Completeness in Hybrid Type Theory[*]

Carlos Areces
Universidad Nacional de Córdoba

Patrick Blackburn
Roskilde Universitet

Antonia Huertas
Universitat Oberta de Catalunya

María Manzano
Universidad de Salamanca

November 27, 2012

Abstract

We show that basic hybridization (adding nominals and @ operators) makes it possible to give straightforward Henkin-style completeness proofs even when the modal logic being hybridized is higher-order. The key ideas are to add nominals as expressions of type t, and to extend to arbitrary types the way we interpret $@_i$ in propositional and first-order hybrid logic. This means: interpret $@_i\alpha_a$, where α_a is an expression of any type a, as an expression of type a that rigidly returns the value that α_a receives at the i-world. The axiomatization and completeness proofs are generalizations of those found in propositional and first-order hybrid logic, and (as is usual in hybrid logic) we automatically obtain a wide range of completeness results for stronger logics and languages. Our approach is deliberately low-tech. We don't, for example, make use of Montague's intensional type s, or Fitting-style intensional models; we build, as simply as we can, hybrid logic over Henkin's logic.

1 Introduction

Hybrid logic is an extension of modal logic in which it is possible to name worlds using special atomic formulas called nominals. Nominals are true at a unique world in any model, thus a nominal i names the world it is true at. Once nominals have been introduced it becomes natural to make a further extension: to add modalities of the form $@_i$, where i is a nominal, and to interpret formulas of the form $@_i\varphi$ as asserting that φ is true at the unique world named by i (for surveys of hybrid logic, see Blackburn [5] and Areces and Ten Cate [3]).

[*] Research partially funded by the Spanish Ministry of Science and Innovation as part of the project *Nociones de Completud* (*Conceptions of Completeness*), Proyectos de Investigación Fundamental, MICINN FFI2009-09345, Universidad de Salamanca, 2009–2011.

This basic hybridization process (that is, adding nominals and @-operators to some modal language of interest) typically has interesting consequences, and here we are concerned with the consequences for completeness. We shall show that hybridization permits relatively simple Henkin-style model building techniques to be used, even when the modal logic being hybridized is higher-order.

Some background remarks. Completeness theory for ordinary (unhybridized) propositional modal logic revolves around the use of canonical models. A canonical model is a large (typically uncountable) model consisting of *all* maximal consistent sets (MCSs) of the logic in question, together with an appropriate accessibility relation (see Hughes and Cresswell [25] for an introduction and the *Handbook of Modal Logic* [10] for advanced material). Contrast this with what has become (since Leon Henkin's [22] groundbreaking work) the standard approach in first-order logic: the method of constants. In such proofs, the model for a consistent set of sentences is built out of equivalence classes of constants taken from a *single* MCS and (for countable languages) the model is countable.

It has long been known that hybridization makes it possible to carry out Henkin-style model constructions in propositional modal logic (see for example Bull [12], Gargov and Goranko [18], Blackburn and Tzakova [9], Blackburn and Ten Cate [8]). The key observation is that models can be built out of equivalence classes of nominals (much as first-order models are built out of equivalence classes of constants in Henkin's construction) and that the $@_i$ operators (or stronger operators, such as the universal modality [20]) make it possible to specify — within a single MCS — which formulas need to be true at which worlds. More recently, it has become clear that hybridization also makes Henkin-style completeness proofs possible when first-order modal logics are hybridized (Braüner and Ghilardi [11] gives a good overview). In this setting a new idea (introduced in Blackburn and Marx [7]) comes into play: overloading the $@_i$ operator so that it can take as arguments not merely formulas but constants too. In this approach, $@_i c$ denotes the individual that the constant c denotes at the world named by i. To put it another way: $@_i c$ is a new constant that rigidly designates what c denotes at the i-world.

The goal of this paper is to investigate whether basic hybridization also leads to simple Henkin-style completeness proofs in the setting of (classical) higher-order modal logic (that is, modal logics built over Church's simple theory of types [14]), and as we shall show, the answer is "yes". The crucial idea is to use $@_i$ as a rigidifier for arbitrary types. We shall interpret $@_i \alpha_a$, where α_a is an expression of any type a, to be an expression of type a that rigidly returns the value that α_a receives at the i-world. As we shall show, this enables us to construct a description of the required model inside a single MCS and hence to prove (generalized) completeness for higher-order hybrid logic.

Higher-order modal logic is not a large field, but it is a significant one, and over the years an impressive body of work has explored it in interestingly different directions (for a useful survey, see Muskens [30]). Currently, higher-order modal logic probably plays its most significant role in natural language semantics. The pioneer here was Richard Montague, who developed various higher-order modal logics, system PTQ being the best known [28]. PTQ made use of three types: t for truth values, e for entities, and s for world/time pairs. Syntactical novelties included two operators, $^\wedge$ (the intensionalizing operator) and $^\vee$ (the extensionalising operator) both of which made use of type s (the intensional type, as it is often called). It is a complex system (Barwise and Cooper [4] once likened it to a Rube-Goldberg machine) but its impact was immense, and rightly so: PTQ opened the door to modern natural language semantics. But PTQ was far from the last word. In his PhD thesis, Montague's student Daniel Gallin [17] proposed an alternative, TY_2; this is a two-sorted version of Church's [14] simple theory of types, the second sort being Montague's intensional type s.[1] The TY_2 system (and more generally, the TY_n systems it spawned) *don't* use modal operators; instead they allow direct quantification over worlds (as a modal logician would say: they incorporate the full first-order modal correspondence language). Systems of this kind have since played a significant role in natural language semantics (see, for example, Groenendijk and Stokhof [21] and Muskens [29]).

In philosophy, perhaps the best known recent work is due to Melvin Fitting [15], who developed a novel approach to higher-order modal logic and used it to investigate Gödels ontological argument for the existence of God. Fitting's approach has been influential. Syntactically, it uses modal operators, but dispenses with the function-argument syntax usual in type theory in favor of a predicate-term syntax reminiscent of first-order logic. But it is his semantic innovation which is likely to be enduring: the use of intensional models, a mechanism which makes it possible to avoid restrictions to rigid terms.

The most recent work comes from computer science, where Gert Smolka and his students [26] have recently turned matters on their heads: starting with classical type theories, they view (propositional) modal and hybrid logics as subsystems defined *within* classical type theory, and use this perspective to guide their search for efficient proof procedures. Previous authors have noted that various kinds of modality can be defined in various type theories, but the systematic use of higher-order logic as a tool for defining and exploring propositional modal logic is novel.

In this paper we will not discuss alternative approaches to higher-order modal reasoning. Indeed, our goal is to add to this variety by demonstrating the effectiveness of even basic hybridization in higher-order settings. To this end, we have restricted the hybrid apparatus to the use of nominals and the @-operators (thus $\downarrow$ does not make an appearance). For most of the paper we use a modal system with only a single $\Box$ (to minimize notational clutter), though we do treat $\neg$, $\wedge$, and $\forall$ as primitives, rather than defining them in terms of λ and $=$, to make it easier for our axiomatization to be compared with existing propositional and first-order modal and hybrid axiomatizations. Our no-frills approach extends to the semantics. We use a constant domain: at all worlds, quantifiers range over a fixed function space constructed over a fixed domain of individuals, and we don't make use of Fitting-style intensionality (so rigidity is

[1] In his thesis, Gallin also proved a (generalized) completeness result for PTQ, which seems to be one of the earliest such results for higher-order modal logic.

a key theme in this paper). We also dispense with type s, working solely with types e and t. In a nutshell: we are simply going to add nominals, a $\Box$, and the $@_i$-operators to Henkin's [23] original higher-order logic.

We call the result BHTT (Basic Hybrid Type Theory). Its axiomatization has a clean and comprehensible form: a Henkin-style higher-order axiomatization, and axioms and rules familiar from the modal and hybrid literature, are woven together with the aid of three new axiom schemas and a hybrid version of the Barcan formulas. This gives rise to a basic completeness result strong enough to automatically support extended completeness results for a wide range of frame classes and multimodal extensions.

2 Syntax and semantics of BHTT

In this section we lay the foundations for the work that follows. We introduce the syntax and *standard* semantics for BHTT, and then motivate and define its *generalized* semantics. We then prove two simple results about rigidity.

Syntax

Definition 1 (Syntax of BHTT) ***Types****: Let t and e be two fixed, but otherwise arbitrary, objects. The set* TYPES *of types of* BHTT *is defined recursively as follows:*

$$\mathsf{TYPES} ::= t \mid e \mid \langle a, b\rangle \textit{ with } a, b \in \mathsf{TYPES} \textit{ and } a \neq t.$$

Meaningful Expressions: *The set* ME_a *of* ***meaningful expressions of type*** *a consists of the basic and complex expressions of type a we now define.*

Basic Expressions*: For each type $a \neq t$, there is a denumerably infinite set* CON_a *of* ***non-logical constants*** *$c_{n,a}$, where n is a natural number. Constants of type t are* ***truth*** *and* ***falsity****, that is,* $\mathsf{CON}_t = \{\top, \bot\}$*. Let* $\mathsf{CON} = \bigcup_a \mathsf{CON}_a$*, and for Δ a set of formulas let* $\mathsf{CON}(\Delta)$ *be the set of constants appearing in formulas of Δ. For each type $a \neq t$, there is a denumerably infinite set* VAR_a *of* ***variables*** *$v_{n,a}$, where n is a natural number. Let* $\mathsf{VAR} = \bigcup_a \mathsf{VAR}_a$*. Finally, for type t, there is a denumerably infinite set* NOM *of nominals i_n, where n is a natural number. Summing up, for each natural number n we have that:*

$$i_n \in \mathsf{ME}_t \mid c_{n,a} \in \mathsf{ME}_a \mid v_{n,a} \in \mathsf{ME}_a \textit{ with } a \neq t.$$

Complex Expressions: *These are recursively generated as follows:*

$$\gamma_{\langle b,a\rangle}\beta_b \in \mathsf{ME}_a \mid \lambda u_b \alpha_a \in \mathsf{ME}_{\langle b,a\rangle} \mid @_i \alpha_a \in \mathsf{ME}_a$$
$$\{\alpha_a = \alpha'_a, \neg\varphi_t, \varphi_t \wedge \psi_t, \forall u_b \varphi_t, \Box\varphi_t\} \subseteq \mathsf{ME}_t,$$

where $\alpha_a, \alpha'_a \in$ ME_a*, $\beta_b \in$* ME_b*, $\gamma_{\langle b,a\rangle} \in$* $\mathsf{ME}_{\langle b,a\rangle}$*, $u_b \in$* VAR_b*, $i \in$* NOM *and $\varphi_t, \psi_t \in$* ME_t*. In what follows, we often explicitly give the type of a meaningful expression (writing, for example, α_a) to emphasize that $\alpha \in$* ME_a*.*

We introduce $\varphi_t \leftrightarrow \psi_t$ as shorthand for $\varphi_t = \psi_t$. The remaining booleans, the existential quantifier $\exists$, and the modal diamond $\Diamond$ are defined as usual.

*Given a meaningful expression α the **set of free variables occurring in** α_a (notation $\mathsf{FREE}(\alpha)$) is defined recursively as follows:*

$$\begin{array}{rcl}
\mathsf{FREE}(\tau) & = & \varnothing \text{ for } \tau \in \mathsf{CON} \cup \mathsf{NOM} \\
\mathsf{FREE}(v) & = & \{v\} \text{ for } v \in \mathsf{VAR} \\
\mathsf{FREE}(\alpha = \beta) = \mathsf{FREE}(\alpha\beta) = \mathsf{FREE}(\alpha \wedge \beta) & = & \mathsf{FREE}(\alpha) \cup \mathsf{FREE}(\beta) \\
\mathsf{FREE}(\neg\alpha) = \mathsf{FREE}(\Box\alpha) & = & \mathsf{FREE}(\alpha) \\
\mathsf{FREE}(\forall u\alpha) = \mathsf{FREE}(\lambda u\alpha) & = & \mathsf{FREE}(\alpha)\backslash\{u\}.
\end{array}$$

*A meaningful expression α_t of type t is called a **sentence** if $\mathsf{FREE}(\alpha_t) = \varnothing$.*

The syntax is (with two exceptions) fairly standard. The two exceptions are the introduction of nominals, and the use of the $@_i$ operators, the two basic tools of hybrid logic. We will discuss these additions in more detail when we have defined the semantics. For the moment, simply note that nominals are of type t and are regarded as forming a distinct syntactic class (they are not constants of type t, of which there are only two, namely $\top$ and $\bot$). Moreover, note that for any expression α_a (of any type a) the result of prefixing it with $@_i$ (where i can be any nominal) yields an expression $@_i\alpha_a$ which is also of type a. Nominals and expressions of the form $@_i\alpha_a$ play a central role in the completeness result: nominals are a key model-building material, and $@_i\alpha_a$ expressions supply the architectural blueprint.

Semantics

Definition 2 (BHTT models) *A **standard structure** (or **standard model**) for BHTT is a pair $\mathcal{M} = \langle \mathcal{S}, \mathsf{F} \rangle$ such that*

1. *$\mathcal{S} = \langle\langle \mathsf{D}_a \rangle_{a \in \mathsf{TYPES}}, W, R\rangle$ is a **standard skeleton**, where:*
   - *(a) $\langle \mathsf{D}_a \rangle_{a \in \mathsf{TYPES}}$, the **standard type hierarchy**, is defined recursively as follows:*

$$\begin{array}{rcl}
\mathsf{D}_t & = & \{T, F\} \text{ is the set of truth values,} \\
\mathsf{D}_e & \neq & \varnothing \text{ is the set of individuals,} \\
\mathsf{D}_{\langle a,b\rangle} & = & \mathsf{D}_b^{\mathsf{D}_a} \text{ is the set of all functions from } \mathsf{D}_a \text{ into } \mathsf{D}_b \\
 & & \text{for } a, b \in \mathsf{TYPES}, a \neq t.
\end{array}$$

   - *(b) $W \neq \varnothing$ is the set of worlds.*
   - *(c) $R \subseteq W \times W$ is the accessibility relation.*
2. *The **denotation function** F assigns to each non-logical constant a function from W to an element in the hierarchy of appropriate type, and to each nominal a function from W to the set of truth values. More precisely:*
   - *(a) For any constant $c_{n,a}$ we define $\mathsf{F}(c_{n,a}) : W \longrightarrow \mathsf{D}_a$. Moreover, $(\mathsf{F}(\top))(w) = T$ and $(\mathsf{F}(\bot))(w) = F$, for any world $w \in W$.*

(b) $\mathsf{F}(i) : W \longrightarrow \{T, F\}$ *such that* $(\mathsf{F}(i))(v) = T$ *for a unique* $v \in W$*. To simplify notation, we sometimes write* $\mathsf{F}(i) = \{v\}$ *and say that* v *is the denotation of* i*.*

Most of the above is standard, familiar from either higher-order or modal logic. The only novelty is the interpretation of nominals. Propositional hybrid logic, which in its earliest form was due to Arthur Prior [31, 32, 6], trades on the idea of using *formulas as terms.* Because nominals are true at precisely one world in any model, they (so to speak) name that world by being true there and nowhere else; and in that way they blur the distinction between terms and formulas. This is what our interpretation of nominals does too: we treat them as formulas (they will be of type t) but the interpretational constraint ensures they can act as "names" of worlds.

Another remark. As we have already mentioned, Montague and Gallin made use of a third type s, the type of possible worlds.[2] Now, in this paper we have restricted ourselves to types e and t. *But nominals are names for possible worlds.* So although we don't have a type s, our object language is very much attuned to entities of this type. BHTT's attunement to worlds will become even more pronounced when we define the semantics of $@_i\alpha_a$, and this attunement is the key to our Henkin construction. Treating formulas as terms takes on new significance in higher-order modal settings; hybridization will allow us to work over the original Church-Henkin type system in a particularly direct fashion.

Definition 3 *An* ***assignment of values to variables*** g *is a function with domain* VAR *such that for any variable* $v_{n,a}$*,* $g(v_{n,a}) \in \mathsf{D}_a$*.*

An assignment g' *is a* v*-variant of* g *if it coincides with* g *on all values except, perhaps, on the value assigned to the variable* v*. We use* g_v^θ *to denote the* v*-variant of* g *whose value for* $v \in \mathsf{VAR}_a$ *is* $\theta \in \mathsf{D}_a$*.*

Definition 4 (BHTT interpretations) *A* ***standard interpretation*** *is a pair* $\langle \mathcal{M}, g\rangle$*, where* $\mathcal{M}$ *is a standard structure for* BHTT *and* g *is a variable assignment on* $\mathcal{M}$*. Given a standard structure* $\mathcal{M} = \langle\langle(\mathsf{D}_a)_{a\in\mathsf{TYPES}}, W, R\rangle, \mathsf{F}\rangle$ *and an assignment* g *we recursively define, for any meaningful expression* α*, the standard interpretation of* α *with respect to the model* $\mathcal{M}$ *and the assignment* g*, at the world* w*, denoted by* $[[\alpha]]^{\mathcal{M},w,g}$*, as follows:*

1. $[[\tau]]^{\mathcal{M},w,g} = (\mathsf{F}(\tau))(w)$*, for* $\tau \in \mathsf{CON} \cup \mathsf{NOM}$
2. $[[v_{n,a}]]^{\mathcal{M},w,g} = g(v_{n,a})$*, for* $v_{n,a} \in \mathsf{VAR}_a$
3. $[[\lambda u_b \alpha_a]]^{\mathcal{M},w,g} = h$*, where* $h : \mathsf{D}_b \longrightarrow \mathsf{D}_a$ *is the function defined by* $h(\theta) = [[\alpha_a]]^{\mathcal{M},w,g_{u_b}^\theta}$*, for any* $\theta \in \mathsf{D}_b$
4. $[[\alpha_{\langle b,a\rangle}\beta_b]]^{\mathcal{M},w,g} = [[\alpha_{\langle b,a\rangle}]]^{\mathcal{M},w,g}([[\beta_b]]^{\mathcal{M},w,g})$
5. $[[\alpha_a = \beta_a]]^{\mathcal{M},w,g} = T$ *iff* $[[\alpha]]^{\mathcal{M},w,g} = [[\beta]]^{\mathcal{M},w,g}$

[2] Strictly speaking, Montague's type s was the type of world/time pairs. But this is irrelevant to the present discussion. The important point is that type s is *the type of the entities at which we evaluate formulas* be they world/time pairs, worlds, times, epistemic states, or something else entirely.

6. $[[\neg\varphi_t]]^{\mathcal{M},w,g} = T$ *iff* $[[\varphi_t]]^{\mathcal{M},w,g} = F$

7. $[[\varphi_t \wedge \varphi'_t]]^{\mathcal{M},w,g} = T$ *iff* $[[\varphi_t]]^{\mathcal{M},w,g} = T$ *and* $[[\varphi'_t]]^{\mathcal{M},w,g} = T$

8. $[[\forall x_a \varphi_t]]^{\mathcal{M},w,g} = T$ *iff for all* $\theta \in \mathsf{D}_a$ $[[\varphi]]^{\mathcal{M},w,g^{\theta}_{x_a}} = T$

9. $[[\Box\varphi_t]]^{\mathcal{M},w,g} = T$ *iff for all* $v \in W$ *such that* $\langle w, v\rangle \in R$, $[[\varphi_t]]^{\mathcal{M},v,g} = T$

10. $[[@_i\alpha_a]]^{\mathcal{M},w,g} = [[\alpha_a]]^{\mathcal{M},v,g}$ *where* $\mathsf{F}(i) = \{v\}$.

The last clause is the novelty. As promised, $@_i\alpha$ is an expression (of the same type as α) that rigidly returns the value of α at the i-world. BHTT's ability to inspect named worlds and determine the semantic values of expressions of arbitrary types at them, is what enables it to specify the blueprint for a model that will satisfy a (consistent) set of formulas.

We now come to Henkin's crucial idea for taming higher-order logic. The standard semantics just defined (ignore for the moment the modal and hybrid components) is the usual semantics for higher-order logic and it is logically intractable: if we define validity as truth in all standard structures, we have a complex (indeed, provably unaxiomatizable) notion of validity. The situation is rendered more unsatisfactory by the existence of plausible looking candidate axiomatizations. These axiomatizations seem to capture all that can be said about higher-order semantics, but in the face of the unaxiomatizability result they must all be incomplete. A puzzling situation. The way around the impasse was provided in 1950 by Henkin, who proposed a more liberal notion of interpretation for higher-order logic (see Henkin [23, 24] and Manzano [27]). His notion of generalized interpretations (defined below) simultaneously lowers the logical complexity of validity (as there are more generalized structures than standard ones, it is, so to speak, easier for a formula to be falsified, and indeed, higher-order validity becomes recursively enumerable) and makes clear just why those plausible looking axiomatizations were so plausible: they are complete with respect to Henkin's generalized semantics.[3] All of which provides the background motivation for the following definitions:

Definition 5 (BHTT skeletons and structures) *A* ***type hierarchy*** *is a family* $\langle \mathsf{D}_a\rangle_{a\in\mathsf{TYPES}}$ *of sets defined recursively as follows:*

$$\begin{array}{rcl} \mathsf{D}_e & \neq & \varnothing \\ \mathsf{D}_t & = & \{T, F\} \\ \mathsf{D}_{\langle a,b\rangle} & \subseteq & \mathsf{D}_b^{\mathsf{D}_a} \text{ for } a, b \in \mathsf{TYPES}, a \neq t. \end{array}$$

A ***skeleton*** $\mathcal{S} = \langle\langle \mathsf{D}_a\rangle_{a\in\mathsf{TYPES}}, W, R\rangle$ *is a triple satisfying all the conditions of a standard skeleton except that* $\langle \mathsf{D}_a\rangle_{a\in\mathsf{TYPES}}$ *is a (not necessarily standard) type hierarchy.*

A ***structure*** *(or* ***model****) is a pair* $\mathcal{M} = \langle \mathcal{S}, \mathsf{F}\rangle$ *where* $\mathcal{S}$ *is a skeleton and* F *is a denotation function.*

[3] While this is true for axiomatic systems, the situation for sequent calculi and tableaux systems is more subtle, and reasonable looking cut-free systems may be incomplete even with respect to the generalized semantics; see Fitting [15], pages xiv–xv and elsewhere, for further discussion.

The idea of using type hierarchies as just defined, rather than the full function space hierarchy, is the big step forward. To interpret expressions of type $\langle a, b\rangle$ we don't need all the set-theoretically possible functions from D_a to D_b. However we do need to ensure that we have chosen *enough* functions to interpret the expressions of our language. Hence we must insist upon closure under interpretation. This prompts the following definition:

Definition 6 (General interpretation) *A* ***general interpretation*** *is a pair $\langle \mathcal{M}, g\rangle$ where $\mathcal{M}$ is a structure, g a variable assignment, and for any meaningful expression in ME_a, its interpretation (as given by Definition 4) is in D_a.*

Summing up: generalized interpretations may lack some set-theoretical possibilities (they need not contain the full set-theoretical function hierarchy) but they are not permitted to lack any structure that the language can actually see. From now on we will work with this more liberal notion of interpretation. That is, from now on, given a (not necessarily standard) model $\mathcal{M}$, an assignment g, and an expression α, we will allow ourselves to interpret α on $\mathcal{M}$ using the clauses given in Definition 4.

We are now ready for the key semantic definitions. Clearly all standard interpretations are generalized interpretations. Hence the following definitions really do generalize the standard notions:

Definition 7 (Consequence and validity) *Let $\Gamma \cup \{\varphi\} \subseteq \mathsf{ME}_t$ and $\mathcal{M}$ be a structure. We define consequence and validity as follows:*

Consequence: *$\Gamma \models \varphi$ iff for all general interpretations $\langle \mathcal{M}, g\rangle$ and all $w \in W$, if $[[\gamma]]^{\mathcal{M},w,g} = T$ for all $\gamma \in \Gamma$ then $[[\varphi]]^{\mathcal{M},w,g} = T$.*

Validity: *$\models \varphi$ iff $\varnothing \models \varphi$.*

Variables, substitution, and rigidity

Before proceeding, we need two small technical lemmas concerning the interpretation of nominals and free variables.

Lemma 8 (Coincidence lemma for nominals) *Let $\langle \mathcal{M}, g\rangle$ and $\langle \mathcal{M}^*, g\rangle$ be two general interpretations such that $\mathcal{M} = \langle \mathcal{S}, \mathsf{F}\rangle$ and $\mathcal{M}^* = \langle \mathcal{S}, \mathsf{F}^*\rangle$ have the same skeleton and F agrees with F^* for all arguments except the nominal i. Let $\alpha_a \in \mathsf{ME}_a$ be any meaningful expression in which i does not occur. Then, for any world w, $[[\alpha_a]]^{\mathcal{M},w,g} = [[\alpha_a]]^{\mathcal{M}^*,w,g}$*

Proof Straightforward. $\dashv$

Lemma 9 (Coincidence lemma for variables) *If g and h are assignments that agree on the free variables of $\alpha_a \in \mathsf{ME}_a$ (that is, $g_{\restriction \mathsf{FREE}(\alpha_a)} = h_{\restriction \mathsf{FREE}(\alpha_a)}$), and $\langle \mathcal{M}, g\rangle$ and $\langle \mathcal{M}, h\rangle$ are general interpretations, then for any world w we have that $[[\alpha_a]]^{\mathcal{M},w,g} = [[\alpha_a]]^{\mathcal{M},w,h}$.*

Proof By induction on the construction of meaningful expressions. We give the cases for $\lambda u_b \alpha_a$ and $\forall u_a \varphi$:

- $[[\lambda u_b \beta_a]]^{\mathcal{M},w,g}$ is the function with domain D_b, such that for any $\theta \in \mathsf{D}_b$ its value is $[[\beta_a]]^{\mathcal{M},w,g^{\theta}_{u_b}}$. We know that $g^{\theta}_{u_b}(u_b) = h^{\theta}_{u_b}(u_b)$ and also that $g^{\theta}_{u_b}(v_q) = h^{\theta}_{u_b}(v_q)$ for any $v_q \in \mathsf{FREE}(\lambda u_b \beta_a) = \mathsf{FREE}(\beta_a) \backslash \{u_b\}$. Thus, by the induction hypothesis, $[[\beta_a]]^{\mathcal{M},w,g^{\theta}_{u_b}} = [[\beta_a]]^{\mathcal{M},w,h^{\theta}_{u_b}}$ for all $\theta \in \mathsf{D}_b$. Therefore $[[\lambda u_b \beta_a]]^{\mathcal{M},w,g} = [[\lambda u_b \beta_a]]^{\mathcal{M},w,h}$, because both are functions returning the same values for all arguments.
- $[[\forall u_b \varphi]]^{\mathcal{M},w,g} = T$ iff for all $\theta \in \mathsf{D}_b$ $[[\varphi]]^{\mathcal{M},w,g^{\theta}_{u_b}} = T$ iff for all $\theta \in \mathsf{D}_b$ $[[\varphi]]^{\mathcal{M},w,h^{\theta}_{u_b}} = T$ iff $[[\forall u_b \varphi]]^{\mathcal{M},w,h} = T$.

$\dashv$

We will also make heavy use of substitution (particularly of rigid terms). Let us define this notion precisely.

Definition 10 (Variable substitution) *For all $\alpha_a \in \mathsf{ME}_a$, the* ***substitution of*** γ_c ***for a variable*** v_c ***in*** α_a*, written $\alpha_a \frac{\gamma_c}{v_c}$, is inductively defined as follows:*

1. $\tau \frac{\gamma_c}{v_c} := \tau$ *for* $\tau \in \mathsf{CON} \cup \mathsf{NOM}$
2. $v_a \frac{\gamma_c}{v_c} := \begin{cases} \gamma_c & \text{if } v_a \in \mathsf{VAR} \text{ and } v_a = v_c \\ v_a & \text{if } v_a \in \mathsf{VAR} \text{ and } v_a \neq v_c \end{cases}$
3. $(\lambda u_p \beta_b)\frac{\gamma_c}{v_c} := \begin{cases} \lambda u_p \beta_b & \text{if } v_c \notin \mathsf{FREE}(\lambda u_p \beta_b) \\ \lambda u_p (\beta_b \frac{\gamma_c}{v_c}) & \text{if } v_c \in \mathsf{FREE}(\lambda u_p \beta_b), \\ & \quad u_p \notin \mathsf{FREE}(\gamma_c) \\ \lambda x_p (\beta_b \frac{x_p}{u_p})\frac{\gamma_c}{v_c} & \text{if } v_c \in \mathsf{FREE}(\lambda u_p \beta_b), \\ & \quad u_p \in \mathsf{FREE}(\gamma_c), x_p \text{ new} \end{cases}$
4. $(\beta_{\langle b,a\rangle} \delta_b)\frac{\gamma_c}{v_c} := \beta_{\langle b,a\rangle}\frac{\gamma_c}{v_c} \delta_b \frac{\gamma_c}{v_c} \mid (\beta_b = \delta_b)\frac{\gamma_c}{v_c} := \beta_b \frac{\gamma_c}{v_c} = \delta_b \frac{\gamma_c}{v_c}$
5. $(\neg\varphi)\frac{\gamma_c}{v_c} := \neg\varphi\frac{\gamma_c}{v_c} \mid (\varphi \wedge \psi)\frac{\gamma_c}{v_c} := \varphi\frac{\gamma_c}{v_c} \wedge \psi\frac{\gamma_c}{v_c} \mid (\Box\psi)\frac{\gamma_c}{v_c} := \Box(\psi\frac{\gamma_c}{v_c})$
6. $(\forall u_p \psi)\frac{\gamma_c}{v_c} := \begin{cases} \forall u_p \psi & \text{if } v_c \notin \mathsf{FREE}(\forall u_p \psi) \\ \forall u_p \psi \frac{\gamma_c}{v_c} & \text{if } v_c \in \mathsf{FREE}(\forall u_p \psi), \\ & \quad u_p \notin \mathsf{FREE}(\gamma_c) \\ \forall x_p (\psi \frac{x_p}{u_p})\frac{\gamma_c}{v_c} & \text{if } v_c \in \mathsf{FREE}(\forall u_p \psi), \\ & \quad u_p \in \mathsf{FREE}(\gamma_c), x_p \text{ new} \end{cases}$
7. $(@_i \beta_b)\frac{\gamma_c}{v_c} := @_i(\beta_b \frac{\gamma_c}{v_c})$.

It is time to define one of the paper's key concepts: ***rigid expressions***. These are expressions that have the same value at all worlds; good examples are $\top$ and $\bot$ (their rigidity is hard-wired into the definition of what denotation functions F are), variables of all types (after all, variable denotations are determined *globally* and *directly* by assignment functions), and expressions prefixed by an @ operator (indeed, these operators were designed with rigidification in mind). Rigid expressions play a key role in our axiomatization and equivalence classes of rigid expressions are the building blocks used in our model construction.

Definition 11 (Rigid meaningful expresions) *The set* RIGIDS *of* ***rigid meaningful expressions*** *is defined inductively as follows:*

$$\mathsf{RIGIDS} ::= \bot \,|\, \top \,|\, v_a \,|\, @_i\theta_a \,|\, \lambda v_b \alpha_a \,|\, \gamma_{\langle b,a\rangle}\beta_b \,|\, \alpha_b = \beta_b \,|\, \neg\varphi_t \,|\, \varphi_t \wedge \psi_t \,|\, \forall v_a \varphi_t,$$

where $\theta_a \in \mathsf{ME}_a$ *and* α_a*,* β_b*,* $\gamma_{\langle b,a\rangle}$*,* φ_t*,* $\psi_t \in \mathsf{RIGIDS}$*. We say that* $\alpha \in \mathsf{RIGIDS}_a$ *if* α *is rigid and of type* a*, that is, if* $\alpha \in \mathsf{RIGIDS} \cap \mathsf{ME}_a$*.*

Lemma 12 *Let* $\langle \mathcal{M}, g\rangle$ *be a general interpretation. If* $\gamma \in \mathsf{RIGIDS}$ *then* $[[\gamma]]^{\mathcal{M},w,g} = [[\gamma]]^{\mathcal{M},v,g}$ *for all* $w, v \in W$*.*

Proof By induction on the construction of rigid expressions. We give the cases for $\lambda v_b \alpha_a$ and $\forall v_a \varphi$:

- If γ is of the form $\lambda v_b \alpha_a$ with α_a rigid, then $[[\lambda v_b \alpha_a]]^{\mathcal{M},w,g}$ is the function h with domain D_b, such that $h(\theta) = [[\alpha_a]]^{\mathcal{M},w,g^{\theta}_{v_b}}$, for any $\theta \in \mathsf{D}_b$. On the other hand, $[[\lambda v_b \alpha_a]]^{\mathcal{M},v,g}$ is the function h' with domain D_b, such that $h'(\theta) = [[\alpha_a]]^{\mathcal{M},v,g^{\theta}_{v_b}}$, for any $\theta \in \mathsf{D}_b$. Now, $[[\alpha_a]]^{\mathcal{M},w,g^{\theta}_{v_b}} = [[\alpha_a]]^{\mathcal{M},v,g^{\theta}_{v_b}}$, using the induction hypothesis for α_a, and thus $[[\lambda v_b \alpha_a]]^{\mathcal{M},w,g} = [[\lambda v_b \alpha_a]]^{\mathcal{M},v,g}$ for all $w, v \in W$.
- If γ is of the form $\forall v_a \varphi$ with φ rigid, then $[[\forall v_a \varphi]]^{\mathcal{M},w,g} = T$ iff $[[\varphi]]^{\mathcal{M},w,g^{\theta}_{v_a}} = T$ for all $\theta \in \mathsf{D}_a$ iff $[[\varphi]]^{\mathcal{M},v,g^{\theta}_{v_a}} = T$ for all $\theta \in \mathsf{D}_a$ (by the induction hypothesis for rigid φ) iff $[[\forall v_a \varphi]]^{\mathcal{M},v,g} = T$. Note that here we make use of the fact that we are quantifying over a constant domain.

$\dashv$

Rigid expressions are well-behaved with respect to substitution:

Lemma 13 (Rigid substitution) *Let* $\langle \mathcal{M}, g\rangle$ *be a general interpretation. Then for all worlds* w*, all* $\alpha_a \in \mathsf{ME}_a$*, all* $\gamma_c \in \mathsf{RIGIDS}_c$ *and any variable* v_c *of type* c*:*

$$[[\alpha_a \frac{\gamma_c}{v_c}]]^{\mathcal{M},w,g} = [[\alpha_a]]^{\mathcal{M},w,g^{\overline{\gamma_c}}_{v_c}}$$

where $\overline{\gamma_c}$ *is an abbreviation for* $[[\gamma_c]]^{\mathcal{M},w,g}$*.*

Proof Straightforward by induction on the construction of meaningful expressions, with the help of Lemma 9. $\dashv$

3 Axiomatization

We now introduce deductions (formal proofs) for BHTT. We will select an infinite set of logical axioms, and several rules of proof which will enable us to prove certain meaningful expressions of type t .

Rules of proof

1. **Modus Ponens:** If $\vdash \varphi$ and $\vdash \varphi \to \psi$, then $\vdash \psi$.
2. **Generalizations:**
   (a) **Gen$_\Box$:** If $\vdash \varphi$, then $\vdash \Box\varphi$.
   (b) **Gen$_@$:** If $\vdash \varphi$, then $\vdash @_i\varphi$.
   (c) **Gen$_\forall$:** If $\vdash \varphi$, then $\vdash \forall x_a\varphi$.
3. **Rigid replacement:** If $\vdash \varphi$, then $\vdash \varphi'$, where φ' is obtained from φ by uniformly replacing nominals by nominals, variables of type a by rigid expressions of type a, and vice-versa (that is, we can replace rigid expressions of type a by variables of type a too).
4. **Name:** If $\vdash @_i\varphi$ and i does not occur in φ, then $\vdash \varphi$.
5. **Bounded Generalization:** If $\vdash @_i\Diamond j \to @_j\varphi$ and $j \neq i$ and j does not occur in φ, then $\vdash @_i\Box\varphi$.

These are all standard rules drawn from the literature on modal and hybrid logic. For a detailed discussion of the Name and Bounded Generalization rules, see Blackburn and Ten Cate [8]. The restriction in the rigid replacement rule that nominals must replace nominals is standard in hybrid logic; it reflects the fact that nominals embody namelike information, and replacement must respect this. The additional restriction we have imposed (that variables can only be freely replaced by rigid terms and vice-versa) reflects the fact that assignment functions interpret variables rigidly, and replacement must respect this too.

Axioms

We will give the logical axioms as general schemas.

1. **Tautologies**: All BHTT instances of propositional tautologies.
2. **Distributivity axioms:**
   (a) **$\Box$-distributivity:** $\vdash \Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi)$.
   (b) **@-distributivity:** $\vdash @_i(\varphi \to \psi) \to (@_i\varphi \to @_i\psi)$.
   (c) **$\forall$-distributivity:** $\vdash \forall x_b(\varphi \to \psi) \to (\forall x_b\varphi \to \forall x_b\psi)$.
3. **Quantifier axioms:**
   (a) **$\forall$-elimination:** For β_b rigid, $\vdash \forall x_b\varphi \to \varphi\frac{\beta_b}{x_b}$.
   (b) **Vacuous:** $\vdash \varphi \to \forall y_a\varphi$, where y_a does not occur free in φ.
4. **Equality axioms:**
   (a) **Reflexivity:** $\vdash \alpha_a = \alpha_a$.
   (b) **Substitution:** For α_a, β_a rigid, $\vdash \alpha_a = \beta_a \to (\delta_c\frac{\alpha_a}{v_a} = \delta_c\frac{\beta_a}{v_a})$.
5. **Functional axioms:**
   (a) **Extensionality:** $\vdash \forall v_b(\gamma_{\langle b,a\rangle}v_b = \delta_{\langle b,a\rangle}v_b) \to \gamma_{\langle b,a\rangle} = \delta_{\langle b,a\rangle}$, where v_b does not occur free in $\gamma_{\langle b,a\rangle}$ or $\delta_{\langle b,a\rangle}$.

(b) **β-conversion:** For rigid β_b, $\vdash (\lambda x_b \alpha_a)\beta_b = \alpha_a \frac{\beta_b}{x_b}$.

(c) **η-conversion:** $\vdash (\lambda x_b \gamma_{\langle b,a\rangle} x_b) = \gamma_{\langle b,a\rangle}$, where x_b is not free in $\gamma_{\langle b,a\rangle}$.

6. **Axioms for @:**

   (a) **Selfdual:** $\vdash @_i\varphi \leftrightarrow \neg @_i \neg\varphi$.

   (b) **Intro:** $\vdash i \to (\varphi \leftrightarrow @_i\varphi)$.

   (c) **Back:** $\vdash \Diamond @_i\varphi \to @_i\varphi$.

   (d) **Ref:** $\vdash @_i i$.

   (e) **Agree:** $\vdash @_i @_j \alpha_a = @_j \alpha_a$.

7. **Domain Axioms:**

   (a) **Hybrid Barcan:** $\vdash \forall x_b @_i\varphi \leftrightarrow @_i \forall x_b \varphi$.

8. **New Axioms:**

   (a) **Equality-at-*i*:** $\vdash @_i(\beta_b = \delta_b) = (@_i\beta_b = @_i\delta_b)$.

   (b) **Rigid function application:** $\vdash @_i(\gamma_{\langle b,a\rangle}\beta_b) = (@_i\gamma_{\langle b,a\rangle})(@_i\beta_b)$.

   (c) **Rigids are rigid:** If α_a is rigid then $\vdash @_i\alpha_a = \alpha_a$.

The axiomatization is not designed to be minimal, it is designed to be perspicuous and to make use of well-known axioms from propositional and first-order modal and hybrid logic, and higher-order logic.[4] Indeed (if we ignore side restrictions to rigid terms) almost all the above axioms should be familiar. The only novelties are **Equality-at-i** (Axiom 8a), **Rigid function application** (Axiom 8b), and **Rigids are rigid** (Axiom 8c). These, together with **Hybrid Barcan** (Axiom 7a) play a key role in the model construction. Note that Hybrid Barcan combines the standard modal Barcan and converse Barcan formulas, but with $@_i$ taking the place of $\Box$. Hybrid Barcan will later lead us to what we call the Rigid Representatives Theorem, which will help us to build the function hierarchy required for the completeness proof.

Definition 14 *A **deduction** of φ is a finite sequence $\alpha_1, \ldots, \alpha_n$ of expressions such that $\alpha_n := \varphi$ and for every $1 \leq i \leq n-1$, either α_i is an axiom, or α_i is obtained from previous expressions in the sequence using the rules of proof. We will write $\vdash \varphi$ whenever we have such a sequence and we will say that φ is a* BHTT*-**theorem**.*

Definition 15 *If $\Gamma \cup \{\varphi\}$ is a set of meaningful expressions of type t, a **deduction of** φ **from** Γ is a deduction of $\vdash \gamma_1 \wedge \ldots \wedge \gamma_n \to \varphi$ where $\{\gamma_1, \ldots, \gamma_n\} \subseteq \Gamma$. We say that a meaningful expression φ of type t is **deducible from a set of expressions** Γ, and we write $\Gamma \vdash \varphi$, iff there is a deduction of φ from Γ.*

Theorem 16 (Soundness) *For all $\varphi \in \mathsf{ME}_t$, we have $\vdash \varphi$ implies $\models \varphi$.*

[4] One obvious redundancy is the @-distributivity axiom: this is a straightforward consequence of Equality-at-i (Axiom 8a).

Proof Straightforward but tedious. $\dashv$

What is provable in BHTT? We shall meet many examples in the course of the completeness proof. Some of these will be important for the completeness proof, but their derivability will be fairly clear. For example, we shall need to make use of existential forms of Hybrid Barcan (that is, expressions of the form $\vdash @_i \exists x_b \varphi \leftrightarrow \exists x_b @_i \varphi$) but it is simple to check that BHTT is powerful enough to derive these from the universal forms given as axioms. Others, such as the Bridge principle $\vdash @_i \Diamond j \wedge @_j \varphi \rightarrow @_i \Diamond \varphi$, are trickier. In any case, we have gathered together all the required BHTT-theorems into the Appendix at the end of the paper. Note that many of the theorems are essentially hybrid rather than higher-order in nature (both Bridge and Barcan are good examples of this). As we have noted, it is the nominals and $@_i$ operators that carry much of the load of specifying the required model, and this is reflected in the form of the theorems we need to derive.

4 Maximal consistent sets

In this section we define and explore maximal consistent sets of BHTT sentences with various useful properties, prove the variant of Lindenbaum's Lemma we shall require, and then prove a result we call the Rigid Representatives Theorem, which will give us valuable information about the building blocks of our type hierarchy.

Definition 17 $\Delta \subseteq \mathsf{ME}_t$ *is* ***inconsistent*** *(or contradictory) iff for every* $\varphi \in \mathsf{ME}_t$*,* $\Delta \vdash \varphi$*.* Δ *is* ***consistent*** *iff it is not inconsistent.* Δ *is a* ***maximally consistent*** *set iff* Δ *is consistent and whenever* $\varphi \in \mathsf{ME}_t$ *and* $\varphi \notin \Delta$*, then* $\Delta \cup \{\varphi\}$ *is inconsistent.*

The following four lemmas note some easy consequences of the definitions and rules of the calculus.

Lemma 18 *Let* $\Delta, \Gamma \subseteq \mathsf{ME}_t$ *and* $\varphi \in \mathsf{ME}_t$*. Then:*

1. *If* Δ *is consistent and* $\Gamma \subseteq \Delta$*, then* Γ *is consistent.*
2. *If* Δ *is inconsistent and* $\Delta \subseteq \Gamma$*, then* Γ *is inconsistent.*
3. $\Delta \subseteq \mathsf{ME}_t$ *is inconsistent iff for some* $\varphi \in \mathsf{ME}_t$*,* $\Delta \vdash \varphi$ *and* $\Delta \vdash \neg\varphi$*.*
4. $\Delta \subseteq \mathsf{ME}_t$ *is inconsistent iff* $\Delta \vdash \bot$*.*
5. *If* Δ *is consistent, then for all* $\varphi \in \mathsf{ME}_t$ *such that* $\Delta \vdash \varphi$ *we have* $\Delta \cup \{\varphi\}$ *is consistent.*
6. Δ *is consistent iff every finite subset of* Δ *is consistent.*

Lemma 19 *Let* $\Delta \subseteq \mathsf{ME}_t$ *be a maximal consistent set and* φ*,* $\psi \in \mathsf{ME}_t$*. Then:*

1. $\Delta \vdash \varphi$ *iff* $\varphi \in \Delta$*.*
2. *If* $\vdash \varphi$ *then* $\varphi \in \Delta$*.*

3. $\neg\varphi \in \Delta$ *iff* $\varphi \notin \Delta$.
4. $\varphi \in \Delta$ *iff* $\neg\varphi \notin \Delta$.
5. $\varphi \wedge \psi \in \Delta$ *iff* $\varphi \in \Delta$ *and* $\psi \in \Delta$.
6. *Either* $\varphi \in \Delta$ *or* $\neg\varphi \in \Delta$ *but not both.*
7. *If* $\Delta \cup \{\varphi\} \vdash \psi$ *and* $\Delta \cup \{\psi\} \vdash \varphi$ *then* $\varphi \in \Delta$ *iff* $\psi \in \Delta$.

Lemma 20 *Let* Δ *be a maximal consistent set. If* $@_i\Box\varphi \in \Delta$ *then for any nominal* j *we have:* $@_i\Diamond j \in \Delta$ *implies* $@_j\varphi \in \Delta$.

Proof Let $@_i\Box\varphi \in \Delta$. Then $\neg @_i\Diamond\neg\varphi \in \Delta$ using the definition of $\Diamond$ and the fact that $@_i$ is selfdual (Axiom 6a). So let j be a nominal such that $@_i\Diamond j \in \Delta$; assume that $@_j\varphi \notin \Delta$. Then $@_j\neg\varphi \in \Delta$, using Part 3 of the previous lemma and the selfduality of $@_j$. Now $\vdash @_i\Diamond j \wedge @_j\neg\varphi \rightarrow @_i\Diamond\neg\varphi$ by Bridge (Claim 57). Hence as Δ is maximal consistent, $@_i\Diamond\neg\varphi \in \Delta$. But this contradicts the consistency of Δ as $\neg @_i\Diamond\neg\varphi \in \Delta$. $\dashv$

Lemma 21 *Let* Δ *be a maximal consistent set. If* $@_i\forall x_a\varphi \in \Delta$*, then* $@_i\varphi\frac{\alpha_a}{x_a} \in \Delta$ *for all* $\alpha_a \in \mathsf{RIGIDS}_a$.

Proof For any rigid expression α_a we have $\vdash \forall x_a\varphi \rightarrow \varphi\frac{\alpha_a}{x_a}$ (Axiom 3a). Applying @-generalization and @-distributivity yields $\vdash @_i\forall x_a\varphi \rightarrow @_i\varphi\frac{\alpha_a}{x_a}$, so this formula must be in Δ by maximal consistency. Hence if $@_i\forall x_a\varphi \in \Delta$ then $@_i\varphi\frac{\alpha_a}{x_a} \in \Delta$. $\dashv$

With these preliminaries noted, we are ready to begin. As we have said, our completeness proof follows Henkin's strategy. The key idea is to build a model out of the information contained in a maximal consistent set of sentences — but not any maximal consistent set will do. When dealing with the quantifiers, Henkin demanded that each $\exists$-formula be witnessed by an appropriate constant, for he defined the domain of quantification out of equivalences classes of constants. We need to do this, but we also need to demand that each $\Diamond$-formula be witnessed by a nominal, for we shall define the worlds and the accessibility relation out of equivalence classes of nominals. Furthermore, we need the maximal consistent set to contain at least one nominal; in the model we shall eventually construct, the equivalence class containing this nominal will be the world that satisfies the consistent set of sentences. In short, we shall be demanding the following three properties:

Definition 22 *Let* Σ *be a set of meaningful expressions.*

1. Σ *is* named *iff one of its elements is a nominal.*
2. Σ *is* $\Diamond$-saturated *iff for all expressions* $@_i\Diamond\varphi \in \Sigma$ *there is a nominal* $j \in \mathsf{NOM}$ *such that* $@_i\Diamond j \in \Sigma$ *and* $@_j\varphi \in \Sigma$.
3. Σ *is* $\exists$-saturated *iff for all expressions* $@_i\exists x_a\varphi \in \Sigma$ *there is a constant* $c_a \in \mathsf{CON}_a$ *such that* $@_i\varphi\frac{@_ic_a}{x_a} \in \Sigma$.

Note the similarity between Clauses 2 and 3: $\Diamond$-saturation is a clear modal analogue of Henkin's notion of $\exists$-saturation. This similarity is underlined by the following two lemmas. First, $\exists$-saturation guarantees us the following:

Lemma 23 *Let Δ be maximal consistent and $\exists$-saturated. If $@_i(\varphi\frac{@_i c_a}{x_a}) \in \Delta$ for all $c_a \in \mathsf{CON}(\Delta)$ then $@_i \forall x_a \varphi \in \Delta$.*

Proof Let $@_i(\varphi\frac{@_i c_a}{x_a}) \in \Delta$ for all $c_a \in \mathsf{CON}(\Delta)$. Assume that $@_i \forall x_a \varphi \notin \Delta$. Then $\neg @_i \forall x_a \varphi \in \Delta$. Therefore $@_i \exists x_a \neg\varphi \in \Delta$, using the definition of $\exists$ and Axiom 6a. As Δ is maximal consistent and $\exists$-saturated, there exists a constant $c_a \in \mathsf{CON}(\Delta)$ such that $@_i(\neg\varphi\frac{@_i c_a}{x_a}) \in \Delta$. Then using the selfduality of $@_i$, we have $\neg @_i(\varphi\frac{@_i c_a}{x_a}) \in \Delta$, contradicting the consistency of Δ. $\dashv$

But now consider the following lemma. Its proof trades on $\Diamond$-saturation, but the underlying strategy is identical:

Lemma 24 *Let Δ be maximal consistent and $\Diamond$-saturated, and suppose that for all nominals j, if $@_i \Diamond j \in \Delta$ then $@_j \varphi \in \Delta$ too. Then $@_i \Box\varphi \in \Delta$.*

Proof Suppose Δ satisfies the statement of the lemma. Now assume that $@_i \Box\varphi \notin \Delta$. Then $\neg @_i \Box\varphi \in \Delta$. Therefore $@_i \Diamond\neg\varphi \in \Delta$, using the definition of $\Diamond$ and Axiom 6a. As Δ is maximal consistent and $\Diamond$-saturated, there exists a nominal j such that $@_i \Diamond j \in \Delta$ and $@_j \neg\varphi \in \Delta$. Then $\neg @_j \varphi \in \Delta$ and $@_j \varphi \in \Delta$, contradicting the consistency of Δ. $\dashv$

As our Henkin proof is for a higher-order logic, part of our completeness proof will involve constructing a type-hierachy. The following lemma, which trades on $\exists$-saturation, will help us do this.

Lemma 25 *Let Δ be maximal consistent and $\exists$-saturated, and let $\gamma_{\langle b,a\rangle}$ and $\gamma'_{\langle b,a\rangle}$ be rigid expressions of type $\langle b,a\rangle$. If for all $c_b \in \mathsf{CON}(\Delta)$ we have $\gamma_{\langle b,a\rangle} @_i c_b = \gamma'_{\langle b,a\rangle} @_i c_b \in \Delta$ then $@_i \forall v_b(\gamma_{\langle b,a\rangle} v_b = \gamma'_{\langle b,a\rangle} v_b) \in \Delta$ for $v_b \notin \mathsf{FREE}(\gamma_{\langle b,a\rangle}) \cup \mathsf{FREE}(\gamma'_{\langle b,a\rangle})$.*

Proof Let $\gamma_{\langle b,a\rangle} @_i c_b = \gamma'_{\langle b,a\rangle} @_i c_b \in \Delta$ for all $c_b \in \mathsf{CON}_b(\Delta)$. We want to prove that $\Delta \vdash @_i \forall v_b(\gamma_{\langle b,a\rangle} v_b = \gamma'_{\langle b,a\rangle} v_b)$. Suppose for the sake of contradiction that $@_i \forall v_b(\gamma_{\langle b,a\rangle} v_b = \gamma'_{\langle b,a\rangle} v_b) \notin \Delta$. Then $\neg @_i \forall v_b(\gamma_{\langle b,a\rangle} v_b = \gamma'_{\langle b,a\rangle} v_b) \in \Delta$, since Δ is maximally consistent. Thus $@_i \exists v_b \neg(\gamma_{\langle b,a\rangle} v_b = \gamma'_{\langle b,a\rangle} v_b) \in \Delta$ using Axiom 6a and the definition of $\exists$. Since Δ is $\exists$-saturated, there is a $c_b \in \mathsf{CON}_b(\Delta)$ such that $@_i \neg(\gamma_{\langle b,a\rangle} @_i c_b = \gamma'_{\langle b,a\rangle} @_i c_b) \in \Delta$. So $\Delta \vdash @_i \neg(\gamma'_{\langle b,a\rangle} @_i c_b = \gamma_{\langle b,a\rangle} @_i c_b)$ and thus $\Delta \vdash \neg @_i(\gamma'_{\langle b,a\rangle} @_i c_b = \gamma_{\langle b,a\rangle} @_i c_b)$ by Axiom 6a.

But by hypothesis $\Delta \vdash \gamma_{\langle b,a\rangle} @_i c_b = \gamma'_{\langle b,a\rangle} @_i c_b$. Hence

$$\Delta \vdash @_i(\gamma_{\langle b,a\rangle} @_i c_b = \gamma'_{\langle b,a\rangle} @_i c_b) = \gamma_{\langle b,a\rangle} @_i c_b = \gamma'_{\langle b,a\rangle} @_i c_b$$

using Axiom 8c and the fact that $\gamma_{\langle b,a\rangle} @_i c_b = \gamma'_{\langle b,a\rangle} @_i c_b$ is rigid. Thus $\Delta \vdash @_i(\gamma_{\langle b,a\rangle} @_i c_b = \gamma'_{\langle b,a\rangle} @_i c_b)$ by modus ponens and the definition of $\leftrightarrow$, contradicting Δ's consistency. $\dashv$

We must now prove that any consistent set of formulas can be extended to a maximal consistent set with all three desirable properties. We need, in short, the following version of Lindenbaum's Lemma:

Lemma 26 (Lindenbaum) *Let Σ be a consistent set of formulas. Then Σ can be extended to a maximal consistent set Σ^ω that is named, $\Diamond$-saturated and $\exists$-saturated.*

Proof Let $\{i_n\}_{n\in\omega}$ be an enumeration of a countably infinite set of new nominals, $\{c_{n,a}\}_{n\in\omega}$ an enumeration of a countably infinite set of new constants of type a, and $\{\varphi_n\}_{n\in\omega}$ an enumeration of the formulas of the extended language. We will build $\{\Sigma^n\}_{n\in\omega}$, a family of subsets of ME_t, by induction:

- $\Sigma^0 = \Sigma \cup \{i_0\}$.
- Assume that Σ^n has already been built. To define Σ^{n+1} we distinguish four cases:
  1. $\Sigma^{n+1} = \Sigma^n$, if $\Sigma^n \cup \{\varphi_n\}$ is inconsistent.
  2. $\Sigma^{n+1} = \Sigma^n \cup \{\varphi_n\}$, if $\Sigma^n \cup \{\varphi_n\}$ is consistent and φ_n is not of the form $@_i\Diamond\psi$ or $@_i\exists x_a\psi$.
  3. $\Sigma^{n+1} = \Sigma^n \cup \{\varphi_n, @_i\Diamond i_m, @_{i_m}\psi\}$, if $\Sigma^n \cup \{\varphi_n\}$ is consistent, $\varphi_n := @_i\Diamond\psi$ and i_m is the first nominal not in Σ^n or φ_n.
  4. $\Sigma^{n+1} = \Sigma^n \cup \{\varphi_n, @_i\psi\frac{@_i c_{m,a}}{x_a}\}$, if $\Sigma^n \cup \{\varphi_n\}$ is consistent, $\varphi_n := @_i\exists x_a\psi$ and $c_{m,a}$ is the first constant of type a not in Σ^n or φ_n.

Now, let $\Sigma^\omega = \bigcup_{n\in\omega}\Sigma^n$. Σ^ω is named, $\Diamond$-saturated and $\exists$-saturated. We only need to prove that it is maximal consistent. This will follow easily once we prove that each Σ^n is consistent, which we shall do by induction.

For the base case, suppose Σ^0 is inconsistent. Hence $\Sigma \cup \{i_0\} \vdash \bot$, hence $\Sigma \vdash i_0 \to \bot$ and hence by Arrow Name (Claim 61) $\Sigma \vdash \bot$, which is impossible.

Now assume as inductive hypothesis that Σ^n is consistent. Now, Σ^{n+1} has only four possible forms:

1. $\Sigma^{n+1} = \Sigma^n$ is consistent by the induction hypothesis.
2. $\Sigma^{n+1} = \Sigma^n \cup \{\varphi_n\}$ is consistent by construction.
3. So suppose $\Sigma^{n+1} = \Sigma^n \cup \{\varphi_n, @_i\Diamond i_m, @_{i_m}\psi\}$, where $\varphi_n := @_i\Diamond\psi$ and i_m is the first new nominal that does not occur in Σ^n or φ_n. By construction, $\Sigma^n \cup \{\varphi_n\}$ is consistent. Suppose that $\Sigma^n \cup \{\varphi_n, @_i\Diamond i_m, @_{i_m}\psi\} \vdash \bot$. Then, $\Sigma^n \cup \{\varphi_n\} \vdash @_i\Diamond i_m \wedge @_{i_m}\psi \to \bot$, hence $\Sigma^n \cup \{\varphi_n\} \vdash @_i\Diamond\psi \to \bot$, by using Paste$_\Diamond$ (Claim 62) and the fact that $i_m \neq i$ and i_m does not occur in ψ or $\bot$. Thus $\Sigma^n \cup \{\varphi_n\} \vdash \bot$, which is impossible.
4. Lastly, suppose $\Sigma^n \cup \{\varphi_n, @_i\psi\frac{@_i c_{m,a}}{x_a}\}$, where $\varphi_n := @_i\exists x_a\psi$ and $c_{m,a}$ is the first new constant of type a that does not occur in Σ^n or φ_n. By construction, $\Sigma^n \cup \{\varphi_n\}$ is consistent. Suppose $\Sigma^n \cup \{\varphi_n, @_i\psi\frac{@_i c_{m,a}}{x_a}\} \vdash \bot$. Then $\Sigma^n \cup \{\varphi_n, \exists x_a @_i\psi\} \vdash \bot$, by Claim 44. Thus $\Sigma^n \cup \{\varphi_n, @_i\exists x_a\psi\} \vdash \bot$, using Claim 45. Thus $\Sigma^n \cup \{\varphi_n\} \vdash \bot$, which is impossible.

We conclude that Σ^ω is consistent. Maximality is clear by construction. $\dashv$

Maximal consistent sets which are named, $\Diamond$-saturated and $\exists$-saturated contain lots of useful information. We will be particularly interested in what they tell us about the equivalence of *rigid* expressions:

Definition 27 *Let Δ be a named, $\Diamond$-saturated and $\exists$-saturated maximal consistent set. Then:*

- *For all α_a, $\beta_a \in \mathsf{RIGIDS}_a$: $\alpha_a \approx_\Delta \beta_a$ iff $\alpha_a = \beta_a \in \Delta$, for every $a \in \mathsf{TYPES} - \{t\}$. The* ***rigidity equivalence class*** *of α_a, notation $[\alpha_a]_\Delta$, is the set $\{\beta_a \mid \alpha_a \approx_\Delta \beta_a\}$.*
- *For φ, $\psi \in \mathsf{ME}_t : \varphi \approx_\Delta \psi$ iff $\varphi = \psi \in \Delta$.The* ***truth equivalence class*** *of φ, notation $[\varphi]_\Delta$, is the set $\{\psi \mid \varphi \approx_\Delta \psi\}$.*

When Δ is clear from context we will usually write $\approx$ instead of $\approx_\Delta$, and $[\alpha]$ instead of $[\alpha]_\Delta$. It is straightforward to check that both rigidity equivalence and truth equivalence are equivalence relations.

And now for a key result: in named and saturated maximal consistent sets, these equivalence classes are all represented by *constants*. This result will simplify the construction of the type hierarchy in the following session; its proof makes use of the Hybrid Barcan axioms.

Theorem 28 (Rigid Representatives) *Let Δ be a maximal consistent set which is named, $\Diamond$-saturated and $\exists$-saturated.*

1. *Let $i \in \mathsf{NOM}$ and $\alpha \in \mathsf{ME}_t$. Then $[\alpha] = [@_i\bot]$ or $[\alpha] = [@_i\top]$.*
2. *Let $i \in \mathsf{NOM}$ and $\alpha_a \in \mathsf{RIGIDS}_a$, such that $a \neq t$. Then there is a constant $c_a \in \mathsf{CON}$ such that $[\alpha_a] = [@_i c_a]$.*

Proof The proof is by induction on type structure.

[Type t**]** Let i be any nominal and $\alpha \in \mathsf{ME}_t$. Assume that $\alpha \in \Delta$. But $\alpha \to (\alpha = \top) \in \Delta$, by propositional logic. Thus $\alpha = \top \in \Delta$, and $\alpha = @_i\top \in \Delta$, by Axiom 8c and maximal consistency. Hence $[\alpha] = [@_i\top]$. On the other hand, if we assume that $\alpha \notin \Delta$, both $\neg\alpha$ and $\neg\alpha \to (\alpha = \bot) \in \Delta$, and similar reasoning lets us conclude that $[\alpha] = [@_i\bot]$. A further remark may be helpful. Since for any nominal i we have that $\vdash @_i\bot = \bot$ and $\vdash @_i\top = \top$, by Axiom 8c $[@_i\bot] = [\bot]$ and $[@_i\top] = [\top]$. That is, the choice of the nominal i is irrelevant; there really are only two truth equivalence classes.

[Type e**]** Let $\alpha_e \in \mathsf{RIGIDS}_e$. By Axiom 4a and Rule 2b, $\vdash @_i(\alpha_e = \alpha_e)$ which can be rewritten as $\vdash @_i(y_e = \alpha_e)\frac{\alpha_e}{y_e}$. By Claim 39 and Modus Ponens $\vdash \exists y_e @_i(y_e = \alpha_e)$. By Existential Hybrid Barcan (Claim 42), we have $\vdash @_i\exists y_e(y_e = \alpha_e) \leftrightarrow \exists y_e @_i(y_e = \alpha_e)$. Therefore, $@_i\exists y_e(y_e = \alpha_e) \in \Delta$ by maximal consistency. By $\exists$-saturation, there exists a constant $c_e \in \mathsf{CON}$ such that $@_i(@_i c_e = \alpha_e) \in \Delta$. Thus $@_i c_e = \alpha_e \in \Delta$, by Axioms 8a and 8c, and so $[\alpha_e] = [@_i c_e]$.

[Inductive step] Let $\gamma_{\langle b,a\rangle} \in \mathsf{RIGIDS}_{\langle b,a\rangle}$. By Rigid Comprehension (Claim 52) we have $\vdash \exists x_{\langle b,a\rangle} @_i \forall v_b(x_{\langle b,a\rangle} v_b = \gamma_{\langle b,a\rangle} v_b)$ with $x_{\langle b,a\rangle}$ and v_b not in $\gamma_{\langle b,a\rangle}$, and so this formula is in Δ by maximal consistency. Existential Hybrid Barcan gives us that $@_i \exists x_{\langle b,a\rangle} \forall v_b(x_{\langle b,a\rangle} v_b = \gamma_{\langle b,a\rangle} v_b) \in \Delta$, hence, as Δ is $\exists$-saturated, there is a constant $c_{\langle b,a\rangle} \in \mathsf{CON}$ such that $@_i \forall v_b(@_i c_{\langle b,a\rangle} v_b = \gamma_{\langle b,a\rangle} v_b) \in \Delta$ too.

Using Axioms 5a and 2b, and Rule 2b we have $@_i(@_i c_{\langle b,a\rangle} = \gamma_{\langle b,a\rangle}) \in \Delta$. By Axioms 8a and 6e $\vdash @_i(@_i c_{\langle b,a\rangle} = \gamma_{\langle b,a\rangle}) = (@_i c_{\langle b,a\rangle} = @_i \gamma_{\langle b,a\rangle})$ and thus $@_i c_{\langle b,a\rangle} = @_i \gamma_{\langle b,a\rangle} \in \Delta$. As $\gamma_{\langle b,a\rangle}$ is rigid we can use Axiom 8c to get $@_i c_{\langle b,a\rangle} = \gamma_{\langle b,a\rangle} \in \Delta$. Hence $[@_i c_{\langle b,a\rangle}] = [\gamma_{\langle b,a\rangle}]$. $\dashv$

5 Completeness of BHTT

We come to the heart of the proof: building generalized interpretations out of (named and saturated) maximal consistent sets. We shall do this in three steps.

Recall that a structure has the form $\mathcal{M} = \langle \mathcal{S}, \mathsf{F}\rangle$ consisting of a skeleton $\mathcal{S} = \langle \langle \mathsf{D}_a\rangle_{a \in \mathsf{TYPES}}, W, R\rangle$ and a denotation function F. In the first step we define the type hierarchy $\langle \mathsf{D}_a\rangle_{a \in \mathsf{TYPES}}$. This is the most technical step, and this is where we make use of the Rigid Representatives Theorem. In the second step we define $\langle W, R\rangle$ and F. This part is straightforward: F is easy to define, and we use the standard hybrid construction of $\langle W, R\rangle$. In the third step we define the general interpretation $\langle \mathcal{M}, g\rangle$ we need, and show that it satisfies all the formulas in the maximal consistent set used to build it. Completeness follows.

Constructing the hierarchy

How does Henkin construct type hierarchies? On page 86 of *Completeness in the Theory of Types* he says this:

> *We now define by induction on α a frame of domains $\{\mathsf{D}_\alpha\}$ and simultaneously a one-one mapping Φ of equivalence classes onto the domains D_α such that $\Phi([A_\alpha])$ is in D_α.*

Our logic and notation are somewhat different, but the proof of following theorem is thoroughly Henkin in spirit.

Theorem 29 (Hierarchy Theorem) *Given a maximal consistent set Δ, which is named, $\Diamond$-saturated and $\exists$-saturated, there exists a family of domains $\langle \mathsf{D}_a\rangle_{a \in \mathsf{TYPES}}$ and a function Φ such that:*

1. *Φ is a bijection from BB (Building Blocks) to $\bigcup_{a \in \mathsf{TYPES}} \mathsf{D}_a$, where*

$$\mathsf{BB} = \bigcup_{a \in \mathsf{TYPES} \setminus \{t\}} \{[\alpha_a] \mid \alpha_a \in \mathsf{RIGIDS}_a\} \cup \{[\varphi] \mid \varphi \in \mathsf{ME}_t\}.$$

2. *$\mathsf{D}_t = \{\Phi([\varphi]) \mid \varphi \in \mathsf{ME}_t\}$ and $\mathsf{D}_a = \{\Phi([\alpha_a]) \mid \alpha_a \in \mathsf{RIGIDS}_a\}$ for $a \neq t$.*

Proof The proof is by induction on $a \in \mathsf{TYPES}$ by simultaneously defining the hierarchy $\langle \mathsf{D}_a \rangle_{a \in \mathsf{TYPES}}$ and the function Φ.

[Type t] We define D_t to be the two elements set $\mathsf{D}_t = \{[@_i\bot], [@_i\top]\}$, and for every $\varphi \in \mathsf{ME}_t$ we define:

$$\Phi([\varphi]) = \begin{cases} [@_i\top] & \text{iff} \quad \varphi \in \Delta \\ [@_i\bot] & \text{iff} \quad \neg\varphi \in \Delta, \end{cases}$$

where the chosen nominal i is arbitrary. It is immediate by the first part of the Rigid Representatives Theorem that Φ is well-defined, one-to-one, and onto.

[Type e] We define $\mathsf{D}_e = \{[@_i c_e] \mid c_e \in \mathsf{CON}(\Delta), i \in \mathsf{NOM}\}$. We also define $\Phi([\alpha_e]) = [\alpha_e]$ for every $\alpha_e \in \mathsf{RIGIDS}_e$. Clearly Φ is well-defined, one-to-one and onto. And by the second part of Rigid Representatives (Theorem 28) we have that $\mathsf{D}_e = \{[\alpha_e] \mid \alpha_e \in \mathsf{RIGIDS}_e\}$, and thus $\mathsf{D}_e = \{\Phi([\alpha_e]) \mid \alpha_e \in \mathsf{RIGIDS}_e\}$.

[Inductive step] Assuming that the theorem holds for $a, b \in \mathsf{TYPES}$ with $b \neq t$, we now prove it for $\langle b, a\rangle \in \mathsf{TYPES}$.

For every $\gamma_{\langle b,a\rangle}$ that is an element of $\mathsf{RIGIDS}_{\langle b,a\rangle}$ we define the value of Φ for the argument $[\gamma_{\langle b,a\rangle}]$ as follows: $\Phi([\gamma_{\langle b,a\rangle}])$ is itself a function with domain D_b and range D_a whose value for any element $\Phi([\beta_b])$ of D_b is the element $\Phi([\gamma_{\langle b,a\rangle}\beta_b])$ of D_a. That is, we have:

$$\Phi([\gamma_{\langle b,a\rangle}])(\Phi([\beta_b])) = \Phi([\gamma_{\langle b,a\rangle}\beta_b]).$$

It is easy to see that Φ does not depend on the particular representative chosen. For suppose $\gamma'_{\langle b,a\rangle} \approx \gamma_{\langle b,a\rangle}$ and $\beta'_b \approx \beta_b$. Thus $\Delta \vdash \gamma'_{\langle b,a\rangle} = \gamma_{\langle b,a\rangle}$ and $\Delta \vdash \beta'_b = \beta_b$, and by Claim 51 we have that $\Delta \vdash \gamma'_{\langle b,a\rangle}\beta'_b = \gamma_{\langle b,a\rangle}\beta_b$. This means that $\gamma_{\langle b,a\rangle}\beta_b \approx \gamma'_{\langle b,a\rangle}\beta'_b$ and so $[\gamma_{\langle b,a\rangle}\beta_b] = [\gamma'_{\langle b,a\rangle}\beta'_b]$. So Φ is well defined.

Next we define: $\mathsf{D}_{\langle b,a\rangle} = \{\Phi([\gamma_{\langle b,a\rangle}]) \mid \gamma_{\langle b,a\rangle} \in \mathsf{RIGIDS}_{\langle b,a\rangle}\}$. Now, $\mathsf{D}_{\langle b,a\rangle}$ clearly has the form we require, and Φ is obviously a mapping onto $\mathsf{D}_{\langle b,a\rangle}$, but is it one-to-one? To see that it is, reason as follows. Let $\Phi([\gamma'_{\langle b,a\rangle}]) = \Phi([\gamma_{\langle b,a\rangle}])$. We need to show that $[\gamma'_{\langle b,a\rangle}] = [\gamma_{\langle b,a\rangle}]$. As they are equal, the functions $\Phi([\gamma'_{\langle b,a\rangle}])$ and $\Phi([\gamma_{\langle b,a\rangle}])$ give the same value for any argument $\Phi([\beta_b]) \in \mathsf{D}_b$, for $\beta_b \in \mathsf{RIGIDS}_b$. By the second part of the Rigid Representatives Theorem each member of D_b is of the form $\Phi([@_i c_b])$ with $c_b \in \mathsf{CON}(\Delta)$, so we can write

$$\Phi([\gamma_{\langle b,a\rangle}])(\Phi([@_i c_b])) = \Phi([\gamma'_{\langle b,a\rangle}])(\Phi([@_i c_b]))$$

for all $c_b \in \mathsf{CON}(\Delta)$. But $\Phi([\gamma_{\langle b,a\rangle}@_i c_b]) = \Phi([\gamma'_{\langle b,a\rangle}@_i c_b])$ because, by the induction hypothesis for elements of type a, the function Φ is one-to-one, and so we have that $[\gamma_{\langle b,a\rangle}@_i c_b] = [\gamma'_{\langle b,a\rangle}@_i c_b]$. Therefore $\Delta \vdash \gamma_{\langle b,a\rangle}@_i c_b = \gamma'_{\langle b,a\rangle}@_i c_b$ for all $c_b \in \mathsf{CON}(\Delta)$. Thus $\Delta \vdash @_i\forall v_b(\gamma_{\langle b,a\rangle}v_b = \gamma'_{\langle b,a\rangle}v_b)$ for v_b not free in $\gamma_{\langle b,a\rangle}$ and $\gamma'_{\langle b,a\rangle}$, by Lemma 25. Now, by Axiom 5a we have that:

$$\vdash \forall v_b(\gamma_{\langle b,a\rangle}v_b = \gamma'_{\langle b,a\rangle}v_b) \to \gamma_{\langle b,a\rangle} = \gamma'_{\langle b,a\rangle}.$$

Hence:

$$\vdash @_i \forall v_b(\gamma_{\langle b,a\rangle} v_b = \gamma'_{\langle b,a\rangle} v_b) \to @_i(\gamma_{\langle b,a\rangle} = \gamma'_{\langle b,a\rangle})$$

by Rule 2b and Axiom 2b. So $\Delta \vdash \gamma_{\langle b,a\rangle} = \gamma'_{\langle b,a\rangle}$ by Axiom 8c. And this means we have that $[\gamma'_{\langle b,a\rangle}] = [\gamma_{\langle b,a\rangle}]$ which means that Φ is one-to-one. $\dashv$

Corollary 30 $\langle \mathsf{D}_a \rangle_{a \in \mathsf{TYPES}}$ *is a type hierarchy.*

Proof By definition, D_t is a two element set. Also, $\mathsf{D}_e \neq \varnothing$, because for every $v_e \in \mathsf{VAR}_e$ the formula $@_i \exists v_e \top \in \Delta$ (Claim 40 and Rule 2b). Hence, as Δ is $\exists$-saturated, there exists a constant $c_e \in \mathsf{CON}(\Delta)$ such that the formula $@_i \top \frac{@_i c_e}{v_e} \in \Delta$ for every variable from the infinite set VAR_e. Thus $[@_i c_e] \in \mathsf{D}_e$. Finally, $\mathsf{D}_{\langle b,a\rangle} \subseteq \mathsf{D}_a^{\mathsf{D}_b}$ as $\mathsf{D}_{\langle b,a\rangle} = \{\Phi([\gamma_{\langle b,a\rangle}]) \mid \gamma_{\langle b,a\rangle} \in \mathsf{RIGIDS}_{\langle b,a\rangle}\}$ and each $\Phi([\gamma_{\langle b,a\rangle}])$ is a function from D_b to D_a. $\dashv$

Defining the structure

That was the tricky part. But with the hierarchy now defined it is straightforward to complete the definition of the structure we require by defining $\langle W, R\rangle$ and F. To this end we first define an equivalence relation between nominals.

Definition 31 *Let Δ be a maximal consistent set. Define, for $i, j \in \mathsf{NOM}$, $i \approx' j$ iff $@_i j \in \Delta$. For $i \in \mathsf{NOM}$, $[i] = \{j \in \mathsf{NOM} : i \approx' j\}$. It is easy to show that $\approx'$ is an equivalence relation on NOM.*

Definition 32 (Basic Hybrid Henkin Structures) *Let Δ be a maximal consistent set which is named, $\Diamond$-saturated and $\exists$-saturated. The* ***Basic Hybrid Henkin Structure*** *$\mathcal{M} = \langle \mathcal{S}, \mathsf{F}\rangle$* ***over*** *$\Delta$ is made up of:*

1. *The skeleton $\mathcal{S} = \langle \langle \mathsf{D}_a \rangle_{a \in \mathsf{TYPES}}, W, R\rangle$, defined by:*
   - *(a) $\langle \mathsf{D}_a \rangle_{a \in \mathsf{TYPES}}$, as given by the Hierarchy Theorem,*
   - *(b) $W = \{[i] \mid i$ is a nominal$\}$,*
   - *(c) $R = \{\langle [i], [j]\rangle \mid @_i \Diamond j \in \Delta\}$.*
2. *F is a function with domain $\mathsf{NOM} \cup \mathsf{CON}$, defined by:*
   - *(a) For $c_{n,a} \in \mathsf{CON}$, $\mathsf{F}(c_{n,a})$ is a function from W to D_a, such that $\mathsf{F}(c_{n,a})([i]) = \Phi([@_i c_{n,a}])$.*
   - *(b) For $i \in \mathsf{NOM}$, $\mathsf{F}(i)$ is a function from W to $\mathsf{D}_t = \{[@_i \top], [@_i \bot]\}$, such that $(\mathsf{F}(i))([j]) = [@_i \top]$ iff $i \in [j]$.*

The set Δ over which a basic hybrid Henkin structure is built is usually clear from context, so often we don't mention it.

Lemma 33 *Any basic hybrid Henkin structure is a well-defined structure.*

Proof We already know that $\mathcal{S}$ is well-defined, for $\langle \mathsf{D}_a \rangle_{a \in \mathsf{TYPES}}$ is a type hierarchy by Corollary 30. The equivalence relation on NOM is easily seen to be well defined, so it only remains to show that R and F are too. For R we have:

1. $i \approx' i'$ and $\langle [i], [j] \rangle \in R$ implies $\langle [i'], [j] \rangle \in R$. We have $\vdash @_i i' \to (@_i \Diamond j \to @_{i'} \Diamond j)$ by Claim 58. Thus $@_i i' \in \Delta$ and $@_i \Diamond j \in \Delta$ implies $@_{i'} \Diamond j \in \Delta$.
2. $j \approx' j'$ and $\langle [i], [j] \rangle \in R$ implies $\langle [i], [j'] \rangle \in R$. We have $\vdash @_i \Diamond j \wedge @_j j' \to @_i \Diamond j'$ by Claim 57 . Thus $@_j j' \in \Delta$ and $@_i \Diamond j \in \Delta$ implies $@_i \Diamond j' \in \Delta$.

Moreover, F is well defined. For if $i \approx' j$ then $(\mathsf{F}(c_{n,a}))([i]) = (\mathsf{F}(c_{n,a}))([j])$ because by Claim 55 we have that $\vdash @_i j \to (@_i c_{n,a} = @_j c_{n,a})$. $\dashv$

General Interpretation and Completeness

One last detail remains: defining our variable assignment. We do so as follows:

Definition 34 *An assignment on a basic hybrid Henkin structure $\mathcal{M}$ is a function mapping each variable v_a to some element of $\mathsf{D}_a = \{\Phi([\alpha_a]) \mid \alpha_a \in \mathsf{RIGIDS}_a\}$. The* ***Hybrid Henkin Assignment*** *on $\mathcal{M}$ is the function g defined as follows. For every $v_a \in \mathsf{VAR}_a$:*

$$g(v_a) = \Phi([v_a]).$$

Note that since all variables v_a are rigid, $@_i v_a = v_a$ for all $i \in \mathsf{NOM}$, and so we could also have defined $g(v_a)$ to be $\Phi([@_i v_a])$.

Theorem 35 *Let $\mathcal{M}$ be a basic hybrid Henkin structure and g its hybrid Henkin assignment. For all meaningful expressions β_b and for all $i \in \mathsf{NOM}$ we have:*

$$[[\beta_b]]^{\mathcal{M},[i],g} = \Phi([@_i \beta_b]).$$

Proof The proof is a conceptually clear but somewhat finicky induction on the formation of expressions. We give a selection of cases.

[Case $j \in \mathsf{NOM}$] $[[j]]^{\mathcal{M},[i],g} = (\mathsf{F}(j))([i])$. By Definition 32, $(\mathsf{F}(j))([i]) = [@_j \top]$ iff $j \in [i]$, which in turn is equivalent to $@_i j \in \Delta$. Therefore, $[[j]]^{\mathcal{M},[i],g} = \Phi([@_i j])$.

[Case $c_b \in \mathsf{CON}$] $[[c_b]]^{\mathcal{M},[i],g} = (\mathsf{F}(c_b))([i]) = \Phi([@_i c_b])$, by definition.

[Case $v_b \in \mathsf{VAR}$] $[[v_b]]^{\mathcal{M},[i],g} = g(v_b) = \Phi([@_i v_b])$, by the definition of g.

[Case $\neg\varphi$] $[[\neg\varphi]]^{\mathcal{M},[i],g} = [@_i \top]$ iff $\Phi([@_i \varphi]) = [@_i \bot]$ (by the induction hypothesis) iff $\Phi([@_i \neg\varphi]) = [@_i \top]$ by Axiom 6a.

[Case $\varphi \wedge \psi$] $[[\varphi \wedge \psi]]^{\mathcal{M},[i],g} = [@_i \top]$ iff $[[\varphi]]^{\mathcal{M},[i],g} = [@_i \top]$ and $[[\psi]]^{\mathcal{M},[i],g} = [@_i \top]$ iff $\Phi([@_i \varphi]) = [@_i \top]$ and $\Phi([@_i \psi]) = [@_i \top]$ (by the induction hypothesis) iff $@_i \varphi \in \Delta$ and $@_i \psi \in \Delta$ iff $@_i \varphi \wedge @_i \psi \in \Delta$ iff $\Phi([@_i \varphi \wedge @_i \psi]) = [@_i \top]$ iff $\Phi([@_i (\varphi \wedge \psi)]) = [@_i \top]$, by Claim 59.

[Case $\alpha_{\langle c,b \rangle} \gamma_c$] By definition, $[[\alpha_{\langle c,b \rangle} \gamma_c]]^{\mathcal{M},[i],g} = [[\alpha_{\langle c,b \rangle}]]^{\mathcal{M},[i],g}([[\gamma_c]]^{\mathcal{M},[i],g})$. By the induction hypothesis we have that $[[\alpha_{\langle c,b \rangle}]]^{\mathcal{M},[i],g} = \Phi([@_i \alpha_{\langle c,b \rangle}])$ and that $[[\gamma_c]]^{\mathcal{M},[i],g} = \Phi([@_i \gamma_c])$. This means:

$$[[\alpha_{\langle c,b \rangle}]]^{\mathcal{M},[i],g}([[\gamma_c]]^{\mathcal{M},[i],g}) = \Phi([@_i \alpha_{\langle c,b \rangle}])(\Phi([@_i \gamma_c])).$$

Using the definition of function Φ for type $\langle c, b\rangle$, the right hand side of this equality is $\Phi([@_i\alpha_{\langle c,b\rangle}@_i\gamma_c])$. By Axiom 8b and properties of maximal consistent sets, $[@_i(\alpha_{\langle c,b\rangle}\gamma_c)] = [@_i\alpha_{\langle c,b\rangle}@_i\gamma_c]$. Thus $[[\alpha_{\langle c,b\rangle}\gamma_c]]^{\mathcal{M},[i],g} = \Phi[@_i(\alpha_{\langle c,b\rangle}\gamma_c)]$.

[Case $@_j\beta_b$**]** $[[@_j\beta_b]]^{\mathcal{M},[i],g} = [[\beta_b]]^{\mathcal{M},[k],g}$, where $[k]$ is the unique element such that $(\mathsf{F}(j))([k]) = [@_i\top]$ — and if this holds it means that $j \in [k]$ and $@_jk \in \Delta$. Thus $[j] = [k]$. Hence $[[\beta_b]]^{\mathcal{M},[k],g} = [[\beta_b]]^{\mathcal{M},[j],g} = \Phi[@j\beta_b]$ using the induction hypothesis for β_b. Using Axiom 6e and properties of maximal consistent sets, $@_i@_j\beta_b = @_j\beta_b \in \Delta$ and so $[@_i@_j\beta_b] = [@_j\beta_b]$ and $\Phi[@_j\beta_b] = \Phi[@_i@_j\beta_b]$. Therefore $[[@_j\beta_b]]^{\mathcal{M},[i],g} = \Phi[@_i@_j\beta_b]$.

[Case $\lambda u_c\alpha_a$**]** We want to prove that $[[\lambda u_c\alpha_a]]^{\mathcal{M},[i],g} = \Phi([@_i(\lambda u_c\alpha_a)])$. On the one hand, $[[\lambda u_c\alpha_a]]^{\mathcal{M},[i],g}$ is the function $h : \mathsf{D}_c \longrightarrow \mathsf{D}_a$, that for every element $\theta \in \mathsf{D}_c$ gives the value $[[\alpha_a]]^{\mathcal{M},[i],g^{\theta}_{u_c}}$ in D_a. As all the elements of D_c are of the form $\Phi([\beta_c])$ with β_c rigid, we can define the function by

$$h(\Phi([\beta_c])) = [[\alpha_a]]^{\mathcal{M},[i],g^{\Phi([\beta_c])}_{u_c}}$$

and then observe that

$$[[\alpha_a]]^{\mathcal{M},[i],g^{\Phi([\beta_c])}_{u_c}} = [[\alpha_a]]^{\mathcal{M},[i],g^{[[\beta_c]]^{\mathcal{M},[i],g}}_{u_c}}$$

because $[[\beta_c]]^{\mathcal{M},[i],g} = \Phi([@_i\beta_c])$ by the induction hypothesis for type c. Moreover, we also know that:

$$[[\alpha_a]]^{\mathcal{M},[i],g^{[[\beta_c]]^{\mathcal{M},[i],g}}_{u_c}} = [[\alpha_a\frac{\beta_c}{u_c}]]^{\mathcal{M},[i],g}$$

using the fact that β_c is rigid and Lemma 13.

On the other hand, $\Phi([@_i(\lambda u_c\alpha_a)])$ is the function $h' : \mathsf{D}_c \longrightarrow \mathsf{D}_a$ that, for $\Phi([\beta_c]) \in \mathsf{D}_c$ with β_c rigid, returns

$$h'(\Phi[\beta_c]) = \Phi([@_i(\lambda u_c\alpha_a)])(\Phi([\beta_c])) = \Phi([@_i((\lambda u_c\alpha_a)\beta_c)]).$$

By Axiom 5b for β_c we have $\vdash (\lambda u_c\alpha_a)\beta_c = \alpha_a\frac{\beta_c}{u_c}$, and thus $@_i((\lambda u_c\alpha_a)\beta_c) = @_i(\alpha_a\frac{\beta_c}{u_c}) \in \Delta$, by Axioms 8a and 2b. Hence $[@_i((\lambda u_c\alpha_a)\beta_c)] = [@_i(\alpha_a\frac{\beta_c}{u_c})]$ and so we have that $\Phi([@_i((\lambda u_c\alpha_a)\beta_c)]) = \Phi([@_i(\alpha_a\frac{\beta_c}{u_c})])$. This in turn means that $h(\Phi([\beta_c])) = h'(\Phi[\beta_c])$ as $[[\alpha_a\frac{\beta_c}{u_c}]]^{\mathcal{M},[i],g} = \Phi([@_i(\alpha_a\frac{\beta_c}{u_c})])$ by the induction hypothesis for type a. Thus $h = [[\lambda u_c\alpha_a]]^{\mathcal{M},[i],g} = \Phi([@_i(\lambda u_c\alpha_a)]) = h'$.

The cases we have given illustrate the kind of argumentation required. The omitted proofs for $=$, $\Box$ are straightforward, but the argument for $\forall$ (like the step for λ given above) probably requires a little more patience and a taste for superscripts. $\dashv$

Corollary 36 *A pair* $\langle\mathcal{M}, g\rangle$ *where* $\mathcal{M}$ *is a basic hybrid Henkin structure and* g *is its Henkin assignment is a general interpretation.*

Proof We have just proved that every expression has an interpretation in the corresponding domain of the hierarchy, but this is precisely what we require of general interpretations. ⊣

Theorem 37 (Henkin's theorem) *Every consistent set of formulas has a general interpretation that satisfies it.*

Proof Let Γ be a consistent set of formulas. By Lemma 26, there exists a maximal consistent extension Δ of Γ which is named, $\Diamond$-saturated and $\exists$-saturated. As Δ is named, there exists a nominal k in Δ. By Theorem 35 and Corollary 36 there is a general interpretation $\langle \mathcal{M}, g\rangle$ such that, for all $\beta_t \in \mathsf{ME}_t$ the following holds:

$$[[\beta_t]]^{\mathcal{M},[k],g} = \Phi([@_k\beta_t]).$$

Let $\varphi_t \in \Gamma$. Therefore $@_k\varphi_t \in \Delta$. Therefore $[[\varphi_t]]^{\mathcal{M},[k],g} = [@_k\top]$ because $\Phi([@_k\varphi_t]) = [@_k\top]$, since $@_k\varphi_t \in \Delta$. ⊣

Theorem 38 (Completeness) *For all $\Gamma \subseteq \mathsf{ME}_t$ and $\varphi \in \mathsf{ME}_t$, the following holds: $\Gamma \models \varphi$ implies $\Gamma \vdash \varphi$*

Proof Standard. ⊣

6 Not quite so basic

We have proved the basic completeness result for BHTT; as we shall now show, we have actually done rather more. Work on propositional and first-order hybrid logic has shown that constructing models out of equivalence classes of nominals has an important advantage: it more-or-less automatically leads to completeness proofs for stronger logics and languages. We retain these advantages even in the setting of higher-order logic. As we shall see, our basic result for BHTT yields further completeness results when we demand that the accessibility relation R have special properties (for example, reflexivity, irreflexivity, symmetry or antisymmetry), or when we enrich BHTT with various useful modalities (such as the universal modality E, the difference operator D, and the Priorean operator pairs F and P).

Additional conditions on R

In BHTT we have a single modality $\Box$ (at least, if we ignore the $@_i$ operators) and no constraints on the relation R used in its interpretation. Thus the completeness result we have proved is a basic (or minimal, or **K**) result for a unimodal language. We will shortly discuss what is involved in adding additional modalities, but let's first ask: if we are interested in imposing restrictions on R (that it be transitive, irreflexive, and trichotomous, for example) how should we proceed?

Hybrid logic provides some general answers. The simplest is this: if we can find *pure formulas* that define the required conditions on R, then adding these formulas as axioms results in a logic complete with respect to the desired conditions. Now, for present purposes, a pure formula is simply an expression of type t that does not contain any non-logical constants or variables; that is, it is built up solely from nominals. And for the example just mentioned (transitivity, irreflexivity, and trichotomy) an easy axiomatization is available:

$$\Diamond\Diamond i \rightarrow \Diamond i \qquad @_i \neg \Diamond i \qquad @_i \Diamond j \vee @_i j \vee @_j \Diamond i.$$

It should be clear that the first formula is valid iff R is transitive, the second iff R is irreflexive, the third iff R is trichotomous (that is, given two worlds, either the first can access the second, or they are identical, or the second can access the first). It follows by standard hybrid logical results that adding these three formulas to our axiomatization of BHTT results in a system that is complete with respect to the class of models whose accessibility relation R is a strict linear order.

The pure axioms lead to general completeness results in this way is a reflection of simple model theoretic facts about hybrid logic; see the section on Model Theory in the survey by Areces and Ten Cate [3] for a clear discussion. A deeper and more difficult question is: what conditions on R can be handled in this way? For some answers, see Ten Cate [13].

Additional modalities

We defined BHTT as a system containing only a single modality $\Box$ (and its defined dual $\Diamond$). For many applications it is common to have a finite collection of box modalities $[\beta]$ and their associated diamonds $\langle\beta\rangle$, where β ranges over the elements of some suitable index set B. Now, if BHTT is extended in the obvious way with such modalities (that is, if each modality $[\beta]$ is interpreted with respect to a binary relation R_β using the familiar Kripke satisfaction definition) the completeness result we have proved extends in the obvious way. Working in the richer language simply means that we need a β-indexed collection of axioms and rules of proof. For each modality $[\beta]$ we have the β-distributivity axioms and the β-back axioms

$$[\beta](\varphi \rightarrow \psi) \rightarrow ([\beta]\varphi \rightarrow [\beta]\psi) \qquad \langle\beta\rangle @_i \varphi \rightarrow @_i \varphi,$$

the β-generalization rule (if $\vdash \varphi$ then $\vdash [\beta]\varphi$), and the β-bounded generalization rule (if $\vdash @_i\langle\beta\rangle j \rightarrow @_j \varphi$ and $j \neq i$ and j does not occur in φ, then $\vdash @_i[\beta]\varphi$). No new ideas are needed to extend our completeness result to such multimodal extensions. Moreover, as just discussed, with the help of pure formulas we can impose additional conditions on the various relations. For example, to axiomatize a multimodal extension that utilizes a symmetric relation R_a and an antisymmetric relation R_b simply add the axioms

$$@_i[a]\langle a\rangle i \quad \text{and} \quad @_i[b](\langle b\rangle i \rightarrow i).$$

Often multimodal extensions involve interactions between modalities, and where these interactions can be handled using pure axioms, completeness results are forthcoming. The classic example is Priorean tense logic.[5] Here we interpret two dual pairs of operators over the same binary relation $<$, the idea being that the F, G pair look forward along the relation, the P, H pair backwards:

$$[[F\varphi_t]]^{\mathcal{M},w,g} = T \text{ iff for some } v \in W \text{ such that } w < v,\ [[\varphi_t]]^{\mathcal{M},v,g} = T$$
$$[[G\varphi_t]]^{\mathcal{M},w,g} = T \text{ iff for all } v \in W \text{ such that } w < v,\ [[\varphi_t]]^{\mathcal{M},v,g} = T$$
$$[[P\varphi_t]]^{\mathcal{M},w,g} = T \text{ iff for some } v \in W \text{ such that } v < w,\ [[\varphi_t]]^{\mathcal{M},v,g} = T$$
$$[[H\varphi_t]]^{\mathcal{M},w,g} = T \text{ iff for all } v \in W \text{ such that } v < w,\ [[\varphi_t]]^{\mathcal{M},v,g} = T$$

Once again, adding such operators to BHTT and axiomatizing them is straightforward. We add the G-distribution and H-distribution axioms

$$G(\varphi \to \psi) \to (G\varphi \to G\psi) \qquad \text{and} \qquad H(\varphi \to \psi) \to (H\varphi \to H\psi),$$

the F-back and P-back axioms

$$F@_i\varphi \to @_i\varphi, \qquad \text{and} \qquad P@_i\varphi \to @_i\varphi,$$

and G- and H-generalization and bounded generalisation rules. But how do we capture the desired interaction between these modalities? By adding the following pure axiom:

$$@_iFj \leftrightarrow @_jPi.$$

It is easy to check that this forces F and P to look forwards and backwards, respectively, along the same relation.[6]

Two other modalities are worth mentioning; both have a long association with hybrid logic: the universal modality E (see Goranko and Passy [20]) and the difference operator D (see De Rijke [33]). Informally, $E\varphi$ says that at *some* point in a model φ is true, while its dual A lets us insist that φ is true at *all* points in a model. And $D\varphi$ says that at a *different* point of the model φ is true, while its dual form $\overline{D}$ insists that φ is true *everywhere else.* More precisely:

$$[[E\varphi_t]]^{\mathcal{M},w,g} = T \text{ iff for some } v \in W,\ [[\varphi_t]]^{\mathcal{M},v,g} = T$$
$$[[A\varphi_t]]^{\mathcal{M},w,g} = T \text{ iff for all } v \in W,\ [[\varphi_t]]^{\mathcal{M},v,g} = T$$
$$[[D\varphi_t]]^{\mathcal{M},w,g} = T \text{ iff for some } v \neq w \in W,\ [[\varphi_t]]^{\mathcal{M},v,g} = T$$
$$[[\overline{D}\varphi_t]]^{\mathcal{M},w,g} = T \text{ iff for all } v \neq w \in W,\ [[\varphi_t]]^{\mathcal{M},v,g} = T$$

Once again, with the help of pure axioms, we can add either of these operators to BHTT. Depending on which of the two operators we want, we add the

[5] For some background discussion of higher-order hybrid tense logic, see [1] and [2].

[6] This in turn permits a further simplification: the Bounded Generalization rule is derivable for pairs of converse operators like F and P. See [8, 9] for further discussion and details.

appropriate distribution axioms, back axioms, and generalization and bounded generalization rules. And here's how we capture their characteristic behaviors:

$$Ei \qquad\qquad\qquad Di \leftrightarrow \neg i.$$

These pure axioms really do accomplish what is required. If R^U is the binary relation interpreting the universal modality, then we want it to have the property that $\forall w \forall w' R^U ww'$. It is easy to see that Ei defines this property. And if R^D is the binary relation interpreting the difference modality, then we want it to have the property that $\forall w \forall w'(R^D ww' \leftrightarrow w \neq w')$. Clearly $Di \leftrightarrow \neg i$ imposes this condition on R^D.

More can be achieved stronger methods. For example, Goranko [19] uses the notion of *local definability* to give a simple hybrid axiomatization of the Until operator, while Blackburn and ten Cate [8] make use of *existential saturation rules* to axiomatize frame classes for which no pure axiom exists. The details of these refinements are irrelevant here; what matters is the basic point. Theorem 38, our basic completeness result, is strong. Using standard hybrid-logical methods, it straightforwardly gives rise to further completeness results.

7 Concluding remarks

In this paper we defined a hybrid type theory called BHTT. We kept it as simple as possible: we used only e and t types together with the most basic hybrid apparatus, nominals and @. We wanted to see whether the Henkin-style completeness techniques used in propositional and first-order hybrid logic extended straightforwardly to higher-order hybrid logic, and they do — at least, if $@_i$ is used as a rigidifying operator for *all* types.

We have sketched how our results extend to richer extensions of the basic system, but that is hardly the end of the story. Many other questions beckon, and three particularly interest us. The first is to adapt BHTT to deal with variable domain semantics. In the setting of first-order logic, shifting between constant and variable domains is relatively straightforward (see Chapter 4 of Fitting and Mendelsohn [16]) but in higher-order settings we work with function hierarchies, not merely individuals, and here the choices are not so clearcut; further experimentation is called for. Secondly, we would like to experiment with a Fitting-style intensional semantics, thus avoiding the restriction to rigid terms. Some authors view rigidity restrictions as unnatural. We don't agree with such sentiments — but Fitting's approach is intriguing and we'd like to explore it. Thirdly, we intend to add the hybrid $\downarrow$ operator (see [19, 5, 3]) to the system. Although we have not used an intensional type s, BHTT is well attuned to the structure of $\langle W, R\rangle$ thanks to the nominals and $@_i$ operators. Adding $\downarrow$, which will let us bind nominals to the world of evaluation on the fly, will further boost this attunement — with useful consequences (we hope) for both logical elegance and applications in natural language semantics.

Acknowledgements

The authors are grateful to the Spanish Ministry of Science and Innovation for funding the project *Nociones de Completud* (*Conceptions of Completeness*), Proyectos de Investigación Fundamental MICINN FFI2009-09345. Hosted by the Department of Philosophy, Logic, and Aesthetics, University of Salamanca, Spain, 2009–2011, it made research visits between the authors possible.

References

[1] Carlos Areces and Patrick Blackburn. Reichenbach, Prior and Montague: a semantic get-together. In *We Will Show Them: Essays in honour of Dov Gabbay on his 60th Birthday*, volume 1, pages 77–87. College Publications, 2005.

[2] Carlos Areces, Patrick Blackburn, Maria Manzano, and Antonia Huertas. Hybrid type theory: A quartet in four movements. *Principia: an International Journal of Epistemology*, 15(2):225–247, 2011.

[3] Carlos Areces and Balder ten Cate. Hybrid logic. In *Handbook of Modal Logic*, pages 821–868. Elsevier, 2007.

[4] Jon Barwise and Robin Cooper. Generalized quantifiers and natural language. *Linguistics and Philosophy*, 4:159–219, 1981.

[5] Patrick Blackburn. Representation, reasoning, and relational structures: a Hybrid Logic manifesto. *Logic Journal of the IGPL*, 8:339–365, 2000.

[6] Patrick Blackburn. Arthur Prior and hybrid logic. *Synthese*, 150:329–372, 2006.

[7] Patrick Blackburn and Maarten Marx. Tableaux for quantified hybrid logic. In U. Egly and C. Fernmüller, editors, *Automated Reasoning with Analytic Tableaux and Related Methods, International Conference, TABLEAUX 2002*, pages 38–52. Copenhagen, Denmark, 2002.

[8] Patrick Blackburn and Balder ten Cate. Pure extensions, proof rules, and hybrid axiomatics. *Studia Logica*, 84:277–322, 2006.

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

[10] Patrick Blackburn, Johan van Benthem, and Frank Wolter. *Handbook of Modal Logic*. Elsevier, 2007.

[11] Torben Brauner and Silvio Ghilardi. First-order modal logic. In *Handbook of Modal Logic*, pages 549–620. Elsevier, 2007.

[12] Robert Bull. An approach to tense logic. *Theoria*, 36:282–300, 1970.

[13] Balder ten Cate. *Model theory for extended modal languages*. PhD thesis, University of Amsterdam, 2005. ILLC Dissertation Series DS-2005-01.

[14] Alonzo Church. A formulation of the simple theory of types. *Journal of Symbolic Logic*, 5:56–68, 1940.

[15] Melvin Fitting. *Types, Tableaus, and Goedel's God.* Kluwer Academic Publishers, 2002.

[16] Melvin Fitting and Richard Mendelsohn. *First-Order Modal Logic.* Springer, 1998.

[17] Daniel Gallin. *Intensional and Higher-Order Modal Logic.* North-Holland, Amsterdam, 1975.

[18] George Gargov and Valentin Goranko. Modal logic with names. *Journal of Philosophical Logic*, 22:607–636, 1993.

[19] Valentin Goranko. Tense logic with reference pointers. *Journal of Logic, Language and Information*, 5(1):1–24, 1996.

[20] Valentin Goranko and Solomon Passy. Using the universal modality: Gains and questions. *Journal of Logic and Computation*, 2:5–30, 1992.

[21] Jeroen Groenendijk and Martin Stokhof. *Studies on the Semantics of Questions and the Pragmatics of Answers.* Doctoral dissertation, University of Amsterdam, 1984.

[22] Leon Henkin. The completeness of the first-order functional calculus. *The Journal of Symbolic Logic*, 14(3):159–166, 1949.

[23] Leon Henkin. Completeness in the theory of types. *The Journal of Symbolic Logic*, 15(2):81–91, 1950.

[24] Leon Henkin. The discovery of my completeness proofs. *The Bulletin of Symbolic Logic*, 2(2):127–158, 1996.

[25] George Hughes and Max Cresswell. *A New Introduction to Modal Logic.* Routledge, 1996.

[26] Mark Kaminski and Gert Smolka. Terminating tableau systems for hybrid logic with difference and converse. *Journal of Logic, Language and Information*, 18(4):437–464, 2009.

[27] Maria Manzano. *Extensions of First Order Logic.* Cambridge University Press, 1996.

[28] Richard Montague. The proper treatment of quantification in ordinary English. In J. Hintikka, J. Moravcsik, and P. Suppes, editors, *Approaches to Natural Language*, pages 221–242. Reidel, 1973. Reprinted in [34].

[29] Reinhard Muskens. Combining Montague Semantics and discourse representation. *Linguistics and Philosophy*, 19:143–186, 1996.

[30] Reinhard Muskens. Higher order modal logic. In *Handbook of Modal Logic*, pages 621–653. Elsevier, 2007.

[31] Arthur Prior. *Past, Present and Future.* Oxford University Press, 1967.

[32] Arthur Prior. *Papers on Time and Tense.* Oxford University Press, New edition, 2003. Edited by Hasle, Øhrstrom, Braüner, and Copeland.

[33] Maarten de Rijke. The modal logic of inequality. *Journal of Symbolic Logic*, 57:566–584, 1992.

[34] Richmond Thomason, editor. *Formal Philosophy. Selected Papers by Richard Montague.* Yale University Press, 1974.

8 Appendix: Theorems of BHTT

We list here the BHTT-theorems (and derived rules) required to push the completeness proof through. Only for the more interesting (or tricky) examples are deductions given. For a detailed discussion of axiomatic proofs in hybrid logic (which discusses the derived rules introduced below in depth) see Blackburn and Ten Cate [8].

Claim 39 $\vdash \varphi\frac{\beta_b}{x_b} \rightarrow \exists x_b \varphi$ *for* β_b *rigid.*

Claim 40 $\vdash \exists x_a \top$.

Claim 41 $\vdash @_i \varphi\frac{@_i\alpha_a}{x_a} \rightarrow @_i \exists x_a \varphi$.

Claim 42 (Existential Hybrid Barcan) $\vdash @_i \exists x_b \varphi \leftrightarrow \exists x_b @_i \varphi$.

Claim 43 *If* $\Gamma \vdash \varphi$ *then* $\Gamma \vdash \forall y_a \varphi$ *for* y_a *not free in* Γ.

Claim 44 *If* $\varphi\frac{@_i c_a}{x_a} \vdash \psi$ *then* $\exists x_a \varphi \vdash \psi$, *for* $@_i c_a$ *not in* φ, ψ.

Claim 45 *If* $\exists x_a @_i \varphi \vdash \psi$ *then* $@_i \exists x_a \varphi \vdash \psi$.

Claim 46 $\vdash \alpha_a = \beta_a \rightarrow (\gamma_t \frac{\alpha_a}{v_a} \leftrightarrow \gamma_t \frac{\beta_a}{v_a})$ *for* α_a *and* β_a *rigids.*

Claim 47 $\vdash \alpha_a = \beta_a \rightarrow \beta_a = \alpha_a$ *for* α_a *and* β_a *rigids.*

Claim 48 (Symmetry) $\vdash \forall x_a y_a (x_a = y_a \rightarrow y_a = x_a)$.

Claim 49 $\vdash \alpha_a = \beta_a \rightarrow (\beta_a = \gamma_a \rightarrow \alpha_a = \gamma_a)$ *for* α_a, β_a *and* γ_a *rigids.*

Claim 50 (Transitivity) $\vdash \forall x_a y_a z_a (x_a = y_a \rightarrow (y_a = z_a \rightarrow x_a = z_a))$.

Claim 51 $\vdash \gamma'_{\langle b,a\rangle} = \gamma_{\langle b,a\rangle} \rightarrow (\beta'_b = \beta_b \rightarrow \gamma'_{\langle b,a\rangle}\beta'_b = \gamma_{\langle b,a\rangle}\beta_b)$ *for* $\gamma'_{\langle b,a\rangle}$, $\gamma_{\langle b,a\rangle}$, β'_b *and* β_b *rigids.*

Claim 52 (Rigid Comprehension) $\vdash \exists x_{\langle b,a\rangle} @_i \forall y_b (x_{\langle b,a\rangle} y_b = \gamma_{\langle b,a\rangle} y_b)$ *for* $\gamma_{\langle b,a\rangle}$ *rigid, and* y_b *and* $x_{\langle b,a\rangle}$ *not in* $\gamma_{\langle b,a\rangle}$.

1. $\vdash (\lambda x_b(\gamma_{\langle b,a\rangle} x_b)) y_b = (\gamma_{\langle b,a\rangle} x_b)\frac{y_b}{x_b}$, for x_b, y_b not in $\gamma_{\langle b,a\rangle}$, by Axiom 5b
2. $\vdash (\lambda x_b(\gamma_{\langle b,a\rangle} x_b)) y_b = \gamma_{\langle b,a\rangle} y_b$
3. $\vdash @_i((\lambda x_b(\gamma_{\langle b,a\rangle} x_b)) y_b = \gamma_{\langle b,a\rangle} y_b)$, by Rule 2b
4. $\vdash \forall y_b @_i((\lambda x_b(\gamma_{\langle b,a\rangle} x_b)) y_b = \gamma_{\langle b,a\rangle} y_b)$, by Rule 2c
5. $\vdash \forall y_b @_i(x_{\langle b,a\rangle} y_b = \gamma_{\langle b,a\rangle} y_b)\frac{\lambda x_b(\gamma_{\langle b,a\rangle} x_b)}{x_b}$
6. $\vdash @_i \forall y_b(x_{\langle b,a\rangle} y_b = \gamma_{\langle b,a\rangle} y_b)\frac{\lambda x_b(\gamma_{\langle b,a\rangle} x_b)}{x_b}$, by Axiom 7a
7. $\vdash \exists x_{\langle b,a\rangle} @_i \forall y_b(x_{\langle b,a\rangle} y_b = \gamma_{\langle b,a\rangle} y_b)$, by Claim 39 as $\lambda x_b(\gamma_{\langle b,a\rangle} x_b)$ is rigid.

Claim 53 $\vdash @_i j \rightarrow @_j i$.

1. $j \rightarrow (i \leftrightarrow @_j i)$, by Axiom 6b

2. $@_i(j \to (i \leftrightarrow @_j i))$, by Rule 2b
3. $(@_i j \to (@_i i \leftrightarrow @_i @_j i))$, by Axiom 2b
4. $@_i @_j i \leftrightarrow @_j i$, by Axiom 6e
5. $@_i j \to (@_i i \leftrightarrow @_j i)$, from lines 3 and 4
6. $@_i i$, by Axiom 6d
7. $@_i j \to @_j i$, from lines 5 and 6

Claim 54 $\vdash @_i j \to (@_i\varphi \leftrightarrow @_j\varphi)$.

1. $\vdash j \to (\varphi \leftrightarrow @_j\varphi)$, by Axiom 6b
2. $\vdash @_i(j \to (\varphi \leftrightarrow @_j\varphi))$, by Rule 2b
3. $\vdash @_i j \to (@_i\varphi \leftrightarrow @_i @_j\varphi)$, by Axiom 2b
4. $\vdash @_i @_j\varphi \leftrightarrow @_j\varphi$, by Axiom 6e
5. $\vdash @_i j \to (@_i\varphi \leftrightarrow @_j\varphi)$, from previous lines

Claim 55 $\vdash @_j k \to @_j\beta_b = @_k\beta_b$.

1. $\vdash @_j k \to (@_j(\beta_b = @_j\beta_b) \leftrightarrow @_k(\beta_b = @_j\beta_b))$, by Claim 54
2. $\vdash @_j k \to (@_j\beta_b = @_j @_j\beta_b \leftrightarrow @_k\beta_b = @_k @_j\beta_b)$, by Axiom 8a
3. $\vdash @_j k \to (@_j\beta_b = @_j\beta_b \leftrightarrow @_k\beta_b = @_j\beta_b)$, by Axiom 6e
4. $\vdash @_j k \to (@_k\beta_b = @_j\beta_b)$, by Axiom 4a

Claim 56 $\vdash @_i j \to (@_j k \to @_i k)$.

Claim 57 (Bridge) $\vdash @_i\Diamond j \wedge @_j\varphi \to @_i\Diamond\varphi$.

1. $\vdash @_j\varphi \to (j \to \varphi)$, by Axioms 6b and 1
2. $\vdash @_k(@_j\varphi \to (j \to \varphi))$, by Rule 2b
3. $\vdash @_i\Diamond k \to @_k(@_j\varphi \to (j \to \varphi))$, by Axiom 1 and Rule 1
4. $\vdash @_i\Box(@_j\varphi \to (j \to \varphi))$, by Rule 5, Bounded Generalization
5. $\vdash @_i(\Box @_j\varphi \to \Box(j \to \varphi))$, by Axiom 2a
6. $\vdash @_i\Box @_j\varphi \to @_i\Box(j \to \varphi))$, by Axiom 2b
7. $\vdash @_i @_j\varphi \to @_i\Box(j \to \varphi))$, by Axioms 6c and 6a
8. $\vdash @_j\varphi \to @_i\Box(j \to \varphi))$, by Axiom 6e
9. $\vdash \Box(j \to \varphi) \to (\Diamond j \to \Diamond\varphi)$, by Axiom 2a and Rule 2a
10. $\vdash @_i\Box(j \to \varphi) \to (@_i\Diamond j \to @_i\Diamond\varphi)$, by Axiom 2b and Rule 2b
11. $\vdash @_j\varphi \to (@_i\Diamond j \to @_i\Diamond\varphi)$, by Axiom 1
12. $\vdash @_i\Diamond j \wedge @_j\varphi \to @_i\Diamond\varphi$, by Axiom 1

Claim 58 $\vdash @_i k \to (@_i\Diamond j \to @_k\Diamond j)$.

Claim 59 $\vdash @_i(\varphi \wedge \psi) = @_i\varphi \wedge @_i\psi$.

Claim 60 ($\mathbf{K}_{@}^{-1}$) $\vdash (@_i\varphi \to @_i\psi) \to @_i(\varphi \to \psi)$.

1. $\vdash @_i\neg(\varphi \to \psi) \to @_i\varphi$, by Axiom 1, Rule 2b and Axiom 2b
2. $\vdash @_i\neg(\varphi \to \psi) \to @_i\neg\psi$, by Axiom 1, Rule 2b and Axiom 2b

3. $\vdash @_i\neg(\varphi \to \psi) \to (@_i\varphi \wedge @_i\neg\psi)$, by Axiom 1
4. $\vdash \neg @_i(\varphi \to \psi) \to (@_i\varphi \wedge \neg @_i\psi)$, by Axiom 6a
5. $\vdash (@_i\varphi \to @_i\psi) \to @_i(\varphi \to \psi)$, by Axiom 1

Claim 61 (Arrow Name) *If* $\vdash i \to \varphi$ *then* $\vdash \varphi$ *for* $i \in \mathsf{NOM}$ *not in* φ*.*

1. $\vdash i \to \varphi$, by hypothesis
2. $\vdash @_i(i \to \varphi)$, by Rule 2b
3. $\vdash @_i i \to @_i\varphi$, by Axiom 2b and Rule 1
4. $\vdash @_i\varphi$, by Axiom 6d and Rule 1
5. $\vdash \varphi$, by Axiom 4

Claim 62 (Paste$_\Diamond$) *If* $\vdash (@_i\Diamond j \wedge @_j\varphi) \to \psi$ *and* $j \neq i$ *and* j *does not occur in* φ *and* ψ*, then* $\vdash @_i\Diamond\varphi \to \psi$*.*

1. $\vdash (@_i\Diamond j \wedge @_j\varphi) \to \psi$, by hypothesis
2. $\vdash @_k @_i\Diamond j \to (@_k @_j\varphi \to @_k\psi)$, by Axiom 1, Rule 2b and Axiom 2b
3. $\vdash @_i\Diamond j \to (@_j\varphi \to @_k\psi)$, by Axiom 6e
4. $\vdash @_i\Diamond j \to (@_j\varphi \to @_j @_k\psi)$, by Axiom 6e
5. $\vdash @_i\Diamond j \to @_j(\varphi \to @_k\psi)$, by Claim 60
6. $\vdash @_i\Box(\varphi \to @_k\psi)$, by Rule 5, Bounded Generalization
7. $\vdash \Box(\varphi \to @_k\psi) \to (\Diamond\varphi \to \Diamond @_k\psi)$, by Axioms 2a and 1, and Rule 2a
8. $\vdash @_i\Box(\varphi \to @_k\psi) \to (@_i\Diamond\varphi \to @_i\Diamond @_k\psi)$, by Axiom 2b and Rule 2b
9. $\vdash @_i\Diamond\varphi \to @_i\Diamond @_k\psi$, by Rule 1
10. $\vdash @_k @_i\Diamond\varphi \to @_k\psi$, by Axioms 6c and 6e
11. $\vdash @_k(@_i\Diamond\varphi \to \psi)$, by Claim 60
12. $\vdash @_i\Diamond\varphi \to \psi$, by Rule 4 ($k$ does not occur in $@_i\Diamond\varphi \to \psi$)