  has the *reductio ad absurdum* property. Of course, this means that proving soundness and completeness of  $\text{Prf}^{\mathcal{HI}}$  boils down to show the equivalence,

$$\rho \text{ is satisfiable iff } \rho \longrightarrow \perp \text{ is not in } \text{Prf}^{\mathcal{HI}}(\Delta, \Sigma)$$

Recall also that it is assumed that the base institution has the typical boolean connectives and the explicit satisfaction property, as well as that its proof system,  $Prf^I$ , is sound, complete and has the *reductio ad absurdum* property.

**Theorem 4.** *If  $Prf^I$  is sound, then  $Prf^{\mathcal{H}I}$  is also sound.*

*Proof.* Consider signature  $(\Delta, \Sigma) \in |Sign^{\mathcal{H}I}|$  and  $\rho \in Sen^{\mathcal{H}I}(\Delta, \Sigma)$ .

If  $Prf^{\mathcal{H}I}$  is sound then sentence  $\rho$ , being satisfiable means that there is no proof arrow  $\rho \longrightarrow \perp$  in  $Prf^{\mathcal{H}I}(\Delta, \Sigma)$ . If such an arrow exists, however, it must come from some of the conditions imposed to  $Prf^{\mathcal{H}I}(\Delta, \Sigma)$ , *i.e.*, some of these conditions must be unsound. We check each one:

1. the condition that proof arrows  $\emptyset \longrightarrow \rho$  in  $Prf^I(\Sigma)$  come to  $Prf^{\mathcal{H}I}(\Delta, \Sigma)$  is, by assumption, sound.
2. the axioms and proof rules from Table 1 were already proved to be sound (*cf.* [3]).
3. composition, inclusion and product rules are, by definition, sound.

The proof of completeness is more complex. For this we resort to a procedure similar to the one used for proving decidability.

**Theorem 5.** *Consider a signature  $(\Delta, \Sigma) \in |Sign^{\mathcal{H}I}|$  and  $\rho \in Sen^{\mathcal{H}I}(\Delta, \Sigma)$ . If there is no arrow  $\rho \longrightarrow \perp$  in  $Prf^{\mathcal{H}I}(\Delta, \Sigma)$  then there is also no arrow  $\sigma(\rho) \longrightarrow \perp$  in  $Prf^{\mathcal{H}PL}(\Delta, Prop)$ ,*

*Proof.* First notice that rules in Table 1 do not distinguish  $\rho$  from  $\sigma(\rho)$ , that is, any such rule may be applied to both sentences. Then observe that, since Table 1 contains all classical tautologies,  $Prf^{PL}$  does not bring new rules to  $Prf^{\mathcal{H}PL}$  and therefore rules in  $Prf^{\mathcal{H}PL}$  are also in  $Prf^{\mathcal{H}I}$ . Both remarks entail that if there are rules in  $Prf^{\mathcal{H}PL}$  that can generate arrow  $\sigma(\rho) \longrightarrow \perp$ , then the same set of rules (also present in  $Prf^{\mathcal{H}I}$ ) can surely generate it there.

Next, we show the converse of Theorem 5 holds as well. For this we define a function to play a role similar to that played by  $\eta$  in sub-section 4.2.

**Definition 16.** *Consider a signature  $(\Delta, \Sigma) \in |Sign^{\mathcal{H}I}|$  and  $\rho \in Sen^{\mathcal{H}I}(\Delta, \Sigma)$ . Then define,*

$$\eta'(\rho) = \begin{cases} \bigvee \{\chi \in \Omega_\rho \mid \chi \longrightarrow \perp \text{ is in } Prf^I\}, & \text{if } B_\rho \neq \emptyset \\ \perp, & \text{otherwise} \end{cases}$$

**Corollary 5.** *Clearly if there is no arrow  $\rho \longrightarrow \perp$  in  $Prf^{\mathcal{H}I}(\Delta, \Sigma)$  then there is also no arrow  $(\rho \mathbin{\wedge} A \neg \eta'(\rho)) \longrightarrow \perp$  in  $Prf^{\mathcal{H}I}(\Delta, \Sigma)$ .*

**Lemma 4.** *Consider a model  $(W, R, m) \in |Mod^{\mathcal{H}PL}(\Delta, Prop)|$  such that  $(W, R, m) \models \sigma(\rho \mathbin{\wedge} A \neg \eta'(\rho))$ . For any  $\chi \in \Omega_\rho$ , if  $\sigma'[\chi]$  is satisfied by a model member of  $img(m)$ ,  $\chi$  is satisfiable.*

*Proof.* If  $\chi$  is unsatisfiable then, by definition of  $\eta'$  and completeness of  $Prf^I$ , occurs as one of the literals in  $\eta'(\rho)$ , hence no model member of  $img(m)$  may satisfy it.

**Theorem 6.** *If  $Prf^I$  is complete then  $Prf^{\mathcal{H}I}$  is also complete.*

*Proof.* We want to prove that given a signature  $(\Delta, \Sigma) \in |Sign^{\mathcal{H}I}|$  and a sentence  $\rho \in Sen^{\mathcal{H}I}(\Delta, \Sigma)$ , if no arrow  $\rho \longrightarrow \perp$  exists in  $Prf^{\mathcal{H}I}(\Delta, \Sigma)$  then  $\rho$  is satisfiable.

Hence, let us assume that there is no arrow  $\rho \longrightarrow \perp$  in  $Prf^{\mathcal{H}I}(\Delta, \Sigma)$ , which by Corollary 5, entails that there is no proof arrow  $\sigma(\rho \mathbin{\&\&} A \neg \eta'(\rho)) \longrightarrow \perp$  in  $Prf^{\mathcal{H}PL}(\Delta, Prop)$  and therefore means that  $\sigma(\rho \mathbin{\&\&} A \neg \eta'(\rho))$  is satisfiable. In other words, we have a model  $M = (W, R, m) \in |Mod^{\mathcal{H}PL}(\Delta, Prop)|$  such that  $M \models^w \sigma(\rho \mathbin{\&\&} A \neg \eta'(\rho))$  for some  $w \in W$ . Then, by Lemma 4, we are able to define a model  $(W, R, m') \in |Mod^{\mathcal{H}I}(\Delta, \Sigma)|$ , in which, for any  $w \in W$ ,  $m'(w)$  is a model for  $\chi$  where  $\sigma'[\chi] = \sigma'[\Omega_\rho^{m(w)}]$ .

It remains to show that  $(W, R, m') \models^w \rho$  for some  $w \in W$ . Since models  $(W, R, m)$  and  $(W, R, m')$  have the same Kripke structure satisfied by sentences  $\rho$  and  $\sigma(\rho \mathbin{\&\&} A \neg \eta'(\rho))$ , it is enough to show that, for all  $\chi \in \Omega_\rho$ ,  $m(w) \models \sigma'[\chi]$  entails that  $m'(w) \models \chi$  for any  $w \in W$ . Actually, this is a direct consequence of the fact that  $m(w) \models \sigma'[\Omega_\rho^{m(w)}]$  entails  $m'(w) \models \Omega_\rho^{m(w)}$ , for all  $w \in W$ , which comes from the definition of  $(W, R, m')$ .

## 5 Conclusions and future work

This paper lays the first steps towards the development of (dedicated) proof tools for hybridised logics, by providing an effective decision algorithm for the satisfiability problem. Additionally the systematic hybridisation of the calculus of a base logic was addressed, and shown to preserve both soundness and completeness.

The next step, from an engineering point of view, is, of course, to develop such a generic, dedicated prover for hybridised logics. A comparison with the strategy of using the parametrised translation to FOL will then be due.

In a similar line of research, lies the development of an alternative decision algorithm, that potentially overcomes the problem detected in the definition of  $\eta$ , which involves calling the decision procedure of the base logic  $2^n$  times, for  $n$  the number of base sentences in the sentence under consideration. Such an algorithm may be based on the tableau technique (for instance, the one implemented in [12]) which opens a number of branches as the possible ways to build a model satisfying a given sentence. If the sentence is unsatisfiable then all branches must be closed. If any branch remains open then the decision procedure of the base logic is called to try to close it. Thus, the number of times the decision procedure of the base logic is called is much smaller than in the approach discussed here.

Other results in the literature abstract the combination of logics pattern by considering the “top logic” itself arbitrary. Such is the case of what is called *parametrisation* of logics in [4] by C. Caleiro, A. Sernadas and C. Sernadas. Similarly, the recent method of *importing* logics suggested by J. Rasga, A. Sernadas and C. Sernadas [16] aims at formalising this kind of asymmetric combinations resorting to a graph-theoretic approach. In both cases some decidability and completeness results are given. It should be interesting to see in which ways the hybridisation method relates to these works.

**Acknowledgements.** This work is funded by ERDF - European Regional Development Fund, through the COMPETE Programme, and by National Funds through FCT within project FCOMP-01-0124-FEDER-028923. M. A. Martins was also supported by the project PEst-OE/MAT/UI4106/2014.

## References

1. C. Areces and B. ten Cate. Hybrid logics. In P. Blackburn, F. Wolter, and J. van Benthem, editors, *Handbook of Modal Logics*. Elsevier, 2006.
2. Pedro Baltazar. Probabilization of logics: Completeness and decidability. *Logica Universalis*, 7(4):403–440, 2013.
3. T. Braüner. *Proof-Theory of Propositional Hybrid Logic*. Hybrid Logic and its Proof-Theory, 2011.
4. Carlos Caleiro, Cristina Sernadas, and Amílcar Sernadas. Parameterisation of logics. In *WADT*, pages 48–62, 1998.
5. Răzvan Diaconescu. *Institution-independent Model Theory*. Birkhäuser Basel, 2008.
6. Răzvan Diaconescu. Institutional semantics for many-valued logics. *Fuzzy Sets Syst.*, 218:32–52, May 2013.
7. Răzvan Diaconescu and Petros Stefaneas. Ultraproducts and possible worlds semantics in institutions. *Theor. Comput. Sci.*, 379(1-2):210–230, July 2007.
8. José Fiadeiro and Amílcar Sernadas. Structuring theories on consequence. In D. Sannella and A. Tarlecki, editors, *Recent Trends in Data Type Specification*, volume 332 of *Lecture Notes in Computer Science*, pages 44–72. Springer Berlin Heidelberg, 1988.
9. Marcelo Finger and Dov Gabbay. Adding a temporal dimension to a logic system. *Journal of Logic, Language and Information*, 1(3):203–233, 1992.
10. Joseph A. Goguen and Rod M. Burstall. Institutions: abstract model theory for specification and programming. *J. ACM*, 39:95–146, January 1992.
11. Joseph A. Goguen and José Meseguer. Models and equality for logical programming. In Hartmut Ehrig, Robert Kowalski, Giorgio Levi, and Ugo Montanari, editors, *TAPSOFT '87*, volume 250 of *Lecture Notes in Computer Science*, pages 1–22. Springer Berlin Heidelberg, 1987.
12. Guillaume Hoffmann and Carlos Areces. Htab: a terminating tableaux system for hybrid logic. *Electr. Notes Theor. Comput. Sci.*, 231:3–19, 2009.
13. Alexandre Madeira, José M. Faria, Manuel A. Martins, and Luís Soares Barbosa. Hybrid specification of reactive systems: An institutional approach. In G. Barthe, A. Pardo, and G. Schneider, editors, *Software Engineering and Formal Methods (SEFM 2011, Montevideo, Uruguay, November 14-18, 2011)*, volume 7041 of *Lecture Notes in Computer Science*, pages 269–285. Springer, 2011.
14. Manuel A. Martins, Alexandre Madeira, Răzvan Diaconescu, and Luís Soares Barbosa. Hybridization of institutions. In A. Corradini, B. Klin, and C. Cîrstea, editors, *Algebra and Coalgebra in Computer Science (CALCO 2011, Winchester, UK, August 30 - September 2, 2011)*, volume 6859 of *Lecture Notes in Computer Science*, pages 283–297. Springer, 2011.
15. Renato Neves, Alexandre Madeira, Manuel A. Martins, and Luís S. Barbosa. Hybridisation at work. In *CALCO TOOLS*, volume 8089 of *Lecture Notes in Computer Science*. Springer, 2013.
16. J. Rasga, A. Sernadas, and C. Sernadas. Importing logics: Soundness and completeness preservation. *Studia Logica*, 101(1):117–155, 2013.