

# Rigid First-Order Hybrid Logic <sup>\*</sup>

Patrick Blackburn<sup>1</sup>, Manuel Martins<sup>2</sup>, María Manzano<sup>3</sup>, and Antonia Huertas<sup>4</sup>

<sup>1</sup> IKH, Roskilde University, Denmark [patrick.rowan.blackburn@gmail.com](mailto:patrick.rowan.blackburn@gmail.com)

<sup>2</sup> CIDMA, University of Aveiro, Portugal [martins@ua.pt](mailto:martins@ua.pt)

<sup>3</sup> University of Salamanca, Spain [mar@usal.es](mailto:mar@usal.es)

<sup>4</sup> Universitat Oberta de Catalunya, Spain [mhuertass@uoc.edu](mailto:mhuertass@uoc.edu)

**Abstract.** Hybrid logic is usually viewed as a variant of modal logic in which it is possible to refer to worlds. But when one moves beyond propositional hybrid logic to first- or higher-order hybrid logic, it becomes useful to view it as a systematic *modal language of rigidification*. The key point is this: @ can be used to rigidify not merely formulas, but other types of symbol as well. This idea was first explored in first-order hybrid logic (without function symbols) where @ was used to rigidify the first-order constants. It has since been used in hybrid type-theory: here one only has function symbols, but they are of every finite type, and @ can rigidify any of them. This paper fills the remaining gap: it introduces a first-order hybrid language which handles function symbols, and allows predicate symbols to be rigidified. The basic idea is straightforward, but there is a slight complication: transferring information about rigidity between the level of terms and formulas. We develop a syntax to deal with this, provide an axiomatization, and prove a strong completeness result for a varying domain (actualist) semantics.

**Keywords:** Hybrid logic · First-order modal logic · Rigidity · Rigid predicate symbols · Function symbols · Varying domains · Actualist semantics · Henkin models

## 1 Introduction

Hybrid logic is usually viewed as a variant of modal logic in which it is possible to refer to worlds. But when one moves beyond propositional hybrid logic to first- or even higher-order hybrid logic, it becomes more useful to view it as a systematic *modal language of rigidification*. Rigidity has long been an important concept in first-order modal logic: a first-order constant is said to be rigid if it denotes the same individual in all worlds, and free first-order variables in most first-order

---

<sup>\*</sup> Accepted authors' manuscript published as: P. Blackburn, M. Martins, M. Manzano, A. Huertas. *Rigid First-Order Hybrid Logic*. In: Iemhoff R., Moortgat M., de Queiroz R. (eds) *Logic, Language, Information, and Computation*. WoLLIC 2019. Lecture Notes in Computer Science, vol 11541, pp 53–69. Springer International Publishing, 2019. [DOI:10.1007/978-3-662-59533-6\_4]. The final publication is available at Springer via [https://rd.springer.com/chapter/10.1007/978-3-662-59533-6\\_4](https://rd.springer.com/chapter/10.1007/978-3-662-59533-6_4).

modal logics are interpreted rigidly. But rigidity is also central to hybrid logic, and it is central at all levels, from the propositional to the type-theoretic.

Recall that propositional hybrid logic makes use of special propositional symbols called *nominals* (typically written as  $i$ ,  $j$  and  $k$ ) that are true at exactly one world in any model: in effect, nominals act as names for the unique world they are true at. Basic hybrid logic also allows us to write expressions of the form  $@_i p$ , which means “evaluate  $p$  at the unique world called  $i$ ”. Read this way,  $@_i$  is a *modality* whose task is to inspect what is going on at the  $i$ -world, and see whether  $p$  is true there or not. But  $@_i$  can also be read as a *rigidifier*: “form the *rigid* proposition  $@_i p$  out of the proposition  $p$ ”. Note that  $@_i p$  is indeed rigid: it is either true at all worlds, or false at all worlds, depending on whether  $p$  is true or false at the  $i$ -world.

As was soon realized, we can use  $@$  to rigidify more than propositions. The paper [5], which explores first-order hybrid logic, took the first step in this direction by introducing expressions of the form  $(@_i c)$ . Here  $c$  is a first-order constant, which may denote different individuals in different worlds, and  $(@_i c)$  is the *rigid* constant that denotes (at all worlds) whatever it is that  $c$  denotes at world  $i$ .<sup>5</sup> For example, if we think of  $c$  as a (non-rigid) constant that names **Donald-Trump** in some world  $w$ , and **Bernie-Sanders** in some other world  $w'$ , then, if  $j$  is the nominal that names world  $w'$ ,  $(@_j c)$  is a *rigid* constant that names **Bernie-Sanders** at all worlds. On the technical side, it was observed that Henkin-style model building techniques could be used to build first-order Kripke models whose frames were defined using equivalence classes of nominals, and whose domains of quantification were defined using equivalence classes of rigidified constants; this led to a number of general completeness and interpolation results (see [2,6]).

But the idea of using  $@$  as a general-purpose rigidifier has been most widely applied in the setting of *higher-order* hybrid logic. A range of higher-order hybrid logics, most based on Church’s theory of simple types, have been defined (see [1,4,10,9]) and general Henkin-style completeness results for them proved. Although these systems differ in various ways, they have two points in common. First, they all allow expressions of the form  $(@_i f)$  to be formed where  $f$  is a function symbol of *any* finite type. That is: in the higher-order setting,  $@$  is totally overloaded — it can rigidify *all* the types of information that these languages can describe. Second, the higher-order Kripke models needed to prove completeness are constructed out the equivalence classes<sup>6</sup> of nominals (to define the frame) and equivalence classes of rigidified function symbols are used to define the needed function hierarchies.

---

<sup>5</sup> Note that in this paper expressions of the form  $(@_i c)$  were introduced *in addition* to expressions of the form  $@_i p$ . As the authors of this paper put it: they deliberately *overloaded* the  $@$  symbol. In this paper, we are going to overload  $@$  even more. Our basic convention will be to omit the enclosing out brackets when propositional information is rigidified (as in  $@_i p$ ), and to use enclosing brackets when other types of information are rigidified (as in  $(@_i c)$ ). More on this later.

<sup>6</sup> Or sets of rigidified function symbols in the partial type theory explored in [9].

One gap in this picture remains, and the purpose of this paper is to start filling it: to strengthen the first-order hybrid language defined in [5] to handle first-order function symbols and rigidified predicate symbols. There are several reasons for exploring such languages. For a start, many of the classical conceptual problems surrounding modal logic arise in first-order modal logic. One could explore them in a higher-order logic, but these are more complex, and bring new philosophical problems along with them. An expressive first-order hybrid language — one that makes it possible to rigidify constants, function symbols, predicate symbols and formulas — offers useful resources for addressing classical conceptual issues. Rigidifying predicate symbols, for example, allows us to express precise distinctions: we can talk about the relation of **Love** with reference to the pairs of individuals in love in the  $i$ -world, that is,  $(@_i\text{Love})$ ; or we can specify that we are interested in its denotation in the  $j$ -world using  $(@_j\text{Love})$ .

There is also a more down-to-earth reason for our interest in this language: it is rare to see function symbols treated in any detail in discussions of first-order modal logic. Most authors skate lightly over the omission, and the reader is left with the impression that extending first-order modal logic to cope with function is a routine extension of what is already known. This seems misguided. First-order modal logic raises a wide range of technical and conceptual challenges, especially if one wants to work with varying domains (that is: with an actualist semantics). This paper provides a general approach to handling function symbols in a first-order modal logic with an actualist semantics.

The basic ideas explored in this paper are straightforward and build on previous work in the hybrid literature (probably [6] and [3] are the two most directly relevant references). However there is one complication: transferring information about rigidity between the first-order terms and formulas (this is an issue that does not arise in the type-theoretic case, where one only has to deal with function symbols). We cope with this by defining a recursive notion of rigidification, which keeps the term syntax relatively simple.

We proceed as follows. In Section 2 we define the syntax of our first-order language of rigidification, and what it means to rigidify a term. In Section 3 we define a varying domain semantics for our language, and note a basic lemma about rigid terms. In Section 4 we provide an axiomatisation, and in Section 5 and the Appendix we prove that it is complete. Finally, in Section 6 we sketch the ways we are developing the work reported here.

## 2 Rigid first-order hybrid logic

We start with a first-order signatures, consisting of  $n$ -ary function and relation symbols:

**Definition 1 (Signature).** *A first-order signature  $\Sigma$  is a pair  $((\text{Func}_n)_{n \in \mathbb{N}}, (\text{Rel}_n)_{n \in \mathbb{N}})$ , where  $\text{Func}_n$  and  $\text{Rel}_n$  are sets of functional and relational symbols of arity  $n$ , respectively. The indexed elements in either family may be empty, and if they are all empty, we have the empty signature. The elements of  $\text{Func}_0$  (if any)*

are called constants, and the elements of  $\text{Rel}_0$  (if any) are called propositional symbols.

We intend to use such signatures in a first-order hybrid language, thus we next to add first-order variables and nominals, and then “rigidify the signature” by allowing any function or relation symbol (including any constants or propositional symbols) to be preceded by rigidifying operators of the form  $@_i$ .

**Definition 2.** A first-order hybrid similarity type  $\tau$  is a tuple  $\langle \Sigma, X, \text{NOM} \rangle$  where  $\Sigma$  is first-order signature,  $X$  is a countably infinite set of variables and  $\text{NOM}$  is a set of symbols, called nominals. The  $\text{NOM}$ -rigidification of  $\Sigma$  (with respect to  $\tau$ ) is the signature:  $@\Sigma = ((@ \text{Func}_n)_{n \in \mathbb{N}}, (@ \text{Rel}_n)_{n \in \mathbb{N}})$ , where  $@ \text{Func}_n = \{(@_i f) : i \in \text{NOM}, f \in \text{Func}_n\}$  and  $@ \text{Rel}_n = \{(@_i P) : i \in \text{NOM}, P \in \text{Rel}_n\}$ .

Given a similarity type  $\tau$ , we define the set of rigid terms, and the set of terms, as follows:

**Definition 3 (Terms).** Let  $\tau$  be a first-order hybrid similarity type.

The set of rigid  $\Sigma$ -terms over  $\tau$ ,  $@\text{Term}(\tau)$ , is recursively defined by:

- for any  $x \in X$ ,  $x \in @\text{Term}(\tau)$ ;
- for any  $f^\circ \in @ \text{Func}_n$ , and all terms  $t_i \in @\text{Term}(\tau)$ ,  $i = 1, \dots, n$ ,  
 $f^\circ(t_1, \dots, t_n) \in @\text{Term}(\tau)$ .

The set of  $\Sigma$ -terms over  $\tau$ ,  $\text{Term}(\tau)$ , is recursively defined by:

- for any  $x \in X$ ,  $x \in \text{Term}(\tau)$ ;
- for any  $f \in \text{Func}_n \cup @ \text{Func}_n$ , and all terms  $t_i \in \text{Term}(\tau)$ ,  $i = 1, \dots, n$ ,  
 $f(t_1, \dots, t_n) \in \text{Term}(\tau)$ .

Clearly every rigid term is a term, that is,  $@\text{Term}(\tau) \subseteq \text{Term}(\tau)$ . We call a term ground if it contains no variables.

The elements of  $\text{Func}_0$  (that is, constants) will play an important role in the completeness proof, as we will then expand our language by adding denumerably many new constants (“Henkin witness constants”) to prove our Lindenbaum lemma. So it is worth noting that (by the previous definition) elements of the form  $(@_i c)$ , where  $c$  is a constant symbol, are indeed rigid terms (that is, elements of  $@\text{Term}(\tau)$ ) as all such expressions belong to  $@ \text{Func}_0$ .

Now for an important definition. Given a term  $t$  and a nominal  $i$ , we can (recursively) rigidify  $t$  at  $i$ , as follows:

**Definition 4 (Rigidification of a term).** Let  $t \in \text{Term}(\tau)$  and  $i \in \text{NOM}$ . The rigidification of  $t$  at  $i$  is the term  $@_i t \in @\text{Term}(\tau)$  recursively defined by:

- if  $t \in X$ ,  $@_i t := t$
- if  $t = f(t_1, \dots, t_n)$  with  $f \in @ \text{Func}_n$ , then  $@_i t := f(@_i t_1, \dots, @_i t_n)$
- if  $t = f(t_1, \dots, t_n)$  with  $f \in \text{Func}_n$ , then  $@_i t := (@_i f)(@_i t_1, \dots, @_i t_n)$

To spell this out: first, the rigidification process ignores variables, as they will always be interpreted rigidly. Second, if the functor prefixing a term is of the form  $(@_j f)$ , which means that we have a syntactic guarantee that it is rigid, then we ignore it and go on to recursively rigidify its arguments. Third, if the functor prefixing a term is of the form  $f$  (that is, we have no syntactic guarantee of its rigidity) we replace the functor  $f$  by the rigid form  $(@_i f)$  and go on to recursively rigidify its arguments. Note that for the special case of constants (functions of arity 0) we have: given a constant  $c$ , and nominals  $i$  and  $j$ , the rigidification of  $c$  with respect to  $i$  is  $(@_i c)$ , and the rigidification of  $c$  with respect to  $j$  is  $(@_j c)$ . So the base case of the recursion is simply the rigidification-of-first-order-constants used in [5]. Also note that when a term  $t \in @Term(\tau)$  is rigidified, the result is simply  $t$  itself. That is, rigidification is the identity map on  $@Term(\tau)$ .

**Definition 5.** *The set of  $Fm(\tau)$  of first-order hybrid formulas is the smallest set such that:*

1.  $NOM \subseteq Fm(\tau)$ ;
2.  $t_1 \approx t_2 \in Fm(\tau)$ , for any  $t_1, t_2 \in Term(\tau)$
3.  $P(t_1, \dots, t_n) \in Fm(\tau)$ , for any  $P \in Rel_n \cup @Rel_n$  and  $t_1, \dots, t_n \in Term(\tau)$ ;
4. if  $\varphi \in Fm(\tau)$  and  $i$  is a nominal, then  $@_i \varphi \in Fm(\tau)$ ;
5. if  $\varphi \in Fm(\tau)$ , then  $\neg\varphi, \Box\varphi \in Fm(\tau)$ ;
6. if  $\varphi \in Fm(\tau)$  and  $\psi \in Fm(\tau)$  then  $\varphi \wedge \psi \in Fm(\tau)$  and  $\varphi \vee \psi \in Fm(\tau)$ .
7. if  $x \in X$  and  $\varphi \in Fm(\tau)$ , then  $\forall x\varphi \in Fm(\tau)$ .

We use familiar abbreviations:  $\Diamond\varphi$  is  $\neg\Box\neg\varphi$ ,  $\exists x\varphi$  is  $\neg\forall x\neg\varphi$ ,  $\varphi \rightarrow \psi$  is  $\neg(\varphi \wedge \neg\psi)$ , and so on. We define  $EXISTS(t)$  to be  $\exists x(x \approx t)$ , provided that  $x$  does not occur in  $t$ , as is standard in varying domain approaches to first-order modal logic.

It is worth explicitly noting some of the syntactic distinctions that can be drawn in this language. Let  $i$  and  $j$  be nominals, let  $c$  and  $d$  be constant symbols, and let  $P$  be a two-place predicate symbol. Then  $P(c, d)$  is a formula, one that displays no syntactic indications concerning rigidity.  $P(@_i c, @_j d)$  is also a formula, though this time the two constants it contains have been rigidified. Furthermore,  $(@_i P)(c, d)$  is also a formula, though here it is the initial predicate has been rigidified. Indeed,  $(@_i P)((@_i c), (@_j d))$  is a formula too, though this time the predicate and both constants have been rigidified. But there are other possibilities. In particular, note that  $@_i P(c, d)$  is also a formula: it is the formula  $P(c, d)$  preceded by  $@_i$ . Note that this is *not* the same formula as  $(@_i P)(c, d)$ . Indeed, under the semantics we shall shortly define, the two formulas have importantly different properties:  $@_i P(c, d)$  is guaranteed to be a rigid proposition (it will either be true at all worlds or false at all worlds) while  $(@_i P)(c, d)$  may vary in truth value from world to world.

Hopefully these examples help make our basic bracketing convention clear: when we combine  $@_i$  with any formula  $\varphi$  (that is: propositional information) then we write the resulting formula as  $@_i \varphi$  (that is: with no enclosing brackets). On the other hand, when we combine  $@_i$  with either a function symbol  $f$ , a constant symbol  $c$ , or a predicate symbol  $P$  of arity  $\geq 1$ , then we write the resulting rigidifications as  $(@_i f)$ ,  $(@_i c)$  and  $(@_i P)$  respectively (that is: with

enclosing brackets). In the case of a predicate symbol  $p$  of arity 0 (that is: the propositional symbol  $p$ ) we write  $@_i p$ , since propositional symbols are formulas.

However one other point should be emphasized: in statements of lemmas and axiom schemas we sometimes write expressions of the form  $@_i t$  (for  $i$  a nominal and  $t$  a term). Here it is important to recall that such expressions are *not* members of the *object* language, rather they are *metalinguistic* abbreviation for the rigidification of  $t$  at  $i$  as defined by Definition 4.

### 3 Semantics

We now define a varying domain (actualist) semantics for our language. There are several choices available; here we simply remark that we have aimed for a general semantics, and typically follow the decisions made in [8]. We will say more about this in the paper’s conclusion.

**Definition 6 (Skeleton).** *A skeleton over  $\tau$  is a tuple  $M = (W, \text{Dom}, D, R)$ , where  $W \neq \emptyset$ ,  $\text{Dom}$  is a nonempty set,  $D : W \rightarrow P(\text{Dom})$  such that  $D(w) \neq \emptyset$  and  $R \subseteq W^2$ . We will usually write  $D_w$  for  $D(w)$ .*

That is: we have a non-empty set of worlds  $W$ , a binary accessibility relation  $R$  between these worlds, a global domain of objects  $\text{Dom}$ , and a function  $D$  which tells us which elements of these domain elements actually exist at any world  $w$ . We call  $D_w$  (for any  $w \in W$ ) a local domain. Local domains can be distinct, which is why this is a “varying domain” semantics.

**Definition 7.** *A model for a rigid first-order hybrid similarity type  $\tau$  is a pair  $\mathcal{M} = (M, I)$ , where  $M$  is a skeleton and  $I$  is the interpretation function such that:*

- For any  $i \in \text{NOM}$ ,  $I(i) \in W$ ,
- For any  $P \in \text{Rel}_n$  and any  $w \in W$ ,  $I_w(P) \subseteq (\text{Dom})^n$ , and
- For any  $f \in \text{Func}_n$  and any  $w \in W$ ,  $I_w(f) : (\text{Dom})^n \rightarrow \text{Dom}$ .

Note that (following [8]) we allow the interpretation of a predicate  $P$  to involve individuals that do not exist in the local domain. Analogously, we interpret function symbols in a way that lets them take as input entities that do not exist at the local domain, and to output non-local entities as well. This seems the simplest and most general starting point, but we’ll say more about this decision in the paper’s conclusion.

**Definition 8.** *Let  $\mathcal{M} = (M, I)$  be a model and  $g : X \rightarrow \text{Dom}$  be a variable assignment. The interpretation of terms is recursively defined as follows:*

- if  $t \in X$ ,  $[t]^{\mathcal{M}, w, g} = g(t)$ .
- if  $t = f(t_1, \dots, t_n)$ ,  $f \in \text{Func}_n$  with  $n \geq 0$ ,  
 $[t]^{\mathcal{M}, w, g} = I_w(f)([t_1]^{\mathcal{M}, w, g}, \dots, [t_n]^{\mathcal{M}, w, g})$

- if  $t = (@_i f)(t_1, \dots, t_n)$ ,  $f \in \text{Func}_n$  with  $n \geq 0$   
 $[t]^{\mathcal{M}, w, g} = I_{I(i)}(f)([t_1]^{\mathcal{M}, w, g}, \dots, [t_n]^{\mathcal{M}, w, g})$

We can now give the satisfaction definition.

**Definition 9.** Let  $\mathcal{M} = (M, I)$  be a model,  $g : X \rightarrow \text{Dom}$  an assignment and  $w \in W$ . Then:

$\mathcal{M}, w, g \models i$	iff $I(i) = w$
$\mathcal{M}, w, g \models t_1 \approx t_2$	iff $[t_1]^{\mathcal{M}, w, g} = [t_2]^{\mathcal{M}, w, g}$
$\mathcal{M}, w, g \models P(t_1, \dots, t_n)$	iff $([t_1]^{\mathcal{M}, w, g}, \dots, [t_n]^{\mathcal{M}, w, g}) \in I_w(P)$ , for $P \in \text{Rel}_n$ and $t_1, \dots, t_n \in \text{Term}(\tau)$
$\mathcal{M}, w, g \models (@_i P)(t_1, \dots, t_n)$	iff $([t_1]^{\mathcal{M}, w, g}, \dots, [t_n]^{\mathcal{M}, w, g}) \in I_{I(i)}(P)$ , for $P \in \text{Rel}_n$ and $t_1, \dots, t_n \in \text{Term}(\tau)$
$\mathcal{M}, w, g \models \neg\varphi$	iff $\mathcal{M}, w, g \not\models \varphi$
$\mathcal{M}, w, g \models \varphi \wedge \psi$	iff $\mathcal{M}, w, g \models \varphi$ and $\mathcal{M}, w, g \models \psi$
$\mathcal{M}, w, g \models @_i \varphi$	iff $\mathcal{M}, I(i), g \models \varphi$
$\mathcal{M}, w, g \models \Box\varphi$	iff for all $w' \in W$ such that $wRw'$ , $\mathcal{M}, w', g \models \varphi$
$\mathcal{M}, w, g \models \forall x\varphi$	iff for all $d \in D_w$ , $\mathcal{M}, w, g[x \mapsto d] \models \varphi$

A formula  $\varphi$  is said to be true at a world  $w$  under the assignment  $g$  if and only if  $\mathcal{M}, w, g \models \varphi$ . It is valid in a model  $\mathcal{M}$ , denoted by  $\mathcal{M} \models \varphi$ , if and only if, for every world  $w$  and every assignment  $g$  we have that  $\mathcal{M}, w, g \models \varphi$ .

**Lemma 1.** For every  $t \in \text{Term}(\tau)$ , every assignment  $g$  on  $\mathcal{M}$ , every world  $w$ , and every nominal  $i$  we have that:

$$[t]^{\mathcal{M}, I(i), g} = [@_i t]^{\mathcal{M}, w, g}$$

*Proof.* By induction on the structure of  $t$ . Recall from Definition 4 that  $@_i t$  is the (recursively defined) *rigidification* of term  $t$ .

## 4 Axiomatisation

This section gives an axiomatisation  $K_\tau$  for first-order hybrid logic, given a first-order hybrid similarity type  $\tau$ . We will take all propositional tautologies as axioms, and in addition:

### Distributivity axioms

$$(K_\Box) \Box(\varphi \rightarrow \psi) \rightarrow (\Box\varphi \rightarrow \Box\psi).$$

$$(K_{@}) @_i(\varphi \rightarrow \psi) \rightarrow (@_i\varphi \rightarrow @_i\psi).$$

### Quantifier axioms

$$(Q1) \forall x(\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \forall x\psi), \text{ where } x \text{ does not occur free in } \varphi.$$

$$(Q2) \forall x\varphi \rightarrow (\text{EXISTS}(\tau) \rightarrow \varphi(\frac{\tau}{x})), \text{ where } \tau \text{ is rigid.}$$

$$(Q3) \exists y\text{EXISTS}(y)$$

### Basic hybrid axioms

$$\begin{aligned}
(\text{Ref}_{@}) & \quad @_i i. \\
(\text{Agree}) & \quad @_i @_j \varphi \leftrightarrow @_j \varphi. \\
(\text{Selfdual}_{@}) & \quad @_i \varphi \leftrightarrow \neg @_i \neg \varphi. \\
(\text{Intro}) & \quad i \rightarrow (\varphi \leftrightarrow @_i \varphi). \\
(\text{Back}) & \quad \diamond @_i \varphi \rightarrow @_i \varphi.
\end{aligned}$$

**Axioms for  $\approx$** 

$$\begin{aligned}
(\text{Ref}_{\approx}) & \quad t_1 \approx t_1, \text{ for all } t_1 \in \text{Term}(\tau). \\
(\text{Sym}_{\approx}) & \quad (t_1 \approx t_2) \rightarrow (t_2 \approx t_1), \text{ for all } t_1, t_2 \in \text{Term}(\tau). \\
(\text{Trans}_{\approx}) & \quad ((t_1 \approx t_2) \wedge (t_2 \approx t_3)) \rightarrow (t_1 \approx t_3), \text{ for all } t_1, t_2, t_3 \in \text{Term}(\tau). \\
(\text{Func}) & \quad (t_1 \approx t'_1 \wedge \dots \wedge t_n \approx t'_n) \rightarrow f(t_1, \dots, t_n) \approx f(t'_1, \dots, t'_n), \\
& \quad \text{where } f \in \text{Func} \cup @\text{Func}, \text{ and } t_i, t'_i \in \text{Term}(\tau), \text{ for } i = 1, \dots, n, n \geq 0. \\
(\text{Pred}) & \quad (t_1 \approx t'_1 \wedge \dots \wedge t_n \approx t'_n) \rightarrow P(t_1, \dots, t_n) \leftrightarrow P(t'_1, \dots, t'_n), \\
& \quad \text{where } P \in \text{Rel} \cup @\text{Rel}, \text{ and } t_i, t'_i \in \text{Term}(\tau), \text{ for } i = 1, \dots, n, n \geq 0.
\end{aligned}$$

**Interactions between  $@$  and  $\approx$** 

$$\begin{aligned}
(\text{Rigidify}) & \quad @_i (c \approx (@_i c)), \text{ for any constant } c. \\
(K_{@ \approx}) & \quad @_i (t_1 \approx t_2) \leftrightarrow (@_i t_1 \approx @_i t_2), \text{ for all } t_1, t_2 \in \text{Term}(\tau). \\
(\text{Nom}_{\approx}) & \quad @_i j \rightarrow (@_i t \approx @_j t), t \in \text{Term}(\tau). \\
(\text{Agree}_{\approx}) & \quad @_i (t_1 \approx t_2) \leftrightarrow (t_1 \approx t_2), \text{ for all } t_1, t_2 \in @\text{Term}(\tau).
\end{aligned}$$

**Linking formula rigidity with predicate-and-term rigidity**

$$\begin{aligned}
(\text{Shuffle-1}) & \quad @_i P(t_1, \dots, t_n) \leftrightarrow (@_i P)(@_i t_1, \dots, @_i t_n). \\
(\text{Shuffle-2}) & \quad @_i (@_j P)(t_1, \dots, t_n) \leftrightarrow (@_j P)(@_i t_1, \dots, @_i t_n).
\end{aligned}$$

As rules of proof we take the following (these proof rules are discussed in detail in [6], and we shall note some results from this paper in what follows). For any formulas  $\varphi$  and  $\psi$ , and any nominals  $i$  and  $j$  we have:

$$\begin{aligned}
(\text{MP}) & \quad \frac{\varphi \rightarrow \psi \quad \varphi}{\psi} \\
(\text{Gen}_{@}) & \quad \frac{\varphi}{@_i \varphi} \\
(\text{Gen}_{\square}) & \quad \frac{\varphi}{\square \varphi} \\
(\text{Gen}_{\forall}) & \quad \frac{\varphi}{\forall x \varphi} \\
(\text{Name}) & \quad \frac{@_i \varphi}{\varphi}, \text{ where } i \text{ does not occur in } \varphi. \\
(\text{BG}) & \quad \frac{@_i \diamond j \rightarrow @_j \varphi}{@_i \square \varphi}, \text{ if } j \neq i \text{ and } j \text{ does not occur in } \varphi. \\
(\text{Subs}) & \quad \frac{\varphi}{\varphi'}, \text{ where } \varphi' \text{ is any formula obtained from } \varphi \text{ by replacing} \\
& \quad \text{nominals by nominals and variables by rigidified terms.}
\end{aligned}$$

As usual, we say that a proof of a formula  $\varphi$  is a finite sequence of formulas such that every formula in the sequence is either an axiom, or is obtained from previous formula(s) in the sequence using the rules of proof. We write  $\vdash \varphi$  whenever we have such a sequence and say that  $\varphi$  is a  $K_{\tau}$ -theorem. If  $\Gamma \cup \{\varphi\}$  is



a set of formulas, a proof of  $\varphi$  from  $\Gamma$  is a proof of  $\vdash_{K_\tau} (\gamma_1 \wedge \dots \wedge \gamma_n) \rightarrow \varphi$  where  $\{\gamma_1, \dots, \gamma_n\} \subseteq \Gamma$ . A formula  $\varphi$  is provable from a set of formulas  $\Gamma$  (officially written as  $\Gamma \vdash_{K_\tau} \varphi$ , though we will usually just write  $\Gamma \vdash \varphi$  instead) if and only if there is a proof of  $\varphi$  from  $\Gamma$ . The Deduction Theorem holds:  $\Gamma \cup \{\varphi\} \vdash \psi$  iff  $\Gamma \vdash \varphi \rightarrow \psi$

**Proposition 1.** *The following are all  $K_\tau$ -theorems:*

$$\begin{aligned} (K^{-1}_{@}) &\vdash (@_i\varphi \rightarrow @_i\psi) \rightarrow @_i(\varphi \rightarrow \psi) \\ (Nom) &\vdash @_ij \rightarrow (@_i\varphi \rightarrow @_j\varphi) \\ (Sym) &\vdash @_ij \rightarrow @_ji \\ (Bridge) &\vdash @_i\Diamond j \wedge @_j\varphi \rightarrow @_i\Diamond\varphi \\ (Conj) &\vdash @_i(\varphi \wedge \psi) \leftrightarrow (@_i\varphi \wedge @_i\psi) \\ (Elim) &\vdash (i \wedge @_i\varphi) \rightarrow \varphi \end{aligned}$$

*Proof.* See [6].

**Proposition 2.** *The following rules are admissible in  $K_\tau$ :*

$$\begin{aligned} (Name') &\frac{i \rightarrow \varphi}{\varphi}, \text{ where } i \text{ does not occur in } \varphi. \\ (Paste_{\Diamond}) &\frac{(@_i\Diamond j \wedge @_j\varphi) \rightarrow \psi}{@_i\Diamond\varphi \rightarrow \psi}, \text{ if } j \neq i \text{ does not occur in } \varphi \text{ or } \psi. \\ (Paste_{\exists}) &\frac{(@_i\text{EXISTS}(t) \wedge @_i\varphi(\frac{@_it}{x})) \rightarrow \psi}{@_i\exists x\varphi \rightarrow \psi}, t \text{ is ground and does not occur in } \psi. \end{aligned}$$

*Proof.* See [6].

**Corollary 1.** *Let  $\Gamma \cup \{\varphi, \psi\}$  be a set of formulas and  $i, j$  nominals. Then:*

1. *if  $i$  does not occur in  $\Gamma \cup \{\varphi\}$ , then*

$$\Gamma \vdash i \rightarrow \varphi \Rightarrow \Gamma \vdash \varphi.$$

2. *if  $j \neq i$  does not occur in  $\Gamma \cup \{\varphi, \psi\}$ , then*

$$\Gamma \vdash (@_i\Diamond j \wedge @_j\varphi) \rightarrow \psi \Rightarrow \Gamma \vdash @_i\Diamond\varphi \rightarrow \psi.$$

3. *if  $t$  is ground and does not occur in  $\Gamma \cup \{\varphi, \psi\}$ , then*

$$\Gamma \vdash ((@_i\text{EXISTS}(t) \wedge @_i\varphi(\frac{@_it}{x})) \rightarrow \psi) \Rightarrow \Gamma \vdash @_i\exists x\varphi \rightarrow \psi.$$

*Proof.* Immediate from the previous proposition.

## 5 Soundness and Completeness

**Theorem 1 (Soundness).** *Every theorem of  $K_\tau$  is valid. That is, for any formula  $\varphi \in \text{Fm}(\tau)$ , we have that  $\vdash \varphi \Rightarrow \models \varphi$ .*

*Proof.* Fairly straightforward. The Distributivity, Quantifier, Basic Hybrid Axioms and the Axioms for  $\approx$  are all familiar from modal, hybrid, first-order or equational logic. The soundness of  $K_{@_\approx}$ ,  $Nom_{\approx}$ , and  $Agree_{\approx}$  rests on Definitions 4 and 8. Note that *Shuffle-2* also holds in the special case  $i = j$ . If you are unfamiliar with hybrid logic, the soundness of the (*Name*) and (*BG*) rules may not be obvious: they are best thought of as analogous to natural deduction rules (the conclusion of each rule “discharges” a nominal in the premiss) and the side conditions are important. For detailed discussion of both rules (and some variants) see [6].

**Definition 10.** *Let  $\Gamma \subseteq \text{Fm}(\tau)$ .*

- $\Gamma$  is said to be  $K_\tau$ -inconsistent if  $\Gamma \vdash_{K_\tau} \varphi$  for any  $\varphi \in \text{Fm}(\tau)$ . Otherwise we say that  $\Gamma$  is  $K_\tau$ -consistent.
- $\Gamma$  is maximal  $K_\tau$ -consistent if  $\Gamma$  is consistent and any set of formulas that properly extends  $\Gamma$  is  $K_\tau$ -inconsistent.
- $\Gamma$  is named if it contains at least one nominal.
- $\Gamma$  is  $\diamond$ -saturated if for all  $@_i \diamond \varphi \in \Gamma$ , there is a nominal  $j$  such that  $@_i \diamond j$  and  $@_j \varphi$  belong to  $\Gamma$ .
- $\Gamma$  is  $\exists$ -saturated if for all formula  $@_i \exists x \varphi \in \Gamma$  there is a constant  $c$  such that  $@_i(\text{EXISTS}(c)) \in \Gamma$  and  $@_i \varphi \frac{(@_i c)}{x} \in \Gamma$ .

**Lemma 2.** *Let  $\Gamma \subseteq \text{Fm}(\tau)$ . Then*

1.  $\Gamma$  is inconsistent iff there is a formula  $\varphi$  such that  $\Gamma \vdash \varphi$  and  $\Gamma \vdash \neg \varphi$ .
2.  $\varphi \in \Gamma$  then  $\Gamma \vdash \varphi$ .
3.  $\Gamma \cup \{\varphi\}$  is inconsistent iff  $\Gamma \vdash \neg \varphi$ .
4. If  $\Gamma$  is maximal consistent then,  $\Gamma \vdash \varphi \Rightarrow \varphi \in \Gamma$ .

*Proof.* Standard.

We are ready to prove the Lindenbaum lemma we require: every  $K_\tau$ -consistent set of formulas can be extended to a named,  $\diamond$ -saturated,  $\exists$ -saturated, maximal  $K_\tau$ -consistent set.

**Lemma 3 (Lindenbaum).** *Let  $(i_n)_{n \in \mathbb{N}}$  and  $(c_n)_{n \in \mathbb{N}}$  be countably infinite sets of new nominals and new constants, respectively. Let  $\bar{\tau}$  be the new signature obtained by extending  $\Sigma$  and NOM with these symbols, and  $K_{\bar{\tau}}$  the first-order hybrid logic over the extended signature. (Note that by the substitution rule,  $K_{\bar{\tau}}$  is a conservative extension of  $K_\tau$ .) Every  $K_\tau$ -consistent set of formulas  $\Gamma$  can be extended to a named,  $\diamond$ -saturated,  $\exists$ -saturated and maximal  $K_{\bar{\tau}}$ -consistent set.*

*Proof.* Let  $\Gamma$  be a  $K_{\bar{\tau}}$ -consistent set of formulas. We also have  $(i_n)_{n \in \mathbb{N}}$  and  $(c_n)_{n \in \mathbb{N}}$ , countably infinite sets of new nominals and constants respectively, at our disposal. We define the set  $\Gamma^*$  to be  $\bigcup_{n \in \mathbb{N}} \Gamma^n$ , where:

$$\Gamma^0 = \Gamma \cup \{i_0\};$$

$$\Gamma^{n+1} = \begin{cases} \Gamma^n & , \text{ if } \Gamma^n \cup \{\varphi_n\} \text{ is inconsistent} \\ \Gamma^n \cup \{\varphi_n, @_i \diamond i_m, @_{i_m} \psi\} & , \text{ if } \varphi_n = @_i \diamond \psi \text{ and } \Gamma^n \cup \{\varphi_n\} \\ & \text{is consistent} \\ \Gamma^n \cup \{\varphi_n, @_i(\text{EXISTS}(c_m)), @_i \psi \frac{@_i c_m}{x}\} & , \text{ if } \varphi_n = @_i \exists x \psi \text{ and } \Gamma^n \cup \{\varphi_n\} \\ & \text{is consistent,} \\ \Gamma^n \cup \{\varphi_n\} & , \text{ otherwise} \end{cases}$$

In these clauses,  $i_m$  is the first new nominal not occurring in  $\Gamma^n$  or in  $\varphi_n$  and  $c_m$  is the first new constant not in  $\Gamma^n$  or in  $\varphi_n$ . We now prove by induction that  $\Gamma^*$  is  $K_{\bar{\tau}}$ -consistent.

Suppose that  $\Gamma^0$  is not consistent. Then  $\Gamma \cup \{i_0\} \vdash \perp$ . Hence, by the Deduction Theorem,  $\Gamma \vdash i_0 \rightarrow \perp$ . Since  $i_0$  does not occur in  $\Gamma \cup \{\varphi\}$ , by Corollary 1 clause 1,  $\Gamma \vdash \perp$ , which is absurd since  $\Gamma$  is consistent.

Next, assume that  $\Gamma^n$  is  $K_{\bar{\tau}}$ -consistent and consider  $\varphi_n$  of the form  $@_i \diamond \psi$ . Suppose for the sake of a contradiction that  $\Gamma^n \cup \{\varphi_n\}$  is consistent, but that  $\Gamma^{n+1}$  is not. Then  $\Gamma^n \cup \{@_i \diamond \psi, @_i \diamond i_m, @_{i_m} \psi\} \vdash \perp$ . Hence, by the Deduction Theorem,  $\Gamma^n \cup \{@_i \diamond \psi\} \vdash (@_i \diamond i_m \wedge @_{i_m} \psi) \rightarrow \varphi$ . By Corollary 1 clause 2,  $\Gamma^n \cup \{@_i \diamond \psi\} \vdash @_i \diamond \psi \rightarrow \perp$ . Applying modus ponens yields  $\Gamma^n \cup \{@_i \diamond \psi\} \vdash \perp$ , which contradicts our assumption that  $\Gamma^n \cup \{\varphi_n\}$  is consistent.

Next, assume that  $\Gamma^n$  is  $K_{\bar{\tau}}$ -consistent and consider  $\varphi_n$  of the form  $@_i \exists x \psi$ . Suppose for the sake of a contradiction that  $\Gamma^n \cup \{\varphi_n\}$  is consistent, but that  $\Gamma^{n+1}$  is not. This means that

$$\Gamma^n \cup \{\varphi_n, @_i(\text{EXISTS}(c_m)), @_i \psi \frac{@_i c_m}{x}\} \vdash \perp.$$

Then, using the Deduction Theorem, we have that

$$\Gamma^n \cup \{\varphi_n\} \vdash (@_i(\text{EXISTS}(c_m)) \wedge @_i \psi \frac{@_i c_m}{x}) \rightarrow \perp.$$

Then, using Corollary 1 clause 3, we have that

$$\Gamma^n \cup \{\varphi_n\} \vdash @_i \exists x \psi \rightarrow \perp.$$

Thus,  $\Gamma^n \cup \{\varphi_n\} \vdash \perp$ , contradicting its consistency.

Since  $\Gamma^n$  is  $K_{\bar{\tau}}$ -consistent for  $n \in \mathbb{N}$ , it follows that  $\Gamma^* := \bigcup_{n \in \mathbb{N}} \Gamma^n$  is also  $K_{\bar{\tau}}$ -consistent. Moreover,  $\Gamma^*$  is also maximal. For suppose for the sake of a contradiction that it is not: that is, suppose that there exists a formula  $\varphi \notin \Gamma^*$  such that  $\Gamma^* \cup \{\varphi\}$  is  $K_{\bar{\tau}}$ -consistent. Then  $\varphi = \varphi_n$ , for some  $n \in \mathbb{N}$ , and  $\Gamma^n \cup \{\varphi_n\}$  is consistent. Consequently,  $\varphi_n \in \Gamma^{n+1}$  which is an absurd since we assumed that  $\varphi \notin \Gamma^*$ . So  $\Gamma^*$  is maximal, and we have proved our Lindenbaum lemma.

In the sequel, given a  $K_\tau$ -consistent set of formulas  $\Gamma$ ,  $\Gamma^*$  will denote the named,  $\diamond$ -saturated,  $\exists$ -saturated, maximal consistent extension of  $\Gamma$ , defined in the proof of Lemma 3.

**Definition 11.** Let  $\Gamma$  be a named, maximal  $K_\tau$ -consistent set of formulas. Binary relations  $\sim_n$  and  $\sim_r$ , over  $\text{NOM}$  and  $\text{@Term}(\tau)$ , respectively, are defined as follows:

$$\begin{aligned} - i \sim_n j &\Leftrightarrow @_i j \in \Gamma, i, j \in \text{NOM} \\ - t \sim_r t' &\Leftrightarrow t \approx t' \in \Gamma, t, t' \in \text{@Term}(\tau) \end{aligned}$$

**Lemma 4.** The relations  $\sim_n$  and  $\sim_r$  are equivalence relations. Moreover, if  $t_k \sim_r t'_k$  for  $k = 1, \dots, n$ , then  $(@_i f)(t_1, \dots, t_n) \sim_r (@_i f)(t'_1, \dots, t'_n)$ .

*Proof.* The proofs that  $\sim_n$  and  $\sim_r$  are equivalence relations are straightforward (and standard). The proof of the last statement uses the axiom:

$$(t_1 \approx t'_1 \wedge \dots \wedge t_n \approx t'_n) \rightarrow f(t_1, \dots, t_n) \approx f(t'_1, \dots, t'_n),$$

where  $f \in \text{Func}_n \cup \text{@Func}_n$ ,  $n \geq 0$ .

**Definition 12.** Suppose  $\Gamma$  is a named,  $\diamond$ -saturated,  $\exists$ -saturated and maximal  $K_\tau$ -consistent set of formulas. Then the Henkin structure

$$\mathcal{M}^\Gamma = ((W^\Gamma, \text{Dom}^\Gamma, D^\Gamma, R^\Gamma, I^\Gamma)$$

is defined by:

- $W^\Gamma = \{|i| : i \text{ is a nominal}\}$
- $\text{Dom}^\Gamma = \{|t| \in \text{@Term}(\tau) : t \text{ is ground}\}$
- $D^\Gamma_{|i|} = \{|t| \in \text{Dom} : @_i \text{EXISTS}(t) \in \Gamma\}$
- $|i| R^\Gamma |j| \text{ iff } @_i \diamond j \in \Gamma$
- $I^\Gamma_{|i|}(i) = |i|$ , for each nominal  $i$
- for each  $f \in \text{Func}_n$  and  $|t_1|, \dots, |t_n| \in \text{Dom}^\Gamma$ ,  
 $I^\Gamma_{|i|}(f)(|t_1|, \dots, |t_n|) = |(@_i f)(t_1, \dots, t_n)|$
- for each  $P \in \text{Rel}_n$ ,  
 $I^\Gamma_{|i|}(P) = \{|t_1|, \dots, |t_n| \in \text{Dom}^\Gamma : (@_i P)(t_1, \dots, t_n) \in \Gamma\}$

Let us briefly check this definition. Note that  $R^\Gamma$  is well defined. For suppose  $i' \in |i|$ , then  $@_i i' \in \Gamma$  so, if  $@_i \diamond j \in \Gamma$ , by (*Nom*),  $@_{i'} \diamond j \in \Gamma$ . Now suppose  $j' \in |j|$ , then  $@_j j' \in \Gamma$  so, if  $@_i \diamond j \in \Gamma$ , by (*Bridge*),  $@_i \diamond j' \in \Gamma$ . We leave the reader to check that the functions and predicate interpretations are well-defined as well.

With this done, we are ready to state the Truth Lemma which establishes that the Henkin structure  $\mathcal{M}^\Gamma$  is the model we are looking for; the Truth Lemma is stated and proved in the appendix. This leads to:

**Theorem 2 (Completeness).** Let  $\tau$  be a first-order hybrid similarity type  $\varphi$  be a sentence and  $\Gamma$  a set of sentences. Then

$$\Gamma \vdash \varphi \Rightarrow \Gamma \vDash \varphi.$$

## 6 Conclusions and future work

We want to view hybrid logic as a general language of rigidification, and use it to explore conceptual and technical issues in first-order modal logic; this paper is our first step in this direction. The completeness result just proved takes us closer to this goal, because it covers not merely the basic logic, but also completeness with respect to any extension obtained by adding pure axioms or existential saturation rules (for a definition and detailed discussion of these concepts, see [6]). Adding pure axioms automatically yields completeness for many different frame conditions (for example, transitivity, reflexivity, and irreflexivity), and for additional modalities (such as the Priorean tense operators and the universal modality). More importantly for present purposes, such tools also immediately yield completeness for conditions of particular relevance to *first-order* modal logic. To give three such examples from [6], adding the pure axiom

$$\@_i\text{EXISTS}(\@_k c) \rightarrow \@_j\text{EXISTS}(\@_k c)$$

gives us a complete axiomatisation for constant domain (possibilist) semantics, adding

$$\@_i\text{EXISTS}(\@_k c) \wedge \@_j\text{EXISTS}(\@_k c) \rightarrow \@_i j$$

gives us completeness with respect to the condition that all local domains be disjoint, and adding the existential saturation rule

$$\text{if } \vdash \@_i\text{EXISTS}(\@_j c) \rightarrow \varphi \text{ then } \vdash \varphi,$$

where  $i$  is a nominal distinct from  $j$  not occurring in  $\varphi$ , gives us completeness with respect to the class of models in which every object in the domain (that is: every element of Dom) is also an element of some local domain. Thus the system defined in this paper already achieves a reasonable degree of generality.

But there are a number of issues that should be explored further. In this paper we have taken a minimalist approach to rigidification syntax. In particular, we did not have expressions of the form  $\@_i t$  in the *object* language, we instead used such expressions as metalinguistic abbreviations for the rigidification of  $t$  at  $i$  (as defined in Definition 4). However, having explored this minimal choice, we are now experimenting with extended versions of the language in which all such expressions are part of the object language. This seems useful for at least two reasons.

First, we want to develop and axiomatize richer forms of rigid first-order hybrid logic which incorporate the  $\downarrow$  binder. This binder is a standard tool in hybrid logic: it binds nominals to the world of evaluation. For example,  $\downarrow i. \diamond \neg i$  is a formula that is true at any world  $w$  in any model if and only if  $w$  is an irreflexive world. Now, when we add the  $\downarrow$  binder to the language explored in this paper, it will let us bind the nominals in rigidified function and predicate symbols. This is an extension worth exploring, but it seems more interesting to add  $\downarrow$  to an extended version of the language in which *all* expressions of the form  $\@_i t$  are available. Why? Because once we add  $\downarrow$ , it seems both natural and

desirable to be able to form arbitrary terms of the form  $\downarrow i.@_i t$ , and this of course requires that we have all terms of the form  $@_i t$  available (and open for binding) in the object language.

Similar remarks apply to the other extension we are exploring: a general treatment of *partial* functions in a varying domain setting. We have explored this combination of ideas in the setting of higher-order hybrid logic [9], and are currently transferring the key ideas down to the first-order setting defined in this paper. Because the standard hybrid logical results concerning pure axioms and existential saturation rules still hold in our approach to partiality (which draws on ideas due to William Farmer [7]), we are confident that this can be done smoothly, and that the result will be a general first-order modal framework for working with partiality in an actualist semantics. But, once again, it seems that this extension may be more usefully carried out in a language in which all expressions of the form  $@_i t$  are available at the object-level. In this paper we have interpreted function symbols in a way that lets them take as input entities that do not exist at the local domain, and to output non-local entities as well. But this hard-wires a lot into the semantics. We hope to find a flexible language in which a wide variety of choices about the semantics of functions (and predicates) can simply be axiomatised using such standard hybrid tools such as pure axioms and existential saturation rules. Partialising the semantics Farmer-style, adding object-level expressions of the form  $@_i t$ , and exploring the impact of  $\downarrow$ , seems a promising route to such a system.

**Acknowledgements.** The authors are grateful to the Spanish Ministerio de Economía y Competitividad for funding the project *Intensionality as a unifier: Logic, Language and Philosophy*, FFI2017-82554, hosted by the Universidad de Salamanca. Patrick Blackburn would also like to thank the Danish Council for Independent Research (FKK) for funding as part of the project: *The Primacy of Tense: A. N. Prior Now and Then*. Manuel Martins was also supported by ERDF, through the COMPETE 2020 Programme, and by FCT, within the projects POCI-01-0145-FEDER-016692 and UID/MAT/04106/2019.

## References

1. Carlos Areces, Patrick Blackburn, Antonia Huertas, and María Manzano. Completeness in hybrid type theory. *Journal of Philosophical Logic*, 43:209–238, 2014.
2. Carlos Areces, Patrick Blackburn, and Maarten Marx. Repairing the interpolation theorem in quantified modal logic. *Annals of Pure and Applied Logic*, 124(1-3):287–299, 2003.
3. Luís S. Barbosa, Manuel A. Martins, and Marta Carreteiro. A hilbert-style axiomatisation for equational hybrid logic. *Journal of Logic, Language and Information*, 23(1):31–52, Mar 2014.
4. Patrick Blackburn, Antonia Huertas, María Manzano, and Klaus Frovin Jørgensen. Henkin and hybrid logic. In *The Life and Work of Leon Henkin*, pages 279–306. Springer, 2014.

5. Patrick Blackburn and Maarten Marx. Tableaux for quantified hybrid logic. In U. Egly and C. Fernmüller, editors, *Automated Reasoning with Analytic Tableaux and Related Methods, International Conference, TABLEAUX 2002*, pages 38–52. Copenhagen, Denmark, 2002.
6. Patrick Blackburn and Balder ten Cate. Pure extensions, proof rules, and hybrid axiomatics. *Studia Logica*, 84:277–322, 2006.
7. William M Farmer. A partial functions version of Church’s simple theory of types. *The Journal of Symbolic Logic*, 55(3):1269–1291, 1990.
8. Melvin Fitting and Richard Mendelsohn. *First-Order Modal Logic*. Springer, 1998.
9. Maria Manzano, Antonia Huertas, Patrick Blackburn, and Manuel Martins. Hybrid partial type theory. Submitted, 2019.
10. Maria Manzano, Manuel Martins, and Antonia Huertas. Completeness in equational hybrid propositional type theory. *Studia Logica*, Oct 2018.

## Appendix

This appendix sketches the definitions and lemmas that lead to the Truth Lemma, and thus to the Completeness Theorem stated in the main text. As a first step, given an assignment function  $g$  on the Henkin structure  $\mathcal{M}^F$  defined in Definition 12, we need an inductive definition of how to substitute a suitable rigid term for a variables inside terms and formulas; the substitution syntactically mirrors the assignment function.

We do so as follows. Given a variable assignment  $g$  into  $\mathcal{M}^F$  (that is,  $g : X \rightarrow \text{Dom}^F$ ) we first define a substitution function  $\hat{g} : X \rightarrow @\text{Term}(\tau)$  in the following way: for any variable  $x$ , we define  $x^{\hat{g}} := t_k$ , where  $t_k$  is the first rigid ground term in  $@\text{Term}(\tau)$  with lowest  $k$  such that  $g(x) = |t_k|$ . Here we assume that  $@\text{Term}(\tau)$  is ordered. We extend  $\hat{g}$  to arbitrary terms  $t$  by defining: if  $t = f(t_1, \dots, t_n)$  then  $t^{\hat{g}} = f(t_1^{\hat{g}}, \dots, t_n^{\hat{g}})$ .

We extend  $\hat{g}$  to formulas in the following way:

- $i^{\hat{g}} := i$ ,  $i \in \text{NOM}$
- $(t_1 \approx t_2)^{\hat{g}} := (t_1^{\hat{g}} \approx t_2^{\hat{g}})$ ,  $t_1, t_2 \in \text{Term}(\tau)$
- $(P(t_1, \dots, t_n))^{\hat{g}} := P(t_1^{\hat{g}}, \dots, t_n^{\hat{g}})$ ,  $P \in \text{Rel}_n \cup @\text{Rel}_n$  and  $t_1, \dots, t_n \in \text{Term}(\tau)$
- $(@_i \varphi)^{\hat{g}} := @_i(\varphi^{\hat{g}})$ ,  $\varphi \in \text{Fm}(\tau)$  and  $i \in \text{NOM}$
- $(\neg \varphi)^{\hat{g}} := \neg(\varphi^{\hat{g}})$  and  $(\diamond \varphi)^{\hat{g}} := \diamond(\varphi^{\hat{g}})$ ,  $\varphi \in \text{Fm}(\tau)$
- $(\varphi \wedge \psi)^{\hat{g}} := \varphi^{\hat{g}} \wedge \psi^{\hat{g}}$  and  $(\varphi \vee \psi)^{\hat{g}} := \varphi^{\hat{g}} \vee \psi^{\hat{g}}$ , for  $\varphi \in \text{Fm}(\tau)$  and  $\psi \in \text{Fm}(\tau)$
- $(\exists x \varphi)^{\hat{g}} := \exists x(\varphi^{\hat{g}_x^x})$ ,  $x \in X$  and  $\varphi \in \text{Fm}(\tau)$ , where  $\hat{g}_x^x = \hat{g} \setminus \{(x, \hat{g}(x))\} \cup \{(x, x)\}$

For any  $t \in \text{Term}(\tau)$  and any assignment  $g$  on  $\mathcal{M}^F$ , in what follows we will simply write  $t^g$  for  $t^{\hat{g}}$ . A similar simplification will be adopted for formulas.

**Lemma 5.** *For any  $t \in \text{Term}(\tau)$  and any assignment  $g$  on  $\mathcal{M}^F$  we have*

$$[t]^{\mathcal{M}^F, |i|, g} = |@_i t^g|$$

*Proof.* By induction on term structure.

( $t \in X$ )

$$\begin{aligned} [x]^{\mathcal{M}^F, |i|, g} &= g(x) \\ &= |t_k|, \text{ where } t_k \text{ is the first ground (and rigid) term in} \\ &\quad \text{@Term}(\tau) \text{ with lowest } k \text{ such that } g(x) = |t_k|. \\ &= |@_i t_k|, \text{ since } t_k \in \text{@Term}(\tau), \text{ by definition } @_i t_k = t_k \\ &= |@_i x^g| \end{aligned}$$

( $t = f(t_1, \dots, t_n), f \in \text{Func}_n, n \geq 0$ )

$$\begin{aligned} [f(t_1, \dots, t_n)]^{\mathcal{M}^F, |i|, g} &= I_{|i|}(f)([t_1]^{\mathcal{M}, |i|, g}, \dots, [t_n]^{\mathcal{M}, |i|, g}) \\ &= I_{|i|}(f)(|@_i t_1^g|, \dots, |@_i t_n^g|) \quad (\text{Ind. Hyp.}) \\ &= |(@_i f)(@_i t_1^g, \dots, @_i t_n^g)| \\ &= |@_i(f(t_1^g, \dots, t_n^g))| \\ &= |@_i t^g| \end{aligned}$$

( $t = (@_j f)(t_1, \dots, t_n), f \in \text{Func}_n, n \geq 0$ )

$$\begin{aligned} [(@_j f)(t_1, \dots, t_n)]^{\mathcal{M}^F, |i|, g} &= I_{|j|}(f)([t_1]^{\mathcal{M}, |i|, g}, \dots, [t_n]^{\mathcal{M}, |i|, g}) \\ &= I_{|j|}(f)(|@_i t_1^g|, \dots, |@_i t_n^g|) \quad (\text{Ind. Hyp.}) \\ &= |(@_j f)(@_i t_1^g, \dots, @_i t_n^g)| \\ &= |@_i((@_j f)(t_1^g, \dots, t_n^g))| \\ &= |@_i t^g| \end{aligned}$$

**Lemma 6 (Truth Lemma).** *For every nominal  $i$ , any assignment  $g$  on  $\mathcal{M}^F$  and every formula  $\varphi$*

$$\mathcal{M}^F, |i|, g \models \varphi \Leftrightarrow @_i \varphi^g \in \Gamma$$

*Proof.* The proof proceeds by induction on the complexity of  $\varphi$ .

–  $\varphi = j$

We have that

$$\mathcal{M}^{\Gamma^*}, |i|, g \models j \text{ iff } |i| = |j| \text{ iff } @_i j \in \Gamma \text{ iff } @_i j^g \in \Gamma.$$

–  $\varphi = t_1 \approx t_2$ ,

$$\begin{aligned} \mathcal{M}^F, |i|, g \models t_1 \approx t_2 &\text{ iff } [t_1]^{\mathcal{M}, |i|, g} = [t_2]^{\mathcal{M}, |i|, g} \\ &\text{ iff } |@_i t_1^g| = |@_i t_2^g|, \text{ by Lemma 5} \\ &\text{ iff } @_i t_1^g \sim_r @_i t_2^g \\ &\text{ iff } @_i t_1^g \approx @_i t_2^g \in \Gamma \\ &\text{ iff } @_i(t_1^g \approx t_2^g) \in \Gamma, \text{ by axiom } K_{@ \approx} \\ &\text{ iff } @_i(t_1 \approx t_2)^g \in \Gamma \end{aligned}$$

–  $\varphi = P(t_1, \dots, t_n)$ , with  $P \in \text{Rel}_n \cup \text{@Rel}_n$  and  $t_1, \dots, t_n \in \text{Term}(\tau)$ ;

If  $P \in \text{Rel}_n$ :

$$\begin{aligned} \mathcal{M}^F, |i|, g \models P(t_1, \dots, t_n) &\text{ iff } ([t_1]^{\mathcal{M}, |i|, g}, \dots, [t_n]^{\mathcal{M}, |i|, g}) \in I_{|i|}(P) \\ &\text{ iff } (|@_i t_1^g|, \dots, |@_i t_n^g|) \in I_{|i|}(P), \text{ by Lemma 5} \\ &\text{ iff } (@_i P)(@_i t_1^g, \dots, @_i t_n^g) \in \Gamma \\ &\text{ iff } @_i(P(t_1^g, \dots, t_n^g)) \in \Gamma, \\ &\quad \text{by the Shuffle-1 Axiom} \\ &\quad (@_i P)(@_i t_1, \dots, @_i t_n) \leftrightarrow @_i(P(t_1, \dots, t_n)) \\ &\text{ iff } @_i((P(t_1, \dots, t_n))^g) \in \Gamma \end{aligned}$$



If  $P = (@_j S)$ , with  $S \in \text{Rel}_n$ :

$$\begin{aligned} \mathcal{M}^\Gamma, |i|, g \models (@_j S)(t_1, \dots, t_n) &\text{ iff } ([t_1]^{\mathcal{M}, |i|, g}, \dots, [t_n]^{\mathcal{M}, |i|, g}) \in I_{|j|}(S) \\ &\text{ iff } (|\@_i t_1^g|, \dots, |\@_i t_n^g|) \in I_{|j|}(S), \text{ by Lemma 5} \\ &\text{ iff } (@_j S)(\@_i t_1^g, \dots, \@_i t_n^g) \in \Gamma \\ &\text{ iff } \@_i((@_j S)(t_1^g, \dots, t_n^g)) \in \Gamma, \\ &\quad \text{by the Shuffle-2 axiom} \\ &\quad (@_j S)(\@_i t_1, \dots, \@_i t_n) \leftrightarrow \@_i((@_j S)(t_1, \dots, t_n)) \\ &\text{ iff } \@_i((@_j S)(t_1, \dots, t_n)^g) \in \Gamma \end{aligned}$$

–  $\varphi = @_j \psi$ .

$$\begin{aligned} \mathcal{M}^\Gamma, |i|, g \models @_j \psi &\text{ iff } \mathcal{M}^\Gamma, |j|, g \models \psi \\ &\text{ iff } \@_j(\psi)^g \in \Gamma, \text{ IH} \\ &\text{ iff } (@_j \psi)^g \in \Gamma \\ &\text{ iff } \@_i(@_j \psi)^g \in \Gamma, \text{ by } \textit{Agree} \end{aligned}$$

–  $\varphi = \neg \psi$ .

$$\begin{aligned} \mathcal{M}^\Gamma, |i|, g \models \neg \psi &\text{ iff } \mathcal{M}^\Gamma, |i|, g \not\models \psi \\ &\text{ iff } \@_i(\psi)^g \notin \Gamma, \text{ IH} \\ &\text{ iff } \neg \@_i(\psi)^g \in \Gamma, \text{ as } \Gamma \text{ is maximal consistent} \\ &\text{ iff } \@_i(\neg \psi)^g \in \Gamma, \text{ by } \textit{Selfdual}_@ \\ &\text{ iff } \@_i(\neg \psi)^g \in \Gamma \end{aligned}$$

–  $\varphi = \diamond \psi$ .

$$\begin{aligned} \mathcal{M}^\Gamma, |i|, g \models \diamond \psi &\text{ iff there is } j \text{ such that } |i|R^\Gamma |j| \text{ and } \mathcal{M}^\Gamma, |j|, g \models \psi \\ &\text{ iff there is } j \text{ such that } |i|R^\Gamma |j| \text{ and } \@_i \psi^g \in \Gamma, \text{ by IH} \\ &\text{ iff } \@_i \diamond \psi^g \in \Gamma, \\ &\quad \text{by } \textit{Bridge} \text{ (since } \@_i \diamond j \in \Gamma \text{) and } \diamond\text{-saturation} \\ &\text{ iff } \@_i(\diamond \psi)^g \in \Gamma \end{aligned}$$

–  $\varphi = \psi_1 \wedge \psi_2$

$$\begin{aligned} \mathcal{M}^\Gamma, |i|, g \models \psi_1 \wedge \psi_2 &\text{ iff } \mathcal{M}^\Gamma, |i|, g \models \psi_1 \text{ and } \mathcal{M}^\Gamma, |i|, g \models \psi_2 \\ &\text{ iff } \@_i(\psi_1)^g \in \Gamma \text{ and } \@_i(\psi_2)^g \in \Gamma, \text{ IH} \\ &\text{ iff } \@_i(\psi_1)^g \wedge \@_i(\psi_2)^g \in \Gamma, \text{ as } \Gamma \text{ is maximal consistent} \\ &\text{ iff } \@_i((\psi_1)^g \wedge (\psi_2)^g) \in \Gamma \\ &\text{ iff } \@_i(\psi_1 \wedge \psi_2)^g \in \Gamma \end{aligned}$$

–  $\varphi = \exists x \psi$ .

$$\begin{aligned} \mathcal{M}^\Gamma, |i|, g \models \exists x \psi &\text{ iff exists } \theta \in D_{|i|} \text{ s.t. } \mathcal{M}, w, g[x \mapsto \theta] \models \varphi \\ &\text{ iff exists } \theta \in D_{|i|} \text{ s.t. } \@_i \varphi^{g[x \mapsto \theta]} \in \Gamma, \text{ induction hypothesis} \\ &\text{ iff}^* \@_i(\exists x \varphi)^g \in \Gamma \end{aligned}$$

Proof of (\*)

The implication “ $\Rightarrow$ ” holds by the Corollary 1 clause 3.

The implication “ $\Leftarrow$ ” holds by  $\exists$ -saturation.  $@_i(\exists x \varphi)^g \in \Gamma$  implies that there exists a constant  $c$  such that  $@_i \text{EXISTS}(c) \in \Gamma$  and  $(\varphi)^{g_x}(x \mapsto \@_i c) \in \Gamma$ . So there is  $\theta := \@_i c \in D_{|i|}$  (because  $@_i \text{EXISTS}(c) \in \Gamma$ ) s.t.  $@_i \varphi^{g[x \mapsto \theta]} \in \Gamma$ .

**Lemma 7.** *Let  $\Gamma$  be a consistent set of sentences. Then, there is a nominal  $k$  such that for every  $\varphi \in \Gamma$ ,*

$$\mathcal{M}^\Gamma, |k| \models \varphi$$

**Theorem 3 (Completeness).** *Let  $\tau$  be a first-order hybrid similarity type  $\varphi$  be a sentence and  $\Gamma$  a set of sentences. Then*

$$\Gamma \vdash \varphi \Rightarrow \Gamma \models \varphi.$$