ARTICLE IN PRESS
YINCO:104754

Information and Computation ••• (••••) ••••••

Contents lists available at ScienceDirect

Information and Computation

www.elsevier.com/locate/yinco

Exorcising the phantom zone

Patrick Blackburn [a,*], Manuel Martins [b,*], María Manzano [c], Antonia Huertas [d]

[a] IKH, Roskilde University, Denmark
[b] CIDMA & Maths Department, University of Aveiro, Portugal
[c] University of Salamanca, Spain
[d] Universitat Oberta de Catalunya, Spain

ARTICLE INFO

Article history:
Available online xxxx

Keywords:
First-order hybrid logic
Partial function semantics
Rigidification
Denotation

ABSTRACT

In this paper we introduce a language of first-order hybrid logic in which function symbols are interpreted by partial functions and prove a number of completeness results. Syntactically, our language builds on the *basic* propositional hybrid language, has a primitive unary predicate symbol DEN which tests whether a term denotes or not, and permits satisfaction operators to rigidify predicate and function symbols. Semantically, our system is actualist, allows terms to be undefined, and has no truth-value gaps. But should we follow Fitting and Mendelsohn and rule out domain elements not belonging to any world, or should we tolerate them? Here we explore both options. As we shall see, while the choice makes no difference when it comes to validity, it has consequences for richer logics.

1. Introduction

Fitting and Mendelsohn's textbook *First-Order Modal Logic* [1] has deservedly become a standard reference. But given the wide range of topics and tools it discusses (for example, tableau systems, predicate abstraction, and definite descriptions) one detail is easily overlooked: when discussing actualist semantics, the authors insist that the members of the domain of the entire model, or to use their own words, the "things it makes sense to talk about" [1, page 102], must belong to the local domain of some world. That is: their varying domain models are not permitted to contain what we call a phantom zone, a region inhabited by entities beyond the reach of any actualist quantifier. This paper explores the technical consequences of this choice in first-order hybrid logic when function symbols are interpreted by partial functions, and in particular, its consequences for completeness results involving *pure-extensions*.

The basic propositional hybrid language extends propositional modal logic with: (a) special atomic formulas, usually written i, j and k, and called *nominals*, and (b) propositional rigidifiers, usually written $@_i$, $@_j$ and $@_k$, and called *satisfaction operators*. Nominals must be true at exactly one world in any model; they act as "names" for the unique world they are true at. Satisfaction operators transform arbitrary propositional information φ into rigid propositions; while φ may vary in truth value from world to word, $@_i\varphi$ has the same truth value at *all* worlds, namely the truth value that φ has at the world picked out by the nominal i. The completeness theory of the propositional hybrid language has been intensively studied (see, for example [2–4]). Completeness is typically proved using a variant of the Henkin method for first-order logic, with

* Corresponding authors at: IKH, Roskilde University, Denmark.
E-mail addresses: patrick.rowan.blackburn@gmail.com (P. Blackburn), martins@ua.pt (M. Martins), mara@usal.es (M. Manzano), mhuertass@uoc.edu (A. Huertas).

https://doi.org/10.1016/j.ic.2021.104754

nominals acting as "Henkin witnesses" for $\Diamond$-formulas. Moreover — a central theme in this paper — the model construction allows completeness to be proved not merely for the basic logic (that is: the set of all validities), but for any pure extension (that is: for any axiomatic extension where only nominals are used in the axioms).

In this paper we define a language of first-order hybrid logic in which function symbols are interpreted by partial functions, and prove a number of completeness results. Syntactically, the key features of our language are: (a) it builds on the *basic* propositional hybrid language, (b) it has a primitive unary predicate symbol DEN which tests whether a first-order term t is defined or not, and (c) it permits satisfaction operators to rigidify not only propositions, but also any predicate or function symbol. Semantically, we use a varying domain semantics with the quantifiers ranging over local domains, that allows terms to be undefined without giving rise to truth-value gaps.[1] But one interesting choice remains: do we follow Fitting and Mendelsohn and exorcise the phantom zone ghosts or not? As we shall see, while the choice is irrelevant to the minimal logic, it makes a difference when we turn to richer systems.

This paper stems from our earlier work on higher-order hybrid logic (see [5–9]), which allowed satisfaction operators to rigidify constants of all types. We first used the DEN predicate in [9] where we worked with partial type theory; there, however, we worked with a more powerful hybrid logic in which DEN was definable, whereas in this paper we will work with the *basic* hybrid language. We say more about these choices in the paper's conclusion.

We proceed as follows. In Section 2 we define our language, paying particular attention to the rigidity notation. In Section 3 we define our semantics, in Section 4 we note some results relating rigidity, denotation, existence, and phantom zones, and in Section 5 we show that our rigidification map works as expected. In Section 6 we axiomatize the basic logic, in Section 7 we prove a Lindenbaum lemma, and in Section 8 we prove our basic completeness results. With this done, we turn to pure extensions, and the distinction between models with and without phantom zones begins to bite: Section 9 shows how to prove pure extension results when phantom zones are permitted, and Section 10 shows how to prove them when they are not. Section 11 concludes.

2. Syntactic preliminaries

We start with the underlying first-order language. Suppose we are given a first-order signature, consisting of n-ary function and relation symbols.

Definition 2.1 *(Signatures).* We call a pair $\Sigma = ((\mathrm{Func}_n)_{n\in\mathbb{N}}, (\mathrm{Rel}_n)_{n\in\mathbb{N}})$, where Func_n and Rel_n are countable sets of functional and relational symbols of arity n, a first-order *signature*. We typically write elements of Func_n using symbols like f and h, and write elements of Rel_n using symbols like P and Q, possibly superscripted or subscripted. The indexed elements in either family may be empty, and if they are all empty, we have the *empty signature*. The elements of Func_0 (if any) are called *constants*, and the elements of Rel_0 (if any) are called *propositional symbols*. We usually write constants using symbols like b and c, and propositional symbols using p, q and r, adding superscripts or subscripts as required.

We then add first-order variables and nominals and *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 $@_i$. That is, we generalize the rigidification-of-first-order-constants notation used in [10] (which allowed us to form the rigid constant $(@_i c)$ out of an arbitrary first-order constant c and nominal i) to all the symbols in the first-order signature.[2]

Definition 2.2 *(Similarity types).* A *first-order hybrid similarity type* τ is a tuple $\langle \Sigma, X, \mathrm{Nom}\rangle$ where Σ is first-order signature, X is a countably infinite set of variables, and Nom is a countably infinite set of symbols called *nominals*. We typically write variables as x, y and z, and nominals as i, j and k, adding superscripts or subscripts as required. We assume that X, Nom, and the sets of function and relation symbols in Σ are pairwise distinct. The Nom-rigidification of Σ (with respect to the similarity type τ) is the first-order signature:

$$@\Sigma = ((@\mathrm{Func}_n)_{n\in\mathbb{N}}, (@\mathrm{Rel}_n)_{n\in\mathbb{N}}),$$

where $@\mathrm{Func}_n = \{(@_i f) : i \in \mathrm{Nom},\ f \in \mathrm{Func}_n\}$ and $@\mathrm{Rel}_n = \{(@_i P) : i \in \mathrm{Nom},\ P \in \mathrm{Rel}_n\}$.

These new symbols provide a stock of symbols that carry a syntactic guarantee of rigidity. The symbols in the original first-order signature *don't* carry any such guarantee: an ordinary function symbol f, predicate symbol Q, constant c, or propositional symbol q (in short, any item in Σ) may receive different interpretations in different worlds — that is, all these symbols may be interpreted non-rigidly. However, in the partial function semantics defined in the following section, $(@_i f)$ will be interpreted as a rigid function: at all worlds it will denote the function that f denotes at the world named i.

[1] Thus while we use partial functions in our semantics and have undefined terms, we do not call the logic(s) presented here "partial logics". The systems presented here are probably better described as "negative" or "Russellian" rather than "partial".

[2] This generalization was first presented at the 26th Workshop on Logic, Language, Information and Computation (WoLLIC 2019), and the proceedings paper [11] axiomatizes validity for a language with total function symbols.

Similarly, $(@_j Q)$ will be interpreted as a rigid predicate: at all worlds it will denote the predicate that Q denotes at the world named j.

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

Definition 2.3 *(Terms).* Let $\tau = \langle \Sigma, X, \mathrm{Nom} \rangle$ be a first-order hybrid similarity type. The set of rigid Σ-*terms* over τ, $@\mathrm{Term}(\tau)$, is recursively defined by:

- for any $x \in X$, $x \in @\mathrm{Term}(\tau)$;
- for any $f^@ \in @\mathrm{Func}_n$, and all terms $t_m \in @\mathrm{Term}(\tau)$, where $1 \leq m \leq n$, $f^@(t_1, \ldots, t_n) \in @\mathrm{Term}(\tau)$.

The set of Σ-*terms* over τ, $\mathrm{Term}(\tau)$, is recursively defined by:

- for any $x \in X$, $x \in \mathrm{Term}(\tau)$;
- for any $f \in \mathrm{Func}_n \cup @\mathrm{Func}_n$, and all terms $t_m \in \mathrm{Term}(\tau)$, where $1 \leq m \leq n$, $f(t_1, \ldots, t_n) \in \mathrm{Term}(\tau)$.

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

For example, if f is a one-place function symbol, h is a three-place function symbol, and b and c are constants, then $h(b, f(f(c)), (@_i f)(x))$ is a term. It is not a rigid term (as neither h, b, c nor the first two occurrences of f are rigid) and it is not ground (because it contains the variable x). On the other hand, $(@_k h)((@_i b), (@_i f)((@_k f)(@_k c)), (@_i f)(x))$ is rigid, though not ground. Note that elements of the form $(@_i b)$ and $(@_k c)$, where b and c are constant symbols, are indeed rigid terms (that is, elements of $@\mathrm{Term}(\tau)$) as all such expressions belong to $@\mathrm{Func}_0$ by Definition 2.2. Rigid constants like these will play an important role in our completeness proofs, as we will use them as Henkin-style witness constants to prove a Lindenbaum lemma.

Now for a key definition. Given any nominal (say i) and any term t, we want to map it to a rigid term $\Downarrow_i(t)$ in which any non-rigid subterms in t have been "anchored" to the value they take in the i-world.[3] This rigidification map should respect any guarantees concerning rigidity that t already possesses, so if t contains variables or rigidified function symbols, these should be ignored. But any non-rigid subterms should be rigidified with respect to i. We define the required rigidification map as follows:

Definition 2.4 *(Rigidification map).* Let $\mathrm{Term}(\tau)$ and $@\mathrm{Term}(\tau)$ be the set of terms and the set of rigid terms over some first-order signature. For any nominal i in the signature, the rigidification map $\Downarrow_i$ is the mapping from $\mathrm{Term}(\tau)$ onto $@\mathrm{Term}(\tau)$ recursively defined as follows:

- if $t \in X$, $\Downarrow_i(t) := t$;
- if $t = f(t_1, \ldots, t_n)$ with $f \in @\mathrm{Func}_n$, then $\Downarrow_i(t) := f(\Downarrow_i(t_1), \ldots, \Downarrow_i(t_n))$; and
- if $t = f(t_1, \ldots, t_n)$ with $f \in \mathrm{Func}_n$, then $\Downarrow_i(t) := (@_i f)(\Downarrow_i(t_1), \ldots, \Downarrow_i(t_n))$.

We call $\Downarrow_i(t)$ the rigidification of t at i.

Consider the three clauses in turn. First, the rigidification map 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 already have a syntactic guarantee of rigidity, we ignore it and go on to recursively rigidify its arguments. Third, if the functor prefixing a term is of the form f (which means that we have no syntactic guarantee of rigidity) then we replace the functor f by the rigid form $(@_i f)$ and go on to recursively rigidify its arguments. For example, consider again the term $h(b, f(f(c)), (@_i f)(x))$. Then:

$$\Downarrow_k(h(b, f(f(c)), (@_i f)(x)))) = (@_k h)((@_k b), (@_k f)((@_k f)((@_k c))), (@_i f)(x)).$$

Note that for the special case of constants (functions of arity 0) we have: given a constant c, and a nominal i, the rigidification of c with respect to i is $(@_i c)$. So the base case of the recursion is simply the rigidification-of-first-order-constants used in [10]. Also note that when a term $t \in @\mathrm{Term}(\tau)$ is rigid, the result is simply t itself. That is, for any nominal i in the signature, the map $\Downarrow_i$ is the identity map on $@\mathrm{Term}(\tau)$, so rigidification always maps $\mathrm{Term}(\tau)$ onto $@\mathrm{Term}(\tau)$.

We are now ready to define the formulas of our first-order hybrid language with partial function symbols. Given the terms we have just defined, these are pretty much what might be expected — but there is one further syntactic novelty. We

[3] In the WoLLIC proceedings paper [11] we re-used @ as a metalinguistic symbol, and wrote $@_i t$ instead of $\Downarrow_i(t)$ for the rigidification map. The $\Downarrow$-notation used here is clearer and less confusing.

introduce a special primitive unary predicate called DEN. This is a "test predicate" that decides whether or not terms in our logic *denote* (that is, pick out a domain entity) or are undefined.

Definition 2.5 *(Formulas).* The set $\mathrm{Fm}(\tau)$ of formulas of first-order hybrid logic with partial function symbols is the smallest set such that:

1. $\mathrm{Nom} \subseteq \mathrm{Fm}(\tau)$;
2. $\mathsf{DEN}(t) \in \mathrm{Fm}(\tau)$, for any $t \in \mathrm{Term}(\tau)$
3. $t_1 \approx t_2 \in \mathrm{Fm}(\tau)$, for any $t_1, t_2 \in \mathrm{Term}(\tau)$
4. $P(t_1, \ldots, t_n) \in \mathrm{Fm}(\tau)$, for any $P \in \mathrm{Rel}_n \cup @\mathrm{Rel}_n$ and $t_1, \ldots, t_n \in \mathrm{Term}(\tau)$;
5. if $\varphi \in \mathrm{Fm}(\tau)$ and i is a nominal, then $@_i\varphi \in \mathrm{Fm}(\tau)$;
6. if $\varphi \in \mathrm{Fm}(\tau)$, then $\neg\varphi,\ \Box\varphi \in \mathrm{Fm}(\tau)$;
7. if $\varphi \in \mathrm{Fm}(\tau)$ and $\psi \in \mathrm{Fm}(\tau)$ then $\varphi \wedge \psi \in \mathrm{Fm}(\tau)$ and $\varphi \vee \psi \in \mathrm{Fm}(\tau)$.
8. if $x \in X$ and $\varphi \in \mathrm{Fm}(\tau)$, then $\forall x\varphi \in \mathrm{Fm}(\tau)$.

We use familiar abbreviations: $\Diamond\varphi$ is $\neg\Box\neg\varphi$, $\exists x\varphi$ is $\neg\forall x\neg\varphi$, $\varphi \to \psi$ is $\neg(\varphi \wedge \neg\psi)$, and so on. As is standard in varying domain first-order modal logic, we also define $\mathsf{EXISTS}(t)$ to be $\exists x(x \approx t)$, where x is a variable not occurring in t. We assume the familiar first-order distinction between free and bound variables, and call any formula with no free variables a sentence. We also assume the usual first-order notion of a term t being substitutable for variable x in a formula φ (that is: being substitutable without accidental variable capture).

Here are some examples illustrating our bracketing conventions. Let i and j be nominals, let b and c be constant symbols, and let P be a two-place predicate symbol. Then $P(b, c)$ is a formula, one with no syntactic guarantees of rigidity. $P((@_ib), (@_jc))$ is also a formula, though this time the two constants in it have been rigidified. Similarly, $(@_iP)(b, c)$ is also a formula, though here it is the initial predicate has been rigidified. Indeed, $(@_iP)((@_ib), (@_jc))$ is a formula too, though this time the predicate and both constants have been rigidified. But there are other possibilities. In particular, note that $@_iP(b, c)$ is also a formula: it is the formula $P(b, c)$ preceded by $@_i$. This is *not* the same formula as $(@_iP)(b, c)$, as $@_iP(c, d)$ is *guaranteed* to be a rigid proposition (it will either be true at all worlds or false at all worlds) while $(@_iP)(b, c)$ may vary in truth value from world to world, depending on the interpretation of b and c.

We can sum up our conventions as follows: when we combine $@_i$ with any formula φ (that is: with propositional information) we write the resulting formula as $@_i\varphi$ (that is: with no enclosing brackets). However, when we combine $@_i$ with either a function symbol f, a constant symbol c, or a predicate symbol P of arity ≥ 1, then we write the resulting rigidifications as $(@_if)$, $(@_ic)$ and $(@_iP)$ respectively (that is: with enclosing brackets). In the case of a predicate symbol p of arity 0 (that is: the propositional symbol p) note that it is possible to write either $@_ip$, as this is the result of applying $@_i$ to the non-rigid formula p, or $(@_ip)$, which is the rigidified version of the 0-ary relation symbol p, which is also a formula. Our semantics will ensure that the two formulas $@_ip$ and $(@_ip)$ mean exactly the same thing (they both pick out the same rigid proposition) and we normally use the simpler $@_ip$.

3. Semantics

With the syntactic preliminaries out of the way, we turn to semantics. Our main aims in this section are to define a semantics for our language, to give a semantic definition of rigidity, and to prove some simple results about rigidity, denotation, existence, and phantom zones.

Definition 3.1 *(Skeletons).* A *skeleton* S is a tuple (W, Dom, D, R), where $W \neq \emptyset$ is called the set of worlds, $\mathrm{Dom} \neq \emptyset$ is called the global domain, $D : W \to P(\mathrm{Dom})$ assigns a non-empty *local domain* to each world w (we usually write D_w instead of $D(w)$ for a local domain), and $R \subseteq W^2$ is the accessibility relation between worlds. If S is a skeleton such that $\mathrm{Dom} = \bigcup_{w \in W} D_w$, then we call it an *FM-skeleton* (a Fitting-Mendelsohn skeleton).

We next need to define interpretations in a way that handles partial functions. We shall model partial functions by adding an extra "undefined element" $\star$, and work with total functions which take this special value to signal their undefinedness on certain input.

Definition 3.2 *(First-order hybrid models with partial functions).* Given a skeleton S and a first-order hybrid similarity type τ, choose some new entity $\star$ (that is: some set-theoretic entity not belonging to either S or τ) to act as the "undefined" value. Then a first-order hybrid model with partial functions over S and τ is a pair $\mathcal{M} = (S, I)$, where I is an interpretation function defined by:

1. $I(i) \in W$.
2. For any $f \in \mathrm{Func}_n$, $I_w(f)$ is a total function from Dom^n to $\mathrm{Dom} \cup \{\star\}$.

3. For any $P \in \mathrm{Rel}_n$, $I_w(P) \subseteq \mathrm{Dom}^n$.

If f is an n-ary function symbol, and for all $d_1, \ldots, d_n \in \mathrm{Dom}$, $I_w(f)(d_1, \ldots, d_n) = \star$, then we say that f is *vacuous* at w.

First, note that we are following Fitting and Mendelsohn [1] in locally interpreting function symbols f and relation symbols P in a way that gives them global reach. That is, any n-ary function symbol is interpreted at any world as a function with domain Dom^n that may return domain elements from outside the local domain. Similarly, relational symbols can be interpreted as relations involving both local and non-local domain elements. Second, consider what this definition means for constants: if $n = 0$ then $I_w(f)$ is a total function from $\mathrm{Dom}^0 = \{\emptyset\}$ to $\mathrm{Dom} \cup \{\star\}$. So at any world w, a constant either picks out an element of Dom or is vacuous, which is what we want. But again, note the globality: the element denoted by a constant at w need not be an element of D_w — it may belong to $D_{w'}$ for some $w' \neq w$, and if we are not working with an FM-skeleton, it may even belong to $\mathrm{Dom} \setminus \bigcup_{w \in W} D_w$, the phantom zone. Finally, note that in the first clause we simply wrote $I(i) \in W$ to emphasize that the interpretation of every nominal is the same in every world: that is, $I(i) = I_w(i) = I_{w'}(i)$ for all $w, w' \in W$, and all nominals i.

Definition 3.3. Let $f : A \to B$. $\mathrm{Def}(f)$ is $\{a \in A \mid \exists b \in B : b \neq \star \text{ and } f(a) = b\}$. Applied to a function f in a model, $\mathrm{Def}(f)$ returns the set of inputs that pick out a “genuine entity”, one belonging to the model’s domain. Note that $\mathrm{Def}(I_w(f)) = \emptyset$ means that f is vacuous at w.

Definition 3.4. Let $\mathcal{M} = (S, I)$ be a model and $g : X \to \mathrm{Dom}$ be a variable assignment (that is: g is a function from the set of variables X in the language to the elements in the domain of the model). The interpretation of terms is recursively defined as follows:

- if $t \in X$, $[t]^{\mathcal{M},w,g} = g(t)$;
- if $t \in \mathrm{Func}_0$, $[t]^{\mathcal{M},w,g} = I_w(t)$;
- if $t = (@_i f) \in @\mathrm{Func}_0$, $[t]^{\mathcal{M},w,g} = I_{I(i)}(f)$;
- if $t = f(t_1, \ldots, t_n)$, $f \in \mathrm{Func}_n$, where $n > 0$ then:

$$[t]^{\mathcal{M},w,g} = \begin{cases} I_w(f)([t_1]^{\mathcal{M},w,g}, \ldots, [t_n]^{\mathcal{M},w,g}) & \text{, if } [t_1]^{\mathcal{M},w,g} \neq \star, \ldots, [t_n]^{\mathcal{M},w,g} \neq \star \text{ and} \\ & ([t_1]^{\mathcal{M},w,g}, \ldots, [t_n]^{\mathcal{M},w,g}) \in \mathrm{Def}(I_w(f)) \\ \star & \text{, otherwise.} \end{cases}$$

- if $t = (@_i f)(t_1, \ldots, t_n)$, $f \in \mathrm{Func}_n$,

$$[t]^{\mathcal{M},w,g} = \begin{cases} I_{I(i)}(f)([t_1]^{\mathcal{M},w,g}, \ldots, [t_n]^{\mathcal{M},w,g}) & \text{, if } [t_1]^{\mathcal{M},w,g} \neq \star, \ldots, [t_n]^{\mathcal{M},w,g} \neq \star \text{ and} \\ & ([t_1]^{\mathcal{M},w,g}, \ldots, [t_n]^{\mathcal{M},w,g}) \in \mathrm{Def}(I_{I(i)}(f)) \\ \star & \text{, otherwise.} \end{cases}$$

Note that the only difference between the two clauses for function application is that when interpreting a term of the form $h(t_1, \ldots, t_n)$ at a world w, we interpret the h as $I_w(f)$ when h is a non-rigid function symbol f, and as $I_{I(i)}(f)$ when h is a rigid function symbol $(@_i f)$.

Definition 3.5 *(Variant assignment).* Let $g : X \to \mathrm{Dom}$ be an assignment and let $d \in \mathrm{Dom}$. Then $g[x \mapsto d]$ is the function such that $g[x \mapsto d](x) = d$, and $g[x \mapsto d](y) = g(y)$ for all $y \neq x$.

Definition 3.6 *(Satisfaction).* Let $\mathcal{M} = (S, I)$ be a model, let $g : X \to \mathrm{Dom}$ be an assignment, and let $w \in W$. Then we define:

$$\begin{array}{rll}
\mathcal{M}, g, w \models i & \text{iff} & I(i) = w. \\
\mathcal{M}, w, g \models \mathsf{DEN}(t) & \text{iff} & [t]^{\mathcal{M},w,g} \neq \star. \\
\mathcal{M}, w, g \models t_1 \approx t_2 & \text{iff} & [t_1]^{\mathcal{M},w,g} = [t_2]^{\mathcal{M},w,g}. \\
\mathcal{M}, w, g \models P(t_1, \ldots, t_n) & \text{iff} & [t_1]^{\mathcal{M},w,g} \neq \star, \ldots, [t_n]^{\mathcal{M},w,g} \neq \star \text{ and} \\
 & & ([t_1]^{\mathcal{M},w,g}, \ldots, [t_n]^{\mathcal{M},w,g}) \in I_w(P), \\
 & & \text{for } P \in \mathrm{Rel}_n \text{ and } t_1, \ldots, t_n \in \mathrm{Term}(\tau). \\
\mathcal{M}, w, g \models (@_i P)(t_1, \ldots, t_n) & \text{iff} & [t_1]^{\mathcal{M},w,g} \neq \star, \ldots, [t_n]^{\mathcal{M},w,g} \neq \star \text{ and} \\
 & & ([t_1]^{\mathcal{M},w,g}, \ldots, [t_n]^{\mathcal{M},w,g}) \in I_{I(i)}(P), \\
 & & \text{for } P \in \mathrm{Rel}_n \text{ and } t_1, \ldots, t_n \in \mathrm{Term}(\tau). \\
\mathcal{M}, w, g \models @_i \varphi & \text{iff} & \mathcal{M}, I(i), g \models \varphi. \\
\mathcal{M}, w, g \models \neg \varphi & \text{iff} & \mathcal{M}, w, g \not\models \varphi. \\
\mathcal{M}, w, g \models \varphi \wedge \psi & \text{iff} & \mathcal{M}, w, g \models \varphi \text{ and } \mathcal{M}, w, g \models \psi. \\
\mathcal{M}, w, g \models \Box \varphi & \text{iff} & \text{for all } v, \text{ if } wRv \text{ then } \mathcal{M}, v, g \models \varphi. \\
\mathcal{M}, w, g \models \forall x \varphi & \text{iff} & \text{for all } d \in D_w, \mathcal{M}, w, g[x \mapsto d] \models \varphi.
\end{array}$$

A formula φ is *satisfied* at a world w in a model $\mathcal{M}$ under the assignment g iff $\mathcal{M}, w, g \models \varphi$. It is *valid in a model* $\mathcal{M}$ (notation: $\mathcal{M} \models \varphi$) iff, for every world w and every assignment g, we have that $\mathcal{M}, w, g \models \varphi$. It is *valid on a skeleton* S (notation: $S \models \varphi$) iff, for every world w, every assignment g, and every interpretation I, we have that $(S, I), w, g \models \varphi$. A formula φ is *valid* (notation: $\models \varphi$) if and only if, for every skeleton S we have that $S \models \varphi$. We extend this notation to sets of formulas and classes of skeletons in the familiar way. In particular, $\mathcal{M}, w, g \models \Gamma$ means that for all formulas $\gamma \in \Gamma$, $\mathcal{M}, w, g \models \gamma$, and $\mathsf{S} \models \varphi$ means that φ is valid on every skeleton S from S.

Some remarks. First, note that DEN is indeed a test predicate which signals whether or not a term denotes at a world. Second, note that the logical equality symbol $\approx$ judges two terms to be equal if they both denote the same entity in the model, or if they are both undefined. That is, $\approx$ judges using set-theoretic equality: no matter what set-theoretic entity we chose for $\star$, we have that $\star = \star$. Third, note that for any relational symbol (whether of the form P or $(@_i P)$), a single undefined term guarantees its falsity; this contrasts with the special symbol $\approx$, which is true when both its arguments are undefined. Fourth, note that for any nominal i and any formula φ, we have that $@_i\varphi$ is either true at all worlds or false at all worlds. Fifth, as previously remarked, though based on partial functions, this semantics does not give rise to truth-value gaps.

Definition 3.7 *(Semantic consequence).* Let τ be a hybrid similarity type, S a class of skeletons, Γ a set of formulas, and φ a formula. Then $\Gamma \models_{\mathsf{S}} \varphi$ iff for all models $\mathcal{M} = (S, I)$ where S is from S, all worlds w in $\mathcal{M}$, and all assignments g on $\mathcal{M}$ we have that: if $\mathcal{M}, w, g \models \Gamma$, then $\mathcal{M}, w, g \models \varphi$. If this holds then we say that γ is a *semantic consequence* of Γ *over* S. If S is the class of all skeletons, then we just write $\Gamma \models \varphi$ and say that γ is a semantic consequence of Γ.

We will use pure extensions later in the paper to axiomatize semantic consequence over various classes of skeletons, paying particular attention to the changes required when working with skeletons belonging to FM, the class of all FM-skeletons.

4. Rigidity, denotation, existence, and phantom zones

What does rigidity mean when we work with partial functions? Informally, rigidity means "has the same value in all worlds in a model". It seems natural that "same value" should include the $\star$ value, thus what we will call semantic rigidity means having the same interpretation, including being undefined, at every world. Here are three distinctions we find useful:

Definition 4.1. Suppose we have a model $\mathcal{M}$ and an assignment g on $\mathcal{M}$. Then we say:

- A term t is *semantically rigid* with respect to $\mathcal{M}$ and g iff for all worlds w and w' we have that $[t]^{\mathcal{M},w,g} = [t]^{\mathcal{M},w',g}$.
- A term t *denotes everywhere* with respect to $\mathcal{M}$ and g iff for all worlds w, we have that $\mathcal{M}, w, g \models \mathsf{DEN}(t)$, that is, iff $[t]^{\mathcal{M},w,g} \neq \star$.
- A term t *denotes nowhere* with respect to $\mathcal{M}$ and g iff for all worlds w, we have that $\mathcal{M}, w, g \not\models \mathsf{DEN}(t)$, that is, iff $[t]^{\mathcal{M},w,g} = \star$.

In any model, with respect to any variable assignment, variables are both semantically rigid and denote everywhere. Note that there are many terms that denote nowhere: if c is interpreted as $\star$ at the i-world in some model (that is: $I_{I(i)}(c) = \star$) then $(@_i c)$ denotes nowhere in that model, and is semantically rigid. We next show that terms *syntactically* guaranteed to be rigid, are indeed *semantically* rigid:

Lemma 4.2. *If t is a term in $@\mathrm{Term}(\tau)$, then t is semantically rigid in any model $\mathcal{M}$, under any variable assignment g on $\mathcal{M}$.*

Proof. We will show by induction on term structure that for any $t \in @\mathrm{Term}(\tau)$, any model $\mathcal{M}$, any assignment g on $\mathcal{M}$ and any $w, w' \in W$ we have that:

$$[t]^{\mathcal{M},w,g} = [t]^{\mathcal{M},w',g}.$$

Base case: $t \in X$. Then $[x]^{\mathcal{M},w,g} = g(x) = [x]^{\mathcal{M},w',g}$.

Inductive step: $t = (@_j f)(t_1, \ldots, t_n)$, where $f \in \mathrm{Func}_n$, $n \geq 0$ and $t_m \in @\mathrm{Term}(\tau)$, where $1 \leq m \leq n$. Then we have that:

$$\begin{aligned}[(@_j f)(t_1, \ldots, t_n)]^{\mathcal{M},w,g} &= I_{I(j)}(f)([t_1]^{\mathcal{M},w,g}, \ldots, [t_n]^{\mathcal{M},w,g}) \\ &= I_{I(j)}(f)([t_1]^{\mathcal{M},w',g}, \ldots, [t_n]^{\mathcal{M},w',g}) \quad \text{(by IH)} \\ &= [(@_j f)(t_1, \ldots, t_n)]^{\mathcal{M},w',g}.\end{aligned}$$

This establishes the result. □

How are "denotes" and "exists" related? Recall that we defined $\mathsf{EXISTS}(t)$ to be $\exists x(x \approx t)$. This is the standard definition in ordinary (total function based) first-order varying-domain modal logic, and it is easy to see that under our partial function based semantics we still have that $\mathsf{EXISTS}(t)$ is true at a world w iff t is interpreted in w by a domain entity that belongs to D_w. Thus if $\mathsf{EXISTS}(t)$ is true at w, then t is not interpreted by $\star$ in w, hence $\mathsf{DEN}(t)$ is true at w too. So:

$$\models \mathsf{EXISTS}(t) \to \mathsf{DEN}(t).$$

It is also easy to see that the converse implication does *not* hold: $\mathsf{DEN}(t)$ may be true at w because t is interpreted at w as an entity d that does not belong to D_w, but to D'_w, the local domain of a different world w'. This suffices to falsify $\mathsf{DEN}(t) \to \mathsf{EXISTS}(t)$ at w.

But there is another way to falsify $\mathsf{DEN}(t) \to \mathsf{EXISTS}(t)$. As we have defined it, $\mathsf{DEN}(t)$ does *not* mean "exists at some local domain in the model", it simply tests whether t is $\star$ or not. So to make the antecedent true we could instead have interpreted t at w as an entity $d \in \mathrm{Dom}$ that does not exist at D_w, or at any other local domain $D_{w'}$; as we do not have to work with FM-skeletons, our models can be based on skeletons containing a non-empty phantom zone $\mathrm{Dom}\setminus \bigcup_{w\in W} D_w$.

It is easy to see that if we are only concerned with validity, it makes no difference whether there is a non-empty phantom zone or not. Recall that FM is the class of all FM-skeletons. Trivially, if φ is valid, then it is valid on every FM-skeleton. We now show that the converse is true as well.

Definition 4.3. Let $S = (W, \mathrm{Dom}, D, R)$ be a skeleton where $Z = (\mathrm{Dom}\setminus \bigcup_{w\in W} D_w) \neq \emptyset$, and let I be an interpretation on S. Choose some new entity z and define:

1. S^z is the skeleton $(W \cup \{z\}, \mathrm{Dom}, D^z, R)$, where $D^z = D \cup \{(z, Z)\}$.
2. I^z is any extension of interpretation I that supplies values for the function symbols f and relation symbols P at the new world z.

Any such model (S^z, I^z) is called a *phantom world* version of (S, I).

Lemma 4.4. *Let (S, I) be a model with non-empty phantom zone Z, and let (S^z, I^z) be any phantom world version of (S, I). Then for any assignment g on S, any world w in S, and any term t:*

$$[t]^{(S,I),w,g} = [t]^{(S^z,I^z),w,g}.$$

Furthermore, for any formula φ:

$$(S, I), w, g \models \varphi \textit{ iff } (S^z, I^z), w, g \models \varphi.$$

Proof. Note that as w is a world in S, $w \neq z$. Furthermore, note that any variable assignment g on S is also an assignment on S^z. The results follow by an easy induction — we have simply gathered the entities in the phantom zone Z into a special world z. $\square$

It follows that if we can falsify a formula φ on an arbitrary skeleton, then we can also falsify it on an FM-skeleton; hence if a formula is valid on every skeleton in FM, then it is valid on every skeleton, so validity and FM-validity coincide. But it is not always so easy to exorcise the phantom zone, as we shall show later in the paper.

5. Three useful lemmas

We are ready to turn to axiomatization and completeness, but first we note three lemmas that will be useful later. The first lemma, and the corollary that follows, are familiar from many other kinds of logic:

Lemma 5.1 *(Coincidence lemma). Let S be a skeleton. Then:*

1. *Let I and I' be interpretations that agree on all symbols in term t. Then, for any assignment g and world w in S: $[t]^{(S,I),w,g} = [t]^{(S,I'),w,g}$.*
2. *Let g and g' be assignments that agree on all variables in term t. Then, for any interpretation I and world w in S: $[t]^{(S,I),w,g} = [t]^{(S,I),w,g'}$.*
3. *Let I and I' be interpretations that agree on all symbols in formula φ. Then, for any assignment g and world w in S: $(S, I), w, g \models \varphi$ iff $(S, I'), w, g \models \varphi$.*
4. *Let g and g' be assignments that agree on all free variables in formula φ. Then, for any interpretation I and world w in S: $(S, I), w, g \models \varphi$ iff $(S, I), w, g' \models \varphi$.*

As a corollary we have that for sentences φ, in any model $\mathcal{M}$ at any world w, we can simply write $\mathcal{M}, w \models \varphi$ rather that $\mathcal{M}, w, g \models \varphi$, because clauses 2 and 4 imply that the choice of g is irrelevant for formulas with no free variables. The next lemma tells us that if all constants and function symbols in some term t are defined at all worlds, then, irrespective of the valuation or the world of evaluation, the term denotes. More precisely:

Lemma 5.2. *Let $t \in \mathrm{Term}(\tau)$, and let (S, I) be a model such that for all function symbols $f \in \mathrm{Func}_n$ in t, and all worlds w in S, we have that $I_w(f)$ is a total function from Dom^n to Dom. Then for all assignments g and worlds w we have $[t]^{(S,I),x,g} \neq \star$.*

Proof. By induction on term structure. □

Lastly, we informally discussed the rigidification map $\Downarrow_i(t)$ in Section 2, but that was before we defined the partial function semantics. So we need to check that:

Lemma 5.3. *For any $t \in \mathrm{Term}(\tau)$ and any assignment g on $\mathcal{M}$ we have*

$$[t]^{\mathcal{M},I(i),g} = [\Downarrow_i(t)]^{\mathcal{M},w,g}, \text{ for any } w \in W$$

Proof. By induction on term structure.

Base case: Suppose $t \in X$. Then $[x]^{\mathcal{M},I(i),g} = g(x) = [x]^{\mathcal{M},w,g} = [\Downarrow_i(x)]^{\mathcal{M},w,g}$.
Inductive step for $t = f(t_1, \ldots, t_n)$, $f \in \mathrm{Func}_n$, $n \geq 0$**:** Suppose $[f(t_1, \ldots, t_n)]^{\mathcal{M},I(i),g} = \star$. This means that:
$[t_1]^{\mathcal{M},I(i),g} = \star$, or … or $[t_n]^{\mathcal{M},I(i),g} = \star$, or $([t_1]^{\mathcal{M},I(i),g}, \ldots, [t_n]^{\mathcal{M},I(i),g}) \notin \mathrm{Def}(I_{I(i)}(f))$.
So by the inductive hypothesis:
$[\Downarrow_i(t_1)]^{\mathcal{M},w,g} = \star$, or … or $[\Downarrow_i(t_n)]^{\mathcal{M},w,g} = \star$, or $([\Downarrow_i(t_1)]^{\mathcal{M},w,g}, \cdots, [\Downarrow_i(t_n)]^{\mathcal{M},w,g} \notin \mathrm{Def}(I_{I(i)}(f))$
Thus $[\Downarrow_i(f(t_1, \ldots, t_n))]^{\mathcal{M},I(i),g} = [(@_i f)(\Downarrow_i(t_1), \ldots, \Downarrow_i(t_1))]^{\mathcal{M},w,g} = \star$ as required.
So suppose instead that $[f(t_1, \ldots, t_n)]^{\mathcal{M},I(i),g} \neq \star$. Then:

$$\begin{aligned}
[f(t_1, \ldots, t_n)]^{\mathcal{M},I(i),g} &= I_{I(i)}(f)([t_1]^{\mathcal{M},I(i),g}, \ldots, [t_n]^{\mathcal{M},I(i),g}) \\
&= I(i)(f)([\Downarrow_i(t_1)]^{\mathcal{M},w,g}, \ldots, [\Downarrow_i(t_n)]^{\mathcal{M},w,g}) \quad \text{(IH)} \\
&= [(@_i f)(\Downarrow_i(t_1), \ldots, \Downarrow_i(t_n))]^{\mathcal{M},w,g} \\
&= [\Downarrow_i((f(t_1, \ldots, t_n)))]^{\mathcal{M},w,g}, \text{ as required.}
\end{aligned}$$

Inductive step for $t = (@_j f)(t_1, \ldots, t_n)$, $f \in \mathrm{Func}_n$, $n \geq 0$**:** If $[(@_j f)(t_1, \ldots, t_n)]^{\mathcal{M},I(i),g} = \star$, the argument is much the same as in the previous step. We handle the other case as follows:

$$\begin{aligned}
[@_j f(t_1, \ldots, t_n)]^{\mathcal{M},I(i),g} &= I_{I(j)}(f)([t_1]^{\mathcal{M},I(i),g}, \ldots, [t_n]^{\mathcal{M},I(i),g}) \\
&= I(j)(f)([\Downarrow_i(t_1)]^{\mathcal{M},w,g}, \ldots, [\Downarrow_i(t_n)]^{\mathcal{M},w,g}) \quad \text{(IH)} \\
&= [(@_j f)(\Downarrow_i(t_1), \ldots, \Downarrow_i(t_n))]^{\mathcal{M},w,g} \\
&= [\Downarrow_i((@_j f)(t_1, \ldots, t_n))]^{\mathcal{M},w,g}, \text{ as required}
\end{aligned}$$

Thus the rigidification maps works as we expect. □

6. The K_τ axiomatisation

We now present K_τ, our basic axiomatisation for first-order hybrid logic with partial function symbols over hybrid similarity type τ. We take all propositional tautologies as axioms, and in addition:

Distributivity axioms
($K_\Box$) $\Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi)$.
($K_@$) $@_i(\varphi \to \psi) \to (@_i\varphi \to @_i\psi)$.

Quantifier axioms
($Q1$) $\forall x(\varphi \to \psi) \to (\varphi \to \forall x\psi)$, where x does not occur free in φ.
($Q2$) $\forall x\varphi \to (\mathsf{EXISTS}(t) \to \varphi(\frac{t}{x}))$, where $t \in @\mathrm{Term}(\tau)$, and t is substitutable for x in φ.
($Q3$) $\exists y \mathsf{EXISTS}(y)$

Basic hybrid axioms
($Ref_@$) $@_i i$.
($Agree$) $@_i @_j \varphi \leftrightarrow @_j \varphi$.
($Selfdual_@$) $@_i\varphi \leftrightarrow \neg @_i \neg\varphi$.
($Intro$) $i \to (\varphi \leftrightarrow @_i\varphi)$.
($Back$) $\Diamond @_i\varphi \to @_i\varphi$.

Axioms for $\approx$
($Ref_\approx$) $t \approx t$, for all $t \in \mathrm{Term}(\tau)$.
($Sym_\approx$) $(t_1 \approx t_2) \to (t_2 \approx t_1)$, for all $t_1, t_2 \in \mathrm{Term}(\tau)$.
($Trans_\approx$) $\big((t_1 \approx t_2) \wedge (t_2 \approx t_3)\big) \to (t_1 \approx t_3)$, for all $t_1, t_2, t_3 \in \mathrm{Term}(\tau)$.
($Func$) $(t_1 \approx t'_1 \wedge \ldots \wedge t_n \approx t'_n) \to f(t_1, \ldots, t_n) \approx f(t'_1, \ldots, t'_n)$,
where $f \in \mathrm{Func} \cup @\mathrm{Func}$, and $t_i, t'_i \in \mathrm{Term}(\tau)$, for $i = 1, \ldots, n$, $n \geq 0$.
($Pred$) $(t_1 \approx t'_1 \wedge \ldots \wedge t_n \approx t'_n) \to P(t_1, \ldots, t_n) \leftrightarrow P(t'_1 \ldots, t'_n)$,
where $P \in \mathrm{Rel} \cup @\mathrm{Rel}$, and $t_i, t'_i \in \mathrm{Term}(\tau)$, for $i = 1, \ldots, n$, $n \geq 0$.

Interactions between @ and ≈

($Rigidify$) $@_i(c \approx (@_i c))$, for any constant c.
($K_{@\approx}$) $@_i(t_1 \approx t_2) \leftrightarrow (\Downarrow_i(t_1) \approx \Downarrow_i(t_2))$, for all $t_1, t_2 \in \mathrm{Term}(\tau)$.
($Nom_{\approx}$) $@_i j \rightarrow (\Downarrow_i(t) \approx \Downarrow_j(t))$, $t \in \mathrm{Term}(\tau)$.
($Agree_{\approx}$) $@_i(t_1 \approx t_2) \leftrightarrow (t_1 \approx t_2)$, for all $t_1, t_2 \in @\mathrm{Term}(\tau)$.

Linking Formula rigidity and predicate-and-term rigidity

($Shuffle$-1) $@_i P(t_1, \ldots, t_n) \leftrightarrow (@_i P)(\Downarrow_i(t_1), \ldots, \Downarrow_i(t_n))$.
($Shuffle$-2) $@_i(@_j P)(t_1, \ldots, t_n) \leftrightarrow (@_j P)(\Downarrow_i(t_1), \ldots, \Downarrow_i(t_n))$.

Denotation axioms

($DenVar$) $\mathsf{DEN}(x)$, $x \in X$
($DenArgF$) $\mathsf{DEN}(f(t_1, \ldots, t_n)) \rightarrow (\mathsf{DEN}(t_1) \wedge \cdots \wedge \mathsf{DEN}(t_n))$
($DenArgP$) $P(t_1, \ldots, t_n)) \rightarrow (\mathsf{DEN}(t_1) \wedge \cdots \wedge \mathsf{DEN}(t_n))$

Other link axioms

($DenRig$) $\mathsf{DEN}(\Downarrow_i(t)) \leftrightarrow @_i \mathsf{DEN}(t)$
($ExRig$) $@_i \mathsf{EXISTS}(\Downarrow_i(t)) \leftrightarrow @_i \mathsf{EXISTS}(t)$
($ExDen$) $\mathsf{EXISTS}(t) \rightarrow \mathsf{DEN}(t)$
($DenDen$) $t_1 \approx t_2 \rightarrow (\mathsf{DEN}(t_1) \leftrightarrow \mathsf{DEN}(t_2))$

Now for the rules of proof. We take modus ponens and the following rule of substitution: if $\vdash \varphi$ then $\vdash \varphi'$, where φ' is any formula obtained from φ' by replacing nominals by nominals, and variables by substitutable rigidified term (that is: we should avoid accidental variable binding when substituting rigidified terms). We also take:

1. **Generalizations:** If $\vdash \varphi$ then $\vdash \Box\varphi$, $\vdash @_i\varphi$, and $\vdash \forall x\varphi$.
2. **Name:** If $\vdash i \rightarrow \varphi$, then $\vdash \varphi$, where i does not occur in φ.
3. **Paste:** If $\vdash @_i \Diamond j \wedge @_j \varphi \rightarrow \theta$ then $\vdash @_i \Diamond \varphi \rightarrow \theta$, where $j \neq i$ does not occur in φ or θ.

The generalization rules are probably familiar to most readers, but as the Name and Paste rules play such an important role in our Lindenbaum construction, the following remarks may be helpful. The Name rule can be read as (something like) a natural deduction rule saying:

If φ can be proved to hold at an arbitrary world named i mentioned in φ, then we can (so to speak) discharge i and prove φ.

The Paste rule is best thought of as (something like) a sequent or tableau rule. If we read it from right-to-left it says:

To prove θ from $@_i \Diamond \varphi$, introduce a new nominal j, and try to prove θ from $@_i \Diamond j$ and $@_j \varphi$ instead.

The Paste rule will be used to license the introduction of Henkin-style witness nominals and rigidified constants in our Lindenbaum construction. For more on these rules, see [3].

We assume the usual definition of formal proof, and write $\vdash \varphi$ to indicate that φ is provable in K_τ, and say that φ is a K_τ-theorem. For any set of K_τ-formulas Γ, we write $\Gamma \vdash \varphi$ to indicate that for some finite subset $\{\gamma_1, \ldots, \gamma_n\}$ of Γ, we have that $\vdash (\gamma_1 \wedge \cdots \wedge \gamma_n) \rightarrow \varphi$.

Proposition 6.1. *The following is a list of K_τ theorems:*

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

Proof. These are standard; see [3] for axiomatic proofs. □

Proposition 6.2. *The following rule, Paste$_\forall$, is derivable in K_τ:*

$$\frac{\vdash (@_i \mathsf{EXISTS}(c) \wedge @_i\varphi(\frac{(@_i c)}{x})) \rightarrow \psi}{\vdash @_i \exists x\varphi \rightarrow \psi}, \text{ where } c \text{ does not occur in } \psi.$$

Proof. In effect this licences the use of first-order Henkin witnesses. See [3] for further discussion and the derivability proof. □

Theorem 6.3 *(Soundness). Every K_τ-theorem is valid: that is, for any formula $\varphi \in \mathrm{Fm}(\tau)$, we have that $\vdash \varphi \Rightarrow \vDash \varphi$.*

Proof. The rules of proof all preserve validity, so proving soundness boils down to checking the validity of the axioms. Here are some examples.

($K_{@\approx}$) On the one hand:
$\mathcal{M}, w, g \vDash @_i(t_1 \approx t_2)$ iff $\mathcal{M}, I(i), g \vDash t_1 \approx t_2$ iff $[t_1]^{\mathcal{M},I(i),g} = [t_2]^{\mathcal{M},I(i),g}$
On the other hand, with the help of Lemma 4.2 we have:
$\mathcal{M}, w, g \vDash \Downarrow_i(t_1) \approx \Downarrow_i(t_2)$ iff $[\Downarrow_i(t_1)]^{\mathcal{M},w,g} = [\Downarrow_i(t_2)]^{\mathcal{M},w,g}$ iff $[t_1]^{\mathcal{M},I(i),g} = [t_2]^{\mathcal{M},I(i),g}$.

($Nom_\approx$) Suppose that $\mathcal{M}, w, g \vDash @_i j$. That is, $I(i) = I(j)$. Then, $[t]^{\mathcal{M},I(i),g} = [t]^{\mathcal{M},I(j),g}$. So, again using Lemma 4.2, $[\Downarrow_i(t)]^{\mathcal{M},w,g} = [\Downarrow_j(t)]^{\mathcal{M},w,g}$. Therefore, $\mathcal{M}, w, g \vDash \Downarrow_i(t) \approx \Downarrow_j(t)$

(*Shuffle*-1)

$$\begin{array}{lll}
\mathcal{M}, w, g \vDash @_i P(t_1, \dots, t_n) & \text{iff} & \mathcal{M}, I(i), g \vDash P(t_1, \dots, t_n) \\
& \text{iff} & [t_1]^{\mathcal{M},I(i),g} \neq \star, \dots, [t_n]^{\mathcal{M},I(i),g} \neq \star \text{ and} \\
& & ([t_1]^{\mathcal{M},I(i),g}, \dots, [t_n]^{\mathcal{M},I(i),g}) \in I_{I(i)}(P) \\
& \text{iff} & [\Downarrow_i(t_1)]^{\mathcal{M},w,g} \neq \star, \dots, [\Downarrow_i(t_n)]^{\mathcal{M},w,g} \neq \star \text{ and} \\
& & ([\Downarrow_i(t_1)]^{\mathcal{M},w,g}, \dots, [\Downarrow_i(t_n)]^{\mathcal{M},w,g}) \in I_{I(i)}(P) \\
& \text{iff} & \mathcal{M}, w, g \vDash P(\Downarrow_i(t_1), \dots, \Downarrow_i(t_n))
\end{array}$$

(*DenRig*)

$$\begin{array}{lll}
\mathcal{M}, w, g \vDash \mathsf{DEN}(\Downarrow_i(t)) & \text{iff} & [\Downarrow_i(t)]^{\mathcal{M},w,g} \neq \star \\
& \text{iff} & [t]^{\mathcal{M},I(i),g} \neq \star \\
& \text{iff} & \mathcal{M}, I(i), g \vDash \mathsf{DEN}(t) \\
& \text{iff} & \mathcal{M}, w, g \vDash @_i\mathsf{DEN}(t)
\end{array}$$

Note that Lemma 4.2 was also needed for both the (*Shuffle*-1) and (*DenRig*) proof, at steps 3–4, and 1–2, respectively. □

7. A Lindenbaum lemma

We now prove the Lindenbaum lemma we need. First some background concepts and notation.

Definition 7.1. Let $\Gamma \subseteq \mathrm{Fm}(\tau)$.

- Γ is *K_τ-inconsistent* iff $\Gamma \vdash \varphi$ for any $\varphi \in \mathrm{Fm}(\tau)$, otherwise Γ is *K_τ-consistent*. When it is clear from context, we sometimes drop the K_τ and simply say consistent and inconsistent.
- Γ is *maximal* K_τ-consistent iff Γ is consistent and any set of formulas that properly extends Γ is inconsistent. We often call such a set an MCS.
- Γ is *named* iff it contains at least one nominal.
- Γ is *$\Diamond$-saturated* iff for all $@_i\Diamond\varphi \in \Gamma$, there is a nominal j such that $@_i\Diamond j$ and $@_j\varphi$ belongs to Γ (note the similarity of this to the Paste rule, and note that j is essentially a "Henkin witness nominal" for the $\Diamond$).
- Γ *is $\exists$-saturated* iff for all formula $@_i\exists x\varphi \in \Gamma$ there is a constant c such that $@_i\mathsf{EXISTS}(c) \in \Gamma$ and $@_i\varphi(\frac{(@_ic)}{x}) \in \Gamma$. Here $\varphi(\frac{(@_ic)}{x})$ is the formula obtained by substituting $(@_ic)$ for all free occurrences of x in φ (note the similarity to this to the Paste$_\forall$ rule of Proposition 6.2, and note that $(@_ic)$ is essentially a "rigid Henkin witness constant" for the $\exists$).

Lemma 7.2. *Let $\Gamma \subseteq \mathrm{Fm}(\tau)$. Then:*

1. *$\Gamma \cup \{\varphi\}$ is inconsistent iff $\Gamma \vdash \bot$.*
2. *If Γ is maximal consistent then: if $\Gamma \vdash \varphi$ then $\varphi \in \Gamma$.*
3. *$\Gamma \cup \{\varphi\} \vdash \psi$ iff $\Gamma \vdash \varphi \to \psi$; that is, the deduction theorem (DDT) holds.*

Proof. Standard. Note that DDT holds because θ is defined to be a logical consequence of Γ iff $\vdash \gamma \to \theta$, where γ is a conjunction of (finitely many) formulas from Γ. And in propositional logic we have that $\vdash (\varphi \wedge \gamma_1 \wedge \cdots \wedge \gamma_n) \to \psi$ iff $\vdash (\gamma_1 \wedge \cdots \wedge \gamma_n) \to (\varphi \to \psi)$. □

Lemma 7.3 *(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, and let $\overline{\tau}$ be the hybrid similarity type obtained by extending* Nom *and Σ with these new symbols. Then $K_{\overline{\tau}}$ is the basic axiomatisation for first-order hybrid logic with partial function symbols over hybrid similarity type $\overline{\tau}$. Every K_τ-consistent set of formulas Γ can be extended to a named, $\Diamond$-saturated, $\exists$-saturated and maximal $K_{\overline{\tau}}$-consistent set.*

Proof. Let Γ be a K_τ-consistent set of formulas and 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. We define Γ^* to be $\bigcup_{n\in\mathbb{N}} \Gamma^n$, where:

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

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

Here i_m is the first new nominal not occurring in Γ^n or in φ_n and c_m is the first new constant not in Γ^n or in φ_n.

We first prove by induction that Γ^* is $K_{\overline{\tau}}$-consistent. So suppose that Γ^0 is inconsistent. Then $\Gamma \cup \{i_0\} \vdash \bot$. So for some finite conjunction γ of formulas from Γ, we have $\vdash \gamma \to \neg i_0$, and hence, $\vdash i_0 \to \neg\gamma$. But i_0 is new, so it does not occur in γ, so we apply the Name rule obtaining $\vdash \neg\gamma$, contradicting the consistency of Γ. We conclude that $\Gamma^0 = \Gamma \cup \{i_0\}$ is consistent after all.

Suppose now that Γ^n is $K_{\overline{\tau}}$-consistent, and consider φ_n of the form $@_i\Diamond\psi$. We know that $\Gamma^n \cup \{@_i\Diamond\psi\}$ is consistent; let us suppose that Γ^{n+1} is inconsistent. Then:

$$\Gamma^n \cup \{\varphi_n, @_i\Diamond i_m, @_{i_m}\psi\} \vdash \bot.$$

Using DDT we have $\Gamma^n \cup \{\varphi_n\} \vdash \neg(@_i\Diamond i_m \wedge @_{i_m}\psi)$, so for some finite conjunction γ of formulas from Γ, $\vdash \gamma \to \neg(@_i\Diamond i_m \wedge @_{i_m}\psi)$, hence, $\vdash (@_i\Diamond i_m \wedge @_{i_m}\psi) \to \neg\gamma$. But i_m is the first new nominal that does not occur in Γ^n or φ_n so we can apply the Paste rule obtaining $\vdash @_i\Diamond\psi \to \bot$ and contradicting the consistency of $\Gamma^n \cup \{\varphi_n\}$.

Suppose now that Γ^n is $K_{\overline{\tau}}$-consistent and consider φ_n of the form $@_i\exists x\psi$. We know that $\Gamma^n \cup \{@_i\exists x\psi\}$ is consistent; let us suppose that Γ^{n+1} is inconsistent. Then:

$$\Gamma^n \cup \{@_i\exists x\psi, @_i(\mathsf{EXISTS}(c_m)), @_i\psi(\frac{(@_i c_m)}{x})\} \vdash \bot.$$

The argument showing that this too leads to contradiction is the same as in the previous case, but using the Paste$_\forall$ rule rather than Paste.

Since Γ^n is $K_{\overline{\tau}}$-consistent for $n \in \mathbb{N}$, $\Gamma^* := \bigcup_{n\in\mathbb{N}} \Gamma^n$ is also $K_{\overline{\tau}}$-consistent.

We now prove that Γ^* is maximal. For suppose it is not. Then there exists a formula $\varphi \notin \Gamma^*$ such that $\Gamma^* \cup \{\varphi\}$ is $K_{\overline{\tau}}$-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 $\varphi \notin \Gamma^*$. $\square$

8. Completeness

In this section we will show how to build models out of the sets of sentences that our Lindenbaum lemma gives us: maximal, named, $\Diamond$-saturated, $\exists$-saturated, K_τ-consistent sets of formulas.

Definition 8.1. Let Γ be a maximal, named, $\Diamond$-saturated, $\exists$-saturated, K_τ-consistent set of formulas. We define binary relations $\sim_n$ and $\sim_r$, over Nom and $@\mathrm{Term}(\tau)$, respectively, by:

- $i \sim_n j \Leftrightarrow @_i j \in \Gamma$, where $i, j \in \mathrm{Nom}$
- $t \sim_r t' \Leftrightarrow t \approx t' \in \Gamma$, where $t, t' \in @\mathrm{Term}(\tau)$

Lemma 8.2. *Both $\sim_n$ and $\sim_r$ are equivalence relations. Moreover, if $t_m \sim_r t'_m$ for $m = 1, \ldots, n$, then $(@_i f)(t_1, \ldots, t_n) \sim_r (@_i f)(t'_1 \ldots, t'_n)$.*

Proof. The result for $\sim_n$ is standard. The axioms for $\approx$ give the result for $\sim_r$. $\square$

We build our models using equivalence classes $|i|_n$ of nominals and $|t|_r$ of terms; we suppress subscripts as it should be clear which equivalence class is meant:

Definition 8.3 *(Canonical K_τ-models).* Let Γ be a maximal, named, $\Diamond$-saturated, and $\exists$-saturated K_τ-consistent set of formulas. Then $\mathcal{M}^\Gamma = ((W^\Gamma, \mathrm{Dom}^\Gamma, D^\Gamma, R^\Gamma), I^\Gamma)$ is defined as follows:

- $W^\Gamma = \{|i| : i \text{ is a nominal}\}$.
- $\mathrm{Dom}^\Gamma = \{|t| : t \in @\mathrm{Term}(\tau) \text{ and } t \text{ is ground and } \mathsf{DEN}(t) \in \Gamma\}$.
- $D^\Gamma_{|i|} = \{|t| \in \mathrm{Dom}^\Gamma : @_i\mathsf{EXISTS}(t) \in \Gamma\}$.
- $|i|R^\Gamma|j|$ iff $@_i\Diamond j \in \Gamma$.

- $I^{\Gamma}(i) = |i|$, for each nominal i.
- For each $f \in \mathrm{Func}_n$ and $|t_1|, \ldots, |t_n| \in \mathrm{Dom}^{\Gamma}$,
$I^{\Gamma}_{|i|}(f)(|t_1|, \ldots, |t_n|) = \begin{cases} |(@_i f)(t_1, \ldots, t_n)| & \text{if } \mathsf{DEN}((@_i f)(t_1, \ldots, t_n)) \in \Gamma \\ \star & \text{otherwise.} \end{cases}$
- For each $P \in \mathrm{Rel}_n$,
$I^{\Gamma}_{|i|}(P) = \{(|t_1|, \ldots, |t_n|) \in \ (Dom^{\Gamma})^n : (@_i P)(t_1, \ldots, t_n) \in \Gamma)\}$.

We call $\mathcal{M}^{\Gamma}$ the *canonical model* over Γ. Note that any assignment g on a canonical model is a map from X to equivalence classes of ground terms t from $@\mathrm{Term}(\tau)$ such that $\mathsf{DEN}(t) \neq \star$.

Before going further, we check that the items in this definition make sense:

- R^{Γ} is well defined. 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 $(Bridge)$, $@_i \Diamond j'$.
- $I^{\Gamma}_{|i|}(f)$ is well defined. Suppose $t_m \sim_r t'_m$, $m = 1, \ldots, n$. Hence, $t_m \approx t'_m \in \Gamma$. By axiom $(Func)$, $(@_i f)(t_1, \ldots, t_n) \approx (@_i f)(t'_1, \ldots, t'_n) \in \Gamma$. That is, $I^{\Gamma}_{|i|}(f)(t_1, \ldots, t_n) = I^{\Gamma}_{|i|}(f)(t'_1, \ldots, t'_n)$.
- $I^{\Gamma}_{|i|}(P)$ is well defined. Suppose $t_m \sim_r t'_m$, $m = 1, \ldots, n$. Hence, $t_m \approx t'_m \in \Gamma$. By axiom $(Pred)$, $(@_i P)(t_1, \ldots, t_n) \in \Gamma$ iff $(@_i P)(t'_1, \ldots, t'_n) \in \Gamma$. That is, $I^{\Gamma}_{|i|}(P)(t_1, \ldots, t_n)$ iff $I^{\Gamma}_{|i|}(P)(t'_1, \ldots, t'_n)$.
- $D^{\Gamma}_i \subseteq \mathrm{Dom}^{\Gamma}$. By definition.

Our completeness proof will make use of variable substitutions of ground terms for free variables. We already have a notation for indicating substitutions for a *single* variable, namely the $\varphi(\frac{\theta}{x})$ notation we used in Axiom *Q2* and the proof of the Lindenbaum lemma. However in our completeness proof we will need to carry out simultaneous substitution on multiple variables, and this motivates the following definition:

Definition 8.4 *(Substitutions).* Let $\sigma : X \to @\mathrm{Term}(\tau)$. We define t^{σ} for any term t as follows:

- if $t \in X$ then $t^{\sigma} = \sigma(x)$
- if $t = f(t_1, \ldots, t_n)$ then $t^{\sigma} = f(t_1^{\sigma}, \ldots, t_n^{\sigma})$.

This extends to formulas in the following way:

- $i^{\sigma} := i$, $i \in \mathrm{Nom}$
- $\mathsf{DEN}(t)^{\sigma} := \mathsf{DEN}(t^{\sigma})$, $t \in \mathrm{Term}(\tau)$
- $(t_1 \approx t_2)^{\sigma} := (t_1^{\sigma} \approx t_2^{\sigma})$, $t_1, t_2 \in \mathrm{Term}(\tau)$
- $(P(t_1, \ldots, t_n))^{\sigma} := P(t_1^{\sigma}, \ldots, t_n^{\sigma})$, $P \in \mathrm{Rel}_n \cup @\mathrm{Rel}_n$ and $t_1, \ldots, t_n \in \mathrm{Term}(\tau)$;
- $(@_i \varphi)^{\sigma} := @_i(\varphi^{\sigma}) \in \mathrm{Fm}(\tau)$, $\varphi \in \mathrm{Fm}(\tau)$ and $i \in \mathrm{Nom}$
- $(\neg\varphi)^{\sigma} := \neg(\varphi^{\sigma})$ and $(\Box\varphi)^{\sigma} := \Box(\varphi^{\sigma})$, $\varphi \in \mathrm{Fm}(\tau)$;
- $(\varphi \wedge \psi)^{\sigma} := \varphi^{\sigma} \wedge \psi^{\sigma}$ and $(\varphi \vee \psi)^{\sigma} := \varphi^{\sigma} \vee \psi^{\sigma}$, for $\varphi \in \mathrm{Fm}(\tau)$ and $\psi \in \mathrm{Fm}(\tau)$
- $(\forall x\varphi)^{\sigma} := \forall x(\varphi^{\sigma^x_x})$, $x \in X$ and $\varphi \in \mathrm{Fm}(\tau)$ $(\sigma^x_x = \sigma \setminus \{(x, \sigma(x))\}) \cup \{(x, x)\})$

Two more pieces of notation will be useful. First we write *id* for the substitution mapping every variable to itself (that is, for the identity map on X). Second, for any function $f : A \to B$, and any $a \in A$ and $b \in B$, we define $f^a_b : A \to B$ by:

$$f^a_b(u) = \begin{cases} b, & \textit{if } u = a \\ f(u), & \textit{otherwise} \end{cases}$$

Note that for any rigid term t, we can write $\varphi(\frac{t}{x})$ as $\varphi^{id^x_t}$.

Definition 8.5 *(Equivalent assignments).* Let Γ be a maximal, named, $\Diamond$-saturated, $\exists$-saturated, K_{τ}-consistent set of formulas, and let g and h be two assignments on its canonical model. We say $g \sim_r h$ iff for all $x \in X$ we have that $g(x) \sim_r h(x)$.

The definition makes sense, as an assignment g on a canonical model maps all variables to ground terms in $@\mathrm{Term}(\tau)$. This brings us to a key lemma. Roughly speaking: τ-equivalent assignments respect $\approx$ and $@$. More precisely:

Lemma 8.6. *Let Γ be a maximal named, $\Diamond$-saturated, and $\exists$-saturated K_{τ}-consistent set of formulas. Let $g, h : X \to @\mathrm{Term}(\tau)$ be assignments such that $g \sim_r h$. Then:*

1. $t^g \approx t^h \in \Gamma$
2. $@_i\varphi^g \in \Gamma$ *iff* $@_i\varphi^h \in \Gamma$.

Proof. Claim (1) follows by induction on term structure. For $t \in X$ we have: $x^g = g(x) \sim_r h(x) = x^h$, thus $x^g \approx x^h \in \Gamma$. For complex terms, we use the induction hypothesis and congruence axioms. Consider $t = f(t_1, \dots, t_n)$ where $f \in \mathrm{Func}_n \cup @\mathrm{Func}_n$, $n \geq 0$. By IH we have that $t_m^g \approx t_m^h \in \Gamma$, for $m = 1, \dots, n$. By the *Func* axiom, $f(t_1^g, \dots, t_n^g) \approx f(t_1^h, \dots, t_n^h) \in \Gamma$. That is:

$$(f(t_1, \dots, t_n))^g \approx (f(t_1, \dots, t_n))^h \in \Gamma.$$

Claim (2) follows by induction on the complexity of φ.

- $\varphi = @_i j$. Trivial, as this expression contains no variables.
- $\varphi = @_i @_j \psi$. Then $\varphi = @_i(@_j\psi)^g = @_i@_j(\psi)^g = @_i@_j(\psi)^h$ (by IH) $= @_i(@_j\psi)^h$.
- $\varphi = @_i\mathsf{DEN}(t)$. First, by (1), $t^g \approx t^h \in \Gamma$, and $t^g \approx t^h \to (\mathsf{DEN}(t^g) \leftrightarrow \mathsf{DEN}(t^h)) \in \Gamma$ as well, since this is an instance of the (*DenDen*) axiom, and thus $\mathsf{DEN}(t^g) \leftrightarrow \mathsf{DEN}(t^h) \in \Gamma$ too. Hence $\varphi = @_i\mathsf{DEN}(t)^g \in \Gamma$ iff $@_i\mathsf{DEN}(t^g) \in \Gamma$ iff (by IH) $@_i\mathsf{DEN}(t^h) \in \Gamma$ iff $@_i\mathsf{DEN}(t)^h \in \Gamma$.
- $\varphi = @_i(t_1 \approx t_2)$. Suppose that $@_i(t_1^g \approx t_2^g) \in \Gamma$. As $t_1^g, t_2^g \in \mathrm{Term}(\tau)$ we can use the ($\mathrm{Agree}_\approx$) axiom, which tells us that $@_i(t_1^g \approx t_2^g) \in \Gamma$ iff $t_1^g \approx t_2^g \in \Gamma$. By IH this happens iff $t_1^h \approx t_2^h \in \Gamma$ iff (using ($\mathrm{Agree}_\approx$) again) $@_i(t_1^h \approx t_2^h) \in \Gamma$.
- $\varphi = @_i P(t_1, \dots, t_n)$, with $P \in \mathrm{Rel}_n \cup @\mathrm{Rel}_n$ and $t_1, \dots, t_n \in \mathrm{Term}(\tau)$. First, by (1), we have that $t_m^g \approx t_m^h \in \Gamma$, for $m = 1, \dots, n$, and since it is an instance of the (*Pred*) axiom, we have $P(t_1^g, \dots, t_n^g) \leftrightarrow P(t_1^h, \dots, t_n^h) \in \Gamma$ too. So $\varphi = @_i P(t_1, \dots, t_n)^g \in \Gamma$ iff $@_i P(t_1^g, \dots, t_n^g) \in \Gamma$ iff (by IH) $@_i P(t_1^h, \dots, t_n^h) \in \Gamma$ iff $@_i P(t_1, \dots, t_n)^h \in \Gamma$.
- $\varphi = @_i(\psi_1 \wedge \psi_2)$. We have $@_i(\psi_1 \wedge \psi_2)^g \in \Gamma$ iff $@_i\psi_1^g \wedge @_i\psi_2^g \in \Gamma$ iff $@_i\psi_1^g$ and $@_i\psi_2^g \in \Gamma$ iff $@_i\psi_1^h$ and $@_i\psi_2^h \in \Gamma$ (IH) iff $@_i\psi_1^h \wedge @_i\psi_2^h \in \Gamma$ iff $@_i(\psi_1 \wedge \psi_2)^h \in \Gamma$. The cases for the other booleans are similar.
- $\varphi = @_i\Diamond\psi$.

$$\begin{array}{lll}
@_i(\Diamond\psi)^g \in \Gamma & \text{iff} & @_i\Diamond(\psi)^g \in \Gamma \\
& \text{iff} & \text{exists } j \text{ s.t. } @_i\Diamond j \in \Gamma \text{ and } @_j\psi^g \in \Gamma \quad \text{, by } \Diamond\text{- saturation} \\
& \text{iff} & \text{exists } j \text{ s.t. } @_i\Diamond j \in \Gamma \text{ and } (@_j\psi)^g \in \Gamma \\
& \text{iff} & \text{exists } j \text{ s.t. } @_i\Diamond j \in \Gamma \text{ and } (@_j\psi)^h \in \Gamma \quad \text{, by IH} \\
& \text{iff} & \text{exists } j \text{ s.t. } @_i\Diamond j \in \Gamma \text{ and } @_j(\psi)^h \in \Gamma \\
& \text{iff} & @_i\Diamond(\psi)^h \in \Gamma \quad \text{, by } \Diamond\text{- saturation} \\
& \text{iff} & @_i(\Diamond\psi)^h \in \Gamma
\end{array}$$

- $\varphi = @_i\exists x\psi$.

$$\begin{array}{lll}
@_i(\exists x\psi)^g \in \Gamma & \text{iff} & @_i\exists x(\psi)^{g_x^x} \in \Gamma \\
& \text{iff} & \text{exists } c \text{ s.t. } @_i\mathsf{EXISTS}(c) \in \Gamma \text{ and } @_i\psi^{g_x^x}(\frac{@_i c}{x}) \in \Gamma \\
& \text{iff} & \text{exists } c \text{ s.t. } @_i\mathsf{EXISTS}(c) \in \Gamma \text{ and } @_i\psi^{g^x_{@_i c}} \in \Gamma \\
& \text{iff} & \text{exists } c \text{ s.t. } @_i\mathsf{EXISTS}(c) \in \Gamma \text{ and } @_i\psi^{h^x_{@_i c}} \in \Gamma \quad \text{by IH} \\
& \text{iff} & \text{exists } c \text{ s.t. } @_i\mathsf{EXISTS}(c) \in \Gamma \text{ and } @_i\psi^{h_x^x}(\frac{@_i c}{x}) \in \Gamma \\
& \text{iff} & @_i(\exists x\psi)^h = @_i\exists x(\psi)^{h_x^x} \in \Gamma
\end{array}$$

Let us look closer at the steps involved in this last case. First, note that while g is an assignment function, g_x^x is not. However g_x^x just plays a "dummy" role, allowing us to form $\psi^{g_x^x}(\frac{@_i c}{x})$, that is, $\psi^{g^x_{@_i c}}$, which *is* an assignment as it sends x to the ground term $@_i c$. Crucially, $g^x_{@_i c} \sim_r h^x_{@_i c}$, so we can apply the inductive hypothesis. With this case established, the lemma follows. □

For each assignment g into $\mathcal{M}^\Gamma$, we now define a substitution function $\sigma_g : X \to @\mathrm{Term}(\tau)$. We do so as follows. Suppose $g(x) = |t|$. As every element t of $|t|$ is ground, every element has the form $(@_k c)$ or $(@_k f)(t_1, \dots, t_n)$. Pick any of these terms t_l as the representative of $|t|$ (for example, pick the term t_l such that $\mathsf{DEN}(t_l)$ is first in the formula enumeration). Note that we can write any representative t_l in the form $\Downarrow_{i_k}(t_l)$ for some nominal k, as $\Downarrow$ is the identity map on $@\mathrm{Term}(\tau)$. Finally, define $\sigma_g(x) = \Downarrow_{i_k}(t_l) = t_l$. So $\sigma_g(x)$ is always the representative of $g(x)$.

Corollary 8.7. *Let* $g : X \to \mathrm{Dom}^\Gamma$ *and* $\theta \in @\mathrm{Term}(\tau)$ *such that* $|\theta| \in \mathrm{Dom}^\Gamma$. *Then*

$$@_i\varphi^{((\sigma_g)_x^x)}(\frac{\theta}{x}) \in \Gamma \textit{ iff } @_i\varphi^{\sigma_{g[x\mapsto|\theta|]}} \in \Gamma$$

Proof. Recall that $\varphi(\frac{\theta}{x}) = \varphi^{id_\theta^x}$. Since σ_g sends variables to ground terms, $(\varphi^{(\sigma_g)_x^x})^{id_\theta^x} = \varphi^{(\sigma_g)_\theta^x}$. So by the previous lemma, it suffices to prove that $(\sigma_g)_\theta^x \sim_r \sigma_{g[x\mapsto|\theta|]}$. On the one hand:

$$(\sigma_g)^x_\theta(u) = \begin{cases} \theta & \text{if } u = x \\ \Downarrow_{i_k}(t_l) & \text{otherwise,} \end{cases}$$

where $\Downarrow_{i_k}(t_l)$ is the representative of $g(u)$.

On the other hand, $\sigma_{g[x\mapsto|\theta|]}(u) = \Downarrow_{i_m}(t_s)$, the representative of $g[x \mapsto |\theta|](u)$. Moreover,

$$g[x \mapsto |\theta|](u) = \begin{cases} |\theta| & \text{if } u = x \\ g(u) & \text{otherwise.} \end{cases}$$

Thus $\sigma_{g[x\mapsto|\theta|]}(x) \sim_r \theta$ and for $u \neq x$, $|\sigma_{g[x\mapsto|\theta|]}(u)| = g(u)$. Hence $(\sigma_g)^x_\theta \sim_r \sigma_{g[x\mapsto|\theta|]}$. $\square$

Lemma 8.8. *For any $t \in \mathrm{Term}(\tau)$, any assignment g on $\mathcal{M}^\Gamma$ and any nominal i, we have*

$$[t]^{\mathcal{M}^\Gamma,|i|,g} = |\Downarrow_i(t^{\sigma_g})|$$

Proof. By induction on term structure.

($t \in X$)

$$\begin{aligned} [x]^{\mathcal{M}^\Gamma,|i|,g} &= g(x) \\ &= |t_k|, \text{ where } t_k \text{ is the representative of } g(x). \\ &= |\Downarrow_i(t_k)|, \text{ since } t_k \in @\mathrm{Term}(\tau), \text{so } \Downarrow_i(t_k) = t_k \\ &= |\Downarrow_i(x^{\sigma_g})| \end{aligned}$$

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

$$\begin{aligned} [f(t_1,\dots,t_n)]^{\mathcal{M}^\Gamma,|i|,g} &= I_{|i|}(f)([t_1]^{\mathcal{M},|i|,g},\dots,[t_n]^{\mathcal{M},|i|,g}) \\ &= I_{|i|}(f)(|\Downarrow_i(t_1^{\sigma_g})|,\dots,|\Downarrow_i(t_n^{\sigma_g})|) \quad \text{(by IH)} \\ &= |(@_i f)(\Downarrow_i(t_1^{\sigma_g}),\dots,\Downarrow_i(t_n^{\sigma_g}))| \\ &= |\Downarrow_i((f(t_1^{\sigma_g},\dots,t_n^{\sigma_g})))| \\ &= |\Downarrow_i(t^{\sigma_g})| \end{aligned}$$

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

$$\begin{aligned} [(@_j f)(t_1,\dots,t_n)]^{\mathcal{M}^\Gamma,|i|,g} &= I_{|j|}(f)([t_1]^{\mathcal{M},|i|,g},\dots,[t_n]^{\mathcal{M},|i|,g}) \\ &= I_{|j|}(f)(|\Downarrow_i(t_1^{\sigma_g})|,\dots,|\Downarrow_i(t_n^{\sigma_g})|) \quad \text{(by IH)} \\ &= |(@_j f)(\Downarrow_i(t_1^{\sigma_g}),\dots,\Downarrow_i(t_n^{\sigma_g}))| \\ &= |\Downarrow_i(((@_j f)(t_1^{\sigma_g},\dots,t_n^{\sigma_g})))| \\ &= |\Downarrow_i(t^{\sigma_g})| \end{aligned}$$

$\square$

Lemma 8.9 *(Truth Lemma). For every nominal i, any assignment g on $\mathcal{M}^\Gamma$ and every formula φ*

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

Proof. The proof proceeds by induction on the complexity of φ.

- $\varphi = j$
  $\mathcal{M}^\Gamma, |i|, g \models j$ iff $|i| = |j|$ iff $@_i j \in \Gamma$ iff $@_i j^{\sigma_g} \in \Gamma$.
- $\varphi = \mathsf{DEN}(t)$,
  $$\begin{aligned} \mathcal{M}^\Gamma, |i|, g \models \mathsf{DEN}(t) \quad &\text{iff} \quad [t]^{\mathcal{M},|i|,g} \neq \star \\ &\text{iff} \quad [\Downarrow_i(t)]^{\mathcal{M},|i|,g} \neq \star \\ &\text{iff} \quad |\Downarrow_i(t^{\sigma_g})| \in \mathrm{Dom} \\ &\text{iff} \quad \mathsf{DEN}(\Downarrow_i(t^{\sigma_g})) \in \Gamma \\ &\text{iff} \quad @_i\mathsf{DEN}(t^{\sigma_g}) \in \Gamma \\ &\qquad\quad \text{(by axiom } \mathsf{DEN}(\Downarrow_i(t)) \leftrightarrow @_i\mathsf{DEN}(t)) \\ &\text{iff} \quad @_i\mathsf{DEN}(t)^{\sigma_g} \in \Gamma \end{aligned}$$
- $\varphi = t_1 \approx t_2$,

$$\begin{array}{lll}\mathcal{M}^\Gamma, |i|, g \models t_1 \approx t_2 & \text{iff} & [t_1]^{\mathcal{M},|i|,g} = [t_2]^{\mathcal{M},|i|,g} \\ & \text{iff} & |\Downarrow_i(t_1^{\sigma_g})| = |\Downarrow_i(t_2^{\sigma_g})|, \text{ by Lemma 8.8} \\ & \text{iff} & \Downarrow_i(t_1^{\sigma_g}) \sim_r \Downarrow_i(t_2^{\sigma_g}) \\ & \text{iff} & \Downarrow_i(t_1^{\sigma_g}) \approx \Downarrow_i(t_2^{\sigma_g}) \in \Gamma \\ & \text{iff} & @_i(t_1^{\sigma_g} \approx t_2^{\sigma_g}) \in \Gamma, \text{ by axiom } K_{@\approx} \\ & \text{iff} & @_i(t_1 \approx t_2)^{\sigma_g} \in \Gamma\end{array}$$

- $\varphi = P(t_1, \ldots, t_n)$, with $P \in \mathrm{Rel}_n \cup @\mathrm{Rel}_n$ and $t_1, \ldots, t_n \in \mathrm{Term}(\tau)$;

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

$$\begin{array}{lll}\mathcal{M}^\Gamma, |i|, g \models P(t_1, \ldots, t_n) & \text{iff} & ([t_1]^{\mathcal{M},|i|,g}, \ldots, [t_n]^{\mathcal{M},|i|,g}) \in I_{|i|}(P) \\ & \text{iff} & (|\Downarrow_i(t_1^{\sigma_g})|, \ldots, |\Downarrow_i(t_n^{\sigma_g})|) \in I_{|i|}(P), \text{ by Lemma 8.8} \\ & \text{iff} & (@_i P)(\Downarrow_i(t_1^{\sigma_g}), \ldots, \Downarrow_i(t_n^{\sigma_g})) \in \Gamma \\ & \text{iff} & @_i(P(t_1^{\sigma_g}, \ldots, t_n^{\sigma_g})) \in \Gamma, \\ & & \text{by the Shuffle-1 Axiom} \\ & & (@_i P)(\Downarrow_i(t_1), \ldots, \Downarrow_i(t_n)) \leftrightarrow @_i(P(t_1, \ldots, t_n)) \\ & \text{iff} & @_i((P(t_1, \ldots, t_n)^{\sigma_g}) \in \Gamma\end{array}$$

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

$$\begin{array}{lll}\mathcal{M}^\Gamma, |i|, g \models (@_j S)(t_1, \ldots, t_n) & \text{iff} & ([t_1]^{\mathcal{M},|i|,g}, \ldots, [t_n]^{\mathcal{M},|i|,g}) \in I_{|j|}(S) \\ & \text{iff} & (|\Downarrow_i(t_1^{\sigma_g})|, \ldots, |\Downarrow_i(t_n^{\sigma_g})|) \in I_{|j|}(S), \text{ by Lemma 8.8} \\ & \text{iff} & (@_j S)(\Downarrow_i(t_1^{\sigma_g}), \ldots, \Downarrow_i(t_n^{\sigma_g})) \in \Gamma \\ & \text{iff} & @_i((@_j S)(t_1^{\sigma_g}, \ldots, t_n^{\sigma_g})) \in \Gamma, \\ & & \text{by the Shuffle-2 axiom} \\ & & (@_j S)(\Downarrow_i(t_1), \ldots, \Downarrow_i(t_n)) \leftrightarrow @_i((@_j S)(t_1, \ldots, t_n)) \\ & \text{iff} & @_i((@_j S)(t_1, \ldots, t_n)^{\sigma_g}) \in \Gamma\end{array}$$

- $\varphi = @_j\psi$.

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

- $\varphi = \neg\psi$.

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

- $\varphi = \Diamond\psi$.

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

- $\varphi = \psi_1 \wedge \psi_2$

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

- $\varphi = \exists x\psi$.

$$\begin{array}{lll}\mathcal{M}^\Gamma, |i|, g \models \exists x\psi & \text{iff} & \text{exists } a = |\theta| \in D^\Gamma_{|i|} \text{ s.t } \mathcal{M}, |i|, g[x \mapsto |\theta|] \models \psi \\ & \text{iff} & \text{exists } a = |\theta| \in D^\Gamma_{|i|} \text{ s.t } @_i\psi^{\sigma_{g[x\mapsto|\theta|]}} \in \Gamma, \text{ IH} \\ & \text{iff} & \text{exists } a = |\theta| \in D^\Gamma_{|i|} \text{ s.t } @_i\psi^{((\sigma_g)^x_x)}(\tfrac{\theta}{x}) \in \Gamma, \text{ by Corollary 8.7} \\ & \text{iff}^{(*)} & @_i\exists x(\psi)^{(\sigma_g)^x_x} \in \Gamma \\ & \text{iff} & @_i(\exists x\psi)^{\sigma_g} \in \Gamma\end{array}$$

Let's check the detail of the $(*)$ step. For the "$\Rightarrow$" implication, note that $|\theta| \in D^\Gamma_{|i|}$ means that $@_i\mathsf{EXISTS}(\theta) \in \Gamma$. Therefore, by the axiom (Q_2) and (Intro), $@_i\exists x(\psi)^{(\sigma_g)^x_x} \in \Gamma$.

The "$\Leftarrow$" implication holds by $\exists$-saturation. Because $@_i \exists x(\psi^{(\sigma_g)^x_x}) \in \Gamma$, there is a constant c such that $@_i \mathsf{EXISTS}(c) \in \Gamma$ and $@_i \psi^{((\sigma_g)^x_x)}(\frac{@_i c}{x}) = @_i \psi^{((\sigma_g)^x_x)^x_{@_i c}} \in \Gamma$. By axiom $(ExRig)$, $@_i \mathsf{EXISTS}(\Downarrow_i(c)) \in \Gamma$ and so $|\Downarrow_i(c)| := |@_i c| \in D_{|i|}$. Therefore, there is a $\theta := |@_i c| \in D^{\Gamma}_{|i|}$ such that $@_i \psi^{((\sigma_g)^x_x)}(\frac{@_i c}{x}) \in \Gamma$. □

Corollary 8.10. *Every K_τ-consistent set of sentences Γ is satisfiable on a named model.*

Proof. Given any K_τ-consistent set of sentences Γ, use the Lindenbaum lemma to expand it to maximal, named, $\Diamond$-saturated, and $\exists$-saturated K_τ-consistent set of sentences $\Gamma^\star$. There is at least one nominal in $\Gamma^\star$; call it k. Build the canonical model $\mathcal{M}^{\Gamma^\star}$. Then $\mathcal{M}^{\Gamma^\star}, |k| \models \Gamma$. □

Theorem 8.11 *(Completeness for K_τ). Let τ be a first-order hybrid similarity type, let φ be a sentence and Γ a set of sentences. Then:*

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

Proof. The standard consequence of the previous corollary. □

Moreover, every K_τ-consistent set of sentences Γ is also satisfiable on a model based on a skeleton in FM: simply take the canonical model $\mathcal{M}^{\Gamma^\star}$ of Corollary 8.10 and form a phantom world version of it (recall Lemma 4.4). This immediately yields:

Theorem 8.12 *(FM-completeness for K_τ). Let τ be a first-order hybrid similarity type, φ a sentence and Γ a set of sentences. Then:*

$$\Gamma \models_{\mathsf{FM}} \varphi \Rightarrow \Gamma \vdash \varphi.$$

Proof. An easy consequence of the preceding remarks. □

9. Pure extensions of K_τ

In this section we prove a semantic result about *pure sentences*. This will allow us to extend Theorem 8.11 in a simple and uniform way to the cover the logics of many modally interesting classes of skeletons. However, using pure formulas to extend Theorem 8.12 to uniformly cover classes of FM-skeletons is not so straightforward, and we will deal with that task in the following section.

First, what is a pure formula? In propositional hybrid logic it is formula that does not contain any ordinary propositional symbols, or put positively, a formula built using only nominals. In our first-order language, we define them as follows:

Definition 9.1 *(Pure formulas).* Fix a hybrid similarity type τ. A formula over this similarity type is pure iff it contains no relation or function symbols of any arity.

Note that the ban on relation symbols means that pure formulas cannot contain ordinary propositional symbols. Moreover, the ban on function symbols means that the only terms we have are variables. Put positively: pure formulas are built using nominals, and atomic formulas of the form $\mathsf{DEN}(x)$ and $x \approx y$. We can still use EXISTS (for variables) as $\mathsf{EXISTS}(y)$ is shorthand for $\exists x(x \approx y)$.

Pure formulas allow us to define some interesting classes of skeletons. Recall from Definition 3.6 that $S \models \varphi$ means that for every world w, every assignment g, and every interpretation I, we have $(S, I), w, g \models \varphi$. Then:

Definition 9.2 *(Defining skeletons).* A formula φ *defines* a class of skeletons S iff

$$S \models \varphi \textit{ iff } S \in \mathsf{S}.$$

A class of skeletons is *definable* iff it can be defined by some formula. Skeleton classes are often specified by some property shared by all skeletons in the class (for example, having a transitive accessibility relation). Thus we often talk of properties (like transitivity) being definable too.

Here we are interested in properties definable by pure sentences. We list some examples of properties and pure sentences that define them:

Reflexivity	$@_i\Diamond i$
Symmetry	$@_i\Box\Diamond i$
Transitivity	$\Diamond\Diamond i \to \Diamond i$
Irreflexivity	$@_i\neg\Diamond i$
Antisymmetry	$@_i\Box(\Diamond i \to i)$
Asymmetry	$@_i\Diamond j \to \neg @_j\Diamond i$
Expanding domains	$\forall y(\mathsf{EXISTS}(y) \to \Box\mathsf{EXISTS}(y))$
Contracting domains	$\forall y(\Diamond\mathsf{EXISTS}(y) \to \mathsf{EXISTS}(y))$
Constant domains	$\forall y(@_i\mathsf{EXISTS}(y) \to @_j\mathsf{EXISTS}(y))$
Disjoint domains	$\forall y(@_i\mathsf{EXISTS}(y) \wedge @_j\mathsf{EXISTS}(y) \to @_i j)$
Convex domains	$\forall y(\mathsf{EXISTS}(y) \to \Box(\Diamond\mathsf{EXISTS}(y) \to \mathsf{EXISTS}(y)))$

The first six items are standard examples from propositional hybrid logic and define properties of the accessibility relation R; the last five define conditions on local domains.[4] It is easy to check that these all define the stated conditions.

Now for the key observation: when pure sentences are added as additional axioms to K_τ, the resulting system (which we call a pure extension) is complete with respect to the class of skeletons that the axioms define (that is, the class of skeletons on which every axiom is valid). We first make this claim precise, and then prove it.

Definition 9.3 *(Pure extensions of K_τ).* Let PE be a set of pure sentences that is closed under uniformly replacing nominals by nominals; that is, if $\rho \in \mathsf{PE}$, and ρ' is obtainable from ρ by uniformly replacing nominals by nominals, then $\rho' \in \mathsf{PE}$ too. Then $K_\tau + \mathsf{PE}$, the pure extension of K_τ by PE, has all the rules and axioms of K_τ, and all the sentences in PE are axioms as well.

For example, we might add all instances of $@_i\neg\Diamond i$ as additional axioms, or all instances of $\forall y(@_i\mathsf{EXISTS}(y) \to @_j\mathsf{EXISTS}(y))$, or all instances of both these sentences (by "instances" we mean any sentence obtained by uniformly replacing nominals by nominals). And now for the semantic lemma that leads to the desired completeness results.[5] This tells us that if all instances of a pure sentence are valid on a named model, then they are valid on the skeleton underlying the model as well.

Lemma 9.4 *(Purity and skeleton validity). Let* PE *be a set of pure sentences that is closed under uniformly replacing nominals by nominals. Let $\mathcal{M} = (S, I)$ be a named model such that for all $\rho \in \mathsf{PE}$ we have that $\mathcal{M} \models \rho$. Then $S \models \rho$.*

Proof. Let $\mathcal{M} = (S, I)$ be a named model such that for all $\rho \in \mathsf{PE}$ we have that $\mathcal{M} \models \rho$. Assume for the sake of contradiction that for some $\rho \in \mathsf{PE}$ we have $S \not\models \rho$. That is, for some world w and some interpretation I' we have $(S, I'), w \not\models \rho$; as ρ is a sentence the choice of assignment is irrelevant. Let $i_1, \ldots, i_n$ be the nominals in ρ, and suppose $I'(i_1) = \{w_1\}, \ldots, I'(i_n) = \{w_n\}$. Because $\mathcal{M}$ is named, there are nominals $j_1, \ldots, j_n$ such that $I(j_1) = \{w_1\}, \ldots, I(j_n) = \{w_n\}$. Let ρ' be the result of uniformly substituting $j_1 \ldots, j_n$ for $i_1, \ldots, i_n$ in ρ. It follows that $\mathcal{M}, w \not\models \rho'$, which is impossible as $\rho' \in \mathsf{PE}$. We conclude that $S \models \rho$ after all. □

Corollary 9.5. *Every $K_\tau + \mathsf{PE}$-consistent set of sentences is satisfiable on a named model based on a skeleton that belongs to the class that* PE *defines.*

Proof. The proof of Corollary 8.10 showing that every K_τ-consistent set of sentences has a model goes through unchanged for $K_\tau + \mathsf{PE}$-consistent sets of sentences: we simply work with $K_\tau + \mathsf{PE}$-MCSs instead of K_τ-MCSs. Crucially, the canonical model this gives us is *named* and it makes all the axioms true at every world, including the axioms in PE (which are closed under uniformly replacing nominals by nominals). Thus the criteria of the previous lemma are satisfied, and so the underlying skeleton belongs to the class defined by PE. □

Theorem 9.6 *(Completeness for $K_\tau + \mathsf{PE}$). Let τ be a first-order hybrid similarity type, φ be a sentence and Γ a set of sentences. Then:*

$$\Gamma \models_{\mathsf{PE}} \varphi \Rightarrow \Gamma \vdash_{\mathsf{PE}} \varphi.$$

Proof. The standard consequence of the previous corollary. □

[4] In their discussion of constant domain models, Fitting and Mendelsohn [1, Chapter 4.6] call expanding domains *monotonic*, and contracting domains *anti-monotonic*. A disjoint domain model is one in which any pair of distinct worlds has distinct local domains. A convex domain model is one in which: if an individual d exists at a world w, then at all world w' such that wRw', if d also exists at some world w'' such that $w'Rw''$, then d exists at w' too.

[5] Various forms of this result for propositional hybrid logic have been known since the early 1970s: see for example [12–14].

To return to our previous example, this tells us that if we add all instances $@_i\neg\Diamond i$ as additional axioms, then the resulting logic is complete with respect to irreflexive skeletons, and if we add all instances of $\forall y(@_i\mathsf{EXISTS}(y) \to @_j\mathsf{EXISTS}(y))$, then we have a logic complete with respect to skeletons with constant domains. The results are additive: if we add all instances of both sentences we have a completeness result for irreflexive skeletons with constant domains. In short, Theorem 9.6 gives us automatic completeness results for many modally interesting properties.

10. Pure extensions of K_τ + FM

So far so good — but Theorem 9.6 has a weakness. Recall that an FM-skeleton is a skeleton $S = (W, \mathrm{Dom}, D, R)$ such that $\mathrm{Dom} = \bigcup_{w\in W} D_w$, or equivalently, for every $d \in \mathrm{Dom}$ there is some $w \in W$ such that $d \in D_w$. That is, an FM-skeleton is one with no phantom zone. Unfortunately, in general we have no idea whether the models built using the methods of the previous section have phantom zones or not. Thus it would be useful to have an analog of Theorem 9.6 that provided automatic completeness proofs for pure formulas *with respect to FM-skeletons*. However, this is trickier. For a start, there is no formula (pure or otherwise) that defines the class of FM-skeletons: this is an immediate consequence of Lemma 4.4. Moreover the method used to prove this lemma was simply to glue on a new "phantom world" that gathered all the phantom zone entities together. This simple model transformation works for plain validity, and for some simple properties (for example: irreflexivity) but it won't provide the sort of general analog of Theorem 9.6 that we would like.

In this section we will show how to strengthen the base logic so that we can prove an analogous completeness result for pure extensions with respect to FM-skeletons. We shall strengthen K_τ by adding the FM rule:[6]

$$\frac{\vdash @_i\mathsf{EXISTS}(t) \to \varphi}{\vdash \varphi}$$

where $t \in @\mathrm{Term}(\tau)$, t is ground, i does not occur in either t or φ, and no symbol in t occurs in φ. Call the resulting system K_τ + FM.

Lemma 10.1. *If S is an FM-skeleton, then S admits the above rule. That is: if the premiss of the rule is valid on S, then so is the conclusion.*

Proof. Assume that S is an FM-skeleton, and further assume that S validates the premiss of this rule. That is, assume that:

$$S \vDash @_i\mathsf{EXISTS}(t) \to \varphi.$$

We need to show that $S \vDash \varphi$. So, let I be an interpretation on S, let $w \in W$, and let g be an assignment. As S validates the premiss, we have:

$$(S, I), w, g \vDash @_i\mathsf{EXISTS}(t) \to \varphi.$$

We now show that $(S, I), w, g \vDash \varphi$. There are three cases to consider.

1. Suppose $(S, I), w, g \vDash @_i\mathsf{EXISTS}(t)$. Then $(S, I), w, g \vDash \varphi$.
2. Suppose $(S, I), w, g \nvDash @_i\mathsf{EXISTS}(t)$, and t is defined. This means that $[t]^{(S,I),I(i),g}$ (the interpretation of t at the world named i) does not belong to the domain of the i-world. However it does belong to the domain of some world v, as S is an FM-skeleton. Let I' be the interpretation that is exactly like I except that $I'(i) = v$. This means that I and I' agree on all symbols in t and φ, as i belongs to neither. So by Lemma 5.1(1) we have $[t]^{(S,I),I(i),g} = [t]^{(S,I'),I(i),g}$ hence $(S, I'), w, g \vDash @_i\mathsf{EXISTS}(t)$, as i now names v. Thus, as $S \vDash @_i\mathsf{EXISTS}(t) \to \varphi$ is valid on S, we have that $(S, I'), w, g \vDash \varphi$. But now, using Lemma 5.1(3), we again conclude that $(S, I), w, g \vDash \varphi$.
3. $(S, I), w, g \nvDash @_i\mathsf{EXISTS}(t)$, and t is undefined. That is, $[t]^{(S,I),I(i),g} = \star$. But there is an interpretation I' that differs from I only on symbols in t such that $[t]^{(S,I'),I(i),g}$ is defined: by Lemma 5.2 it suffices to choose an I' that assigns a total function from Dom^n to Dom for every function symbol in t. Let d be $[t]^{(S,I'),I(i),g}$. Then:
   - If d is in the domain of the world named i, $(S, I'), w, g \vDash @_i\mathsf{EXISTS}(t)$, hence reasoning as in Case 1 we have $(S, I'), w, g \vDash \varphi$. Lemma 5.1(3) then yields $(S, I), w, g \vDash \varphi$.
   - If d is not in the domain of the world named i, $(S, I'), w, g \nvDash @_i\mathsf{EXISTS}(t)$, but t is now defined, hence reasoning as in Case 2 we have $(S, I'), w, g \vDash \varphi$. Lemma 5.1(3) then yields $(S, I), w, g \vDash \varphi$.

So in all three cases we have that $(S, I), w, g \vDash \varphi$. As our choices of I, w and g were arbitrary, this means we have $S \vDash \varphi$, as required. □

Theorem 10.2. *K_τ + FM is sound with respect to the class of FM-skeletons.*

[6] A variant of this rule was suggested in the concluding remarks of [3] for handling FM-skeletons in first-order hybrid logic. It does not seem to have been explored since.

Proof. The axioms and rules of K_τ are sound with respect to all skeletons. Lemma 10.1 shows that the FM rule is sound with respect to FM-skeletons. $\square$

Now, we immediately have an easy completeness result for $K_\tau + \mathsf{FM}$: Theorem 8.12 tells us that K_τ is complete with respect to the class of FM-skeletons, thus (trivially) $K_\tau + \mathsf{FM}$ is complete with respect to this class too; adding a sound rule does not diminish its deductive power. However, the completeness result for K_τ was proved using K_τ-MCSs that were named, $\Diamond$-saturated, and $\exists$-saturated. We will now prove a more useful completeness result for $K_\tau + \mathsf{FM}$ using $K_\tau + \mathsf{FM}$-MCSs which have all these properties, and which, in addition are FM-saturated. As we shall see, this will let us put pure sentences to work and generalize Theorem 8.12 in a way that applies to FM-skeletons. This is the concept we need:

Definition 10.3 *(FM-saturation).* An MCS Γ is FM-saturated iff for all ground terms $t \in @\mathrm{Term}(\tau)$ in Γ, there is a nominal i such that $@_i\mathsf{EXISTS}(t) \in \Gamma$.

Now, potentially there are many ways that we could use $K_\tau + \mathsf{FM}$-MCSs to build a model, but the next lemma tells us that as long as we do so in a way that satisfies the first three conditions used to define canonical K_τ models (recall Definition 8.3), then the resulting model will be based on an FM-skeleton:

Lemma 10.4. *Let Γ be a $K_\tau + \mathsf{FM}$-MCS that is FM-saturated. Suppose we use Γ to build a canonical model $\mathcal{M}^\Gamma$ in a way that satisfies the following three conditions:*

1. $W^\Gamma = \{|i| : i$ *is a nominal*$\}$.
2. $\mathrm{Dom}^\Gamma = \{|t| : t \in @\mathrm{Term}(\tau)$ *and t is ground and* $\mathsf{DEN}(t) \in \Gamma\}$.
3. $D^\Gamma_{|i|} = \{|t| \in \mathrm{Dom}^\Gamma : @_i\mathsf{EXISTS}(t) \in \Gamma\}$.

Then for every $d \in \mathrm{Dom}^\Gamma$ there is some $w \in W$ such that $d \in D^\Gamma_w$. That is, the model we construct will be based on an FM-skeleton.

Proof. For suppose not. Then there is some $d \in \mathrm{Dom}^\Gamma$ such that d does not belong to the local domain of any world w. But by condition 2, any such d is of the form $|t|$ for some ground rigid term t, and furthermore $\mathsf{DEN}(t) \in \Gamma$. Hence, as Γ is FM-saturated we have, for some nominal i, that $@_i\mathsf{EXISTS}(t) \in \Gamma$. Hence, by conditions 1 and 3, $d = |t|$ belongs to the local domain $D^\Gamma_{|i|}$ of world $|i|$. $\square$

This explains our interest in FM-saturation, and a new completeness result is now in sight. But we still need to show that there are MCSs with all four properties we need. The Lindenbaum lemma tells us how to form an MCS that is named, $\Diamond$-saturated, and $\exists$-saturated, but we need to do some extra work to obtain an MCS that has these three properties and is FM-saturated as well.

Lemma 10.5. *Let Γ be K_τ + FM-consistent set of formulas, let t be a ground rigid term occurring in Γ, and let i be a nominal not occurring in Γ (a "new nominal"). Then $\Gamma \cup \{@_i\mathsf{EXISTS}(t)\}$ is consistent.*

Proof. Suppose $\Gamma \cup \{@_i\mathsf{EXISTS}(t)\}$ is inconsistent. Thus $\Gamma \cup \{@_i\mathsf{EXISTS}(t)\} \vdash \bot$. Hence $\Gamma \vdash \neg @_i\mathsf{EXISTS}(t)$. So for some finite conjunction θ of formulas from Γ, we have $\vdash \theta \to \neg @_i\mathsf{EXISTS}(t)$, and hence, $\vdash @_i\mathsf{EXISTS}(t) \to \neg\theta$. But as i is new it does not occur in Γ, hence it does not occur in θ, so we can apply the FM rule and deduce that $\vdash \neg\theta$, contradicting the consistency of Γ. Thus $\Gamma \cup \{@_i\mathsf{EXISTS}(t)\}$ was consistent after all. $\square$

Lemma 10.6 *(FM-presaturation). Let Γ be a finite or countably infinite $K_\tau + \mathsf{FM}$-consistent set of formulas. Then Γ can be extended to a finite or countably infinite $K_\tau + \mathsf{FM}$-consistent set $(\Gamma)^{\mathsf{FMP}}$ (an FM-presaturated set) in which for every rigid constant t that occurs in Γ, there is a formula of the form $@_i\mathsf{EXISTS}(t)$.*

Proof. Let Γ be such a set. If no rigid constants t occur in Γ there is nothing to do. Otherwise, enumerate the distinct ground rigid constants t occurring in Γ as t_n, where $n > 0$. This enumeration will either be countably infinite or of length $m > 0$ for some natural number m. Then choose enough new nominals i_n (either m or countably many) to match this enumeration and use them as follows: Define Γ^0 to be Γ. Suppose Γ^k has been defined. Then define Γ^{k+1} to be:

$$\Gamma^{k+1} = \Gamma^k \cup \{@_{i_{k+1}}\mathsf{EXISTS}(t_{k+1})\}.$$

By assumption, $\Gamma^0 = \Gamma$ is consistent, and the previous lemma shows that each expansion step preserves consistency. Define $(\Gamma)^{\mathsf{FMP}}$ to be Γ^m if Γ was finite, and $\bigcup_{k\in\omega}\Gamma^k$ otherwise. Either way, $(\Gamma)^{\mathsf{FMP}}$ is consistent, and we have a finite or countably infinite FM-presaturated set. $\square$

Lemma 10.7. *Let Γ be a finite or countably infinite $K_\tau + \mathsf{FM}$-consistent set of formulas. Then we can extend Γ to a countably infinite $K_\tau(\mathsf{FM})$-consistent set of formulas that is maximal, named, $\Diamond$-saturated, $\exists$-saturated, and also* FM*-saturated.*

Proof. First, define Γ_0 to be Γ, and Γ_1 to be $(\Gamma_1)^{\mathsf{FMP}}$. Then, for all odd numbers n, define Γ_{n+1} to be the Lindenbaum expansion of Γ_n, and for all even numbers n define Γ_{n+1} to be $(\Gamma_n)^{\mathsf{FMP}}$. That is, alternate the Lindenbaum expansion and the presaturation process countably many times. Finally, define Γ_ω to be the union of this chain of sets. Γ_ω is the countably infinite MCS we need. □

And we now have the desired completeness result. First, we define the canonical model for $K_\tau + \mathsf{FM} + \mathsf{PE}$ as we did for K_τ in Definition 8.3, but we now use the MCS guaranteed to exist by the previous lemma. As our definition of canonical model is unchanged, Lemma 10.4 applies, thus the model is based on an FM-skeleton. As before we have a named model, so we can prove an analog of Corollary 9.5:

Corollary 10.8. *Every $K_\tau + \mathsf{FM} + \mathsf{PE}$-consistent set of sentences is satisfiable on a named model based on an FM-skeleton that belongs to the class that* PE *defines.*

Proof. As for Corollary 9.5. □

This immediately yields the desired analog of Theorem 9.6:

Theorem 10.9 *(FM-Completeness for $K_\tau + \mathsf{FM} + \mathsf{PE}$). Let τ be a first-order hybrid similarity type, φ be a sentence and Γ a set of sentences. Then*

$$\Gamma \models_{\mathsf{FM+PE}} \varphi \Rightarrow \Gamma \vdash_{\mathsf{FM+PE}} \varphi.$$

Proof. Immediate by the previous corollary. □

11. Concluding remarks

In this paper we proved a number of completeness results for a first-order language with partial function symbols built over the basic hybrid language. There were two main syntactic enrichments: allowing all predicate and functions symbols to be rigidified, and including a DEN predicate to detect undefined terms. We think that the language of this paper, and the results we have proved here, are useful tools for exploring first-order modal logic further — but to close this paper we want to focus more on where this paper came from rather than where we plan to take it next.

As we mentioned earlier, it was our experience with *higher*-order hybrid logic that led us to a "rigidify everything!" policy. Moreover, our experience with partial type theory suggested the usefulness of DEN-like operators. However it also made us aware that there were some interesting issues involving phantom zones, partial functions, and the definition of DEN, and that it could be useful to explore them in the simpler setting of *first*-order hybrid logic.

When we hybridized partial type theory, we did not work over the basic hybrid language, and we did not use the primitive DEN operator used here; rather, we worked with a stronger hybrid logic in which a variant of the DEN operator was definable. This logic contained two standard hybrid logical tools: the universal modality, and the $\downarrow$-binder. For a detailed discussion of these extensions, the interested reader should consult [4]; here we will say just enough to make our main point clear.

The universal modality is an additional modality that explores the universal relation $W \times W$ on worlds. Its box-form $A\varphi$ is true at a world w iff φ is true at *all* worlds, and its diamond-form $E\varphi$ is true at w iff there *exists* some world that makes φ true. Adding the universal modality increases the expressive power at our disposal. For a start, we can now define the satisfaction operators: either $A(i \to \varphi)$ or $E(i \wedge \varphi)$ captures the effect of $@_i\varphi$. Indeed, we can now define the class of FM-skeletons using a pure sentence:

$$\mathsf{DEN}(@_i c) \to E(\mathsf{EXISTS}(@_i c)).$$

This says: if the constant $(@_i c)$ is defined, then *there is some world* at which the domain element d that $(@_i c)$ denotes exists. That is: because the universal modality sees all worlds, d cannot hide inside the phantom zone.

We also added the hybrid $\downarrow$-binder and defined:

$$\mathsf{DEN}_w(c) := \downarrow i.E(\mathsf{EXISTS}((@_i c))).$$

That is: the $\downarrow$-binder binds the nominal i to the evaluation world (it "remembers" where the evaluation started). Then E looks for a world with a local domain in which the entity that c picks out *at the world of evaluation* exists.[7] To spell this out:

[7] So DEN_w, unlike DEN, does not directly test for $\star$ at the world of evaluation.

c denotes$_w$ (at some world of evaluation, which we label i) iff there is some world w where the value that c received (at the i-world) exists in the local domain of w. As this makes clear, $\downarrow$ is powerful. Indeed $\downarrow$ is why hybrid logical approaches to first-order modal logic don't need the predicate abstraction used in Fitting and Mendelsohn's textbook [1]: the combination of $\downarrow$ and @ offers an alternative way of capturing modal scope distinctions. Consider an utterance of "It is necessary that the President of France likes wine". The sentence $\Box{\downarrow}i.LW((@_i pf))$ captures one reading: *in all possible worlds, whoever happens to be the President of France there, likes wine*. On the other hand, ${\downarrow}i.\Box LW((@_i pf))$ captures: *in all possible worlds, the individual who is the President of France in the utterance world, likes wine*.

In forthcoming work we plan to further explore the similarities and differences between FM-style and hybrid approaches to first-order modal logic — but, for now, back to the phantom zone. The completeness theory of A and $\downarrow$ are well understood, and the defined DEN_w operator worked well in the higher-order setting. But we wanted something simpler, which led to the primitive "test for $\star$" version of DEN and the questions underlying this paper: how to make such a DEN work in a first-order logic with partial function symbols built over the *basic* hybrid language (that is: without A and $\downarrow$), and, in particular, how to make it work with pure extensions while avoiding problems with phantom zones. And as we have seen, partial functions and pure extensions work together just fine, whether you choose to exorcise the phantom zone or not.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

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 Program, and by FCT, within the projects PTDC/CCI-COM/30947/2017, UIDB/04106/2020 and UIDP/04106/2020.

References

[1] M. Fitting, R. Mendelsohn, First-Order Modal Logic, Springer, 1998.
[2] B. Cate, Model theory for extended modal languages, Ph.D. thesis, University of Amsterdam, 2005, iLLC Dissertation Series DS-2005-01.
[3] P. Blackburn, B. ten Cate, Pure extensions, proof rules, and hybrid axiomatics, Stud. Log. 84 (2006) 277–322.
[4] C. Areces, B. ten Cate, Hybrid logic, in: Handbook of Modal Logic, Elsevier, 2007, pp. 821–868.
[5] C. Areces, P. Blackburn, A. Huertas, M. Manzano, Completeness in hybrid type theory, J. Philos. Log. 43 (2014) 209–238.
[6] P. Blackburn, A. Huertas, M. Manzano, K.F. Jørgensen, Henkin and hybrid logic, in: The Life and Work of Leon Henkin, Springer, 2014, pp. 279–306.
[7] M. Manzano, M.A. Martins, A. Huertas, A semantics for equational hybrid propositional type theory, Bull. Sect. Log. 43 (3–4) (2014) 121–138.
[8] M. Manzano, M. Martins, A. Huertas, Completeness in equational hybrid propositional type theory, Stud. Log. 107 (6) (2019) 1159–1198, https://doi.org/10.1007/s11225-018-9833-5.
[9] M. Manzano, A. Huertas, P. Blackburn, M. Martins, V. Aranda, Hybrid partial type theory, 2021, submitted for publication.
[10] P. Blackburn, M. Marx, Tableaux for quantified hybrid logic, in: U. Egly, C. Fernmüller (Eds.), Automated Reasoning with Analytic Tableaux and Related Methods, International Conference, TABLEAUX 2002, Copenhagen, Denmark, 2002, pp. 38–52.
[11] P. Blackburn, M. Martins, M. Manzano, A. Huertas, Rigid first-order hybrid logic, in: International Workshop on Logic, Language, Information, and Computation, Springer, 2019, pp. 53–69.
[12] R. Bull, An approach to tense logic, Theoria 36 (3) (1970) 282–300.
[13] G. Gargov, V. Goranko, Modal logic with names, J. Philos. Log. 22 (6) (1993) 607–636.
[14] P. Blackburn, M. Tzakova, Hybrid languages and temporal logic, Log. J. IGPL 7 (1) (1999) 27–54.