Maria Manzano
Manuel Martins
Antonia Huertas

Completeness in Equational Hybrid Propositional Type Theory

Abstract. Equational Hybrid Propositional Type Theory (EHPTT) is a combination of propositional type theory, equational logic and hybrid modal logic. The structures used to interpret the language contain a hierarchy of propositional types, an algebra (a nonempty set with functions) and a Kripke frame.

The main result in this paper is the proof of completeness of a calculus specifically defined for this logic. The completeness proof is based on the three proofs Henkin published last century: (i) Completeness in type theory (ii) The completeness of the first-order functional calculus and (iii) Completeness in propositional type theory. More precisely, from (i) and (ii) we take the idea of building the model described by the maximal consistent set; in our case the maximal consistent set has to be named, $\Diamond$- saturated and extensionally algebraic-saturated due to the hybrid and equational nature of EHPTT. From (iii), we use the result that any element in the hierarchy has a name. The challenge was to deal with all the heterogeneous components in an integrated system.

Keywords: Propositional Type Theory, Hybrid Logic, Equational Logic, Completeness

1. Introduction

In [15] and [16] Manzano and Moreno investigate the concepts of *identity, equality, nameability and completeness* and their mutual relationships on the following areas: first-order logic, second and higher order logic, type theory, first-order modal logic, modal type theory, hybrid type theory and propositional type theory. Concerning identity, we were interested in how to define it using other logical concepts (identity as *definiendum*) as well as in the opposite scheme (identity as *definiens*). The conclusion, as explained in the section "The Kingdom of Identity" of [16], was that identity and lambda are the main logical constants that can be used to define all the other logical constants. As far as equality is concerned, considered as an equivalence relation between terms, the modal environment is crucial to reveal the difference between identity of objects and equality of terms that can receive different denotations at different worlds. Adding to modal logic

Accepted authors' manuscript published as *Completeness in Equational Hybrid Propositional Type Theory*, Mara Manzano, Manuel A. Martins, Antonia Huertas, Studia Logica, Vol. 107(6): 1159-1198, 2019. The final publication is available at `https://rd.springer.com/article/10.1007%2Fs11225-018-9833-5`.

the hybrid perspective, you are able to express identity of worlds as well as accessibility relation between worlds inside the formal language. Finally, equational logic was chosen because there identity is crucial and there are not interferences with other logical concepts.

Our point of departure is the following quote from Ramsey [18], according to which Wittgenstein maintained that logic is nothing but identities: *"The preceding and other considerations let Wittgenstein to the view that mathematics does not consist of tautologies, but of what he called 'equations', for which I should prefer to substitute 'identities'... (It) is interesting to see whether a theory of mathematics could not be constructed with identities for its foundations. I have spent a lot of time developing such a theory, and found that it was faced with what seemed to me unsuperable difficulties."*

The relation of identity is usually understood as the binary relation which holds between any object and itself and which fails to hold between any two distinct objects. By equality we mean a binary relation between terms of the formal language which is reflexive, symmetric and transitive.

What are the logics where equality/identity plays a relevant role? In Henkin [10] we found a logic where equality ($\equiv$) and lambda (λ) are the only primitive logical constants, the logic of Propositional Type Theory (PTT). The main results of that paper are the definitions of the other logical constants, as well as the introduction of a calculus based on equality and lambda. The completeness proof is rather curious as it is not adapted from his well known completeness proofs for type theory and first-order logic. In this case, Henkin was able to give a name to each object in the propositional type hierarchy and he obtained the completeness result as an easy corollary. Another logic where identity is central is Equational Logic (EL) and, therefore, we also incorporate its vision and tools into our paper.

Finally, we wanted to explore intensional contexts, where terms receive different denotations in different worlds. Moreover, when Kripke semantics is used and a domain of worlds appears, we wonder how to treat identity in this domain. That is why we took Hybrid Logic (HL), where the identity relation between worlds can be defined using nominals and the @ operator.

Our proposal is a logic that is able to incorporate all these capabilities. Equational Hybrid Propositional Type Theory (EHPTT) is a combined logic with equational significant ingredients. The language we will use includes as logical symbols $\equiv$ and λ from PTT, but also $\Diamond$ and @ from HL. In a previous paper [13] we prove that there is no need to include a special equality symbol for equations as primitive, since it can be defined with lambda and equality. Our language contains variables of all propositional types as well as individual ones; as non-logical constants we have individual and function constants. Finally, the hybrid logic provides nominals to name worlds and formulas of the form $@_i \Diamond j$ to express that the world named by j is accessible from i.

The structures used to interpret this language include the propositional type hierarchy built upon the set of the two truth values, an algebra, a Kripkean frame with a domain of worlds, an accesibility relation between worlds and the denotation of nominals.

The main result in this paper is the presentation of a set of axioms and the proof that such axiomatization is sound and complete with respect to the intensional Kripke style semantics presented in [13]. Our completeness proof for EHPTT owes much to the three proofs Henkin published last century: (i) *Completeness in the Theory of Types* [7] (ii), *The Completeness of the First-order Functional Calculus* [8] and (iii) Completeness of *A Theory of Propositional Types* [10] (see also [14] for a detailed explanation of these and other Henkin's completeness proofs).

The proof of completeness for our logic EHPTT follows, as usual, from the fact that any consistent set of meaningful sentences has a countable model. To achieve completeness, we use Henkin's method to build the model which is perfectly described by a maximal consistent set of formulas –but not any maximal consistent set will do. To deal with the complexity of our logic, the maximal consistent set needs to be also named and $\Diamond$-saturated, as required in hybrid logic. It is also compulsory for the maximal consistent set to be extensionally algebraic-saturated, so that one can distinguish different functions with rigidified constants witnessing such inequalities. The last property is used by Andrews in his book [2] to prove completeness for a full theory of types based on lambda and equality, and we have adapted the property to our hybrid situation.

What strategy shall one use to build the heterogeneous structures needed for EHPTT?

The Kripkean universe of worlds is just the set of equivalence classes of nominals defined by the sentences of the form $@_i j$ in the maximal consistent set; the accesibility relation includes all pairs of equivalent classes of nominals our oracle declares are accesible from one another via sentences of the form $@_i \Diamond j$.

How do we construct the rest of the structure? In this part we will follow Henkin's [8] recipe almost to the letter. We will do it, as we did in [3], via a function Φ to be defined using equivalent classes of expressions obtained from the information our oracle provides; namely, the maximal consistent set just built.

To define the domain of individuals and the hierarchy of propositional types, as well as the interpretation of the non-logical constants, we not only define the function Φ as acting on expressions of the form $@_i F$ but, simultaneously, we define the domains D^*_α for each type, even for propositional types. In this particular case, instead of taking the standard hierarchy as it is usually defined, we do it through the names of the objects in the hierarchy. In essence, we shall take equivalence classes as elements of the domains, but for type $\langle\alpha\beta\rangle$ we need functions from D^*_α to D^*_β. So we define Φ as a map which corresponds, in a proper sense, to the equivalence classes.

Why do we proceed that way in the propositional case? To better integrate our heterogeneous logic ingredients, the propositional type hierarchy is also built with the function Φ, aside from the final result being the standard one, it also entails that each type in the hierarchy is the value of its own name under Φ. In this case, the already mentioned result of PTT concerning the nameability of each type is necessary.

What elements are there in the algebra? From Henkin's first-order completeness proof [7] we take the idea of building the structure out of equivalent classes of individuals using them as objects. Here we do a similar thing, but in our case we need rigidified terms provided by the satisfaction operator @. Nominals and expressions of the form $@_i\tau$ play a central role in the completeness result: nominals are the building blocks of the world universe, while $@_i\tau$ expressions supply the architectural blueprint in the algebraic construction.

Outline of the paper. The ouline of the paper, organized in five more sections, is the following.

Section 2 is an overview of the main results of Henkin's propositional type theory [10]. Also, its language and semantics are formulated in an updated form that makes it easier to use. There, the classical connectives and quantifiers are defined using only lambda and equality. Moreover, the combination of lambda plus equality goes much further, as they provide a name for each object in the propositional type hierarchy. The tool used to name all the objects, the descriptor operator, is also presented and commented in order to understand the beautiful and useful construction of Henkin. Names for propositional types and denotations do match, as we can prove a *Nameability Theorem* saying that one can associate with each element χ of an arbitrary type a closed formula χ^{N} of the corresponding type such that the interpretation of this sentence turns out to be χ. Finally, Henkin's calculus and the main idea behind its interesting and novel completeness theorem is described.

In section 3, the syntax and semantics of the EHPTT logic is presented.

As it is the result of the combination of PTT, HL and EL, there are characteristics elements of each of them. Furthermore, the integration of the three logics into the new one, is illustrated by the fact that the logical symbols of EHPTT are only four: the lambda operator λ, used for building functions in the hierarchy of types and also the algebraic equations; the equality symbol between expressions of several types $\equiv$; and the modal-hybrid operators of possibility $\Diamond$ and satisfaction @. All the remaining logical symbols (propositional connectives, quantifiers, algebraic equations, intensional modal expressions, etc.) can be derived from those. Moreover, the semantics of EHPTT is intensional, since the interpretation of algebraic individual constants and nominals are functions on the set of possible worlds. In addition, it contains the standard hierarchy of types and focuses on equations and the identity relation.

In section 4, the EHPTT calculus is defined. On the one hand, it extends that of Henkin's in [10]. On the other hand, it contains the most characteristic axioms of the basic hybrid logic that capture the operator @. Likewise, the functional aspect of algebraic logic and propositional type theory is also an essential element.

In sections 5 and 6, the completeness proof for EHPTT is developed. It is a Henkin-style completeness proof for Type Theory. Thus, in order to build the model of a consistent set of formulas, a maximal consistent set fulfilling the appropriate properties is used. We define a function Φ on rigidified expressions of the form $@_i F$, and simultaneously the domains of the model that the maximal consistent set describes. Besides, the "names" of the objects in the hierarchy of types play a relevant role, as in Henkin's original proof. This is the "type theoretical" part of the proof. On the other hand, the hybrid element of EHPTT is represented by the central role of the rigidified expressions, and by two important properties of the maximal consistent set used in the proof: those of being named (containing at least a world) and being $\Diamond$-saturated. Finally, the equational part of the logic requires another key property of the maximal consistent set of the proof; that of being extensionally algebraic-saturated.

2. Background: Henkin propositional types

In this section we will sketch briefly the main results of Henkin's propositional type theory as presented in [10] and also discussed in [15, 16].

In the first place, we introduce the formalism and its semantics. We have adapted his language in order to make an easier reading and recognize his theory within ours. Using only lambda and equality, Henkin was able to introduce the classical connectives and quantifiers as defined operators. And not only that, the language also provides a name for each object in the hierarchy. The fundamental tool used to name all the objects is the descriptor operator, which is also defined in the language using, in this case, an election function for each type. Due to the fact that each type is finite, we easily construct such a function without requiring the axiom of choice.

Finally, we recall Henkin's calculus and describe the main idea behind this interesting and novel completeness theorem. Henkin uses the names of the objects of the hierarchy to get that completeness result.

2.1. Language and semantics

According to Henkin's definition, the *hierarchy of propositional types*, $\mathfrak{PT}$, is the least class of sets containing D_t as an element, which is closed under passage from D_α and D_β to $\mathsf{D}_{\langle\alpha\beta\rangle}$. Here D_t is the two truth values set, $\mathsf{D}_t = \{T, F\}$, while $\mathsf{D}_{\langle\alpha,\beta\rangle}$ is the set of all functions mapping D_α to D_β.[1] Let us call PT to the set of type symbols[2]:

$$\mathsf{PT} ::= t \mid \langle\alpha\beta\rangle \text{ for any } \alpha, \beta \in \mathsf{PT}$$

To build the *theory of propositional types,* Henkin introduces a formal language with variables for each type, the lambda abstractor, λ, and a collection of equality constants, $\equiv_{\langle\alpha,\langle\alpha t\rangle\rangle}$, one for each propositional type $\alpha \in \mathsf{PT}$. To be more specific, expressions of this theory are either: (1) variables of any type X_α, (2) the constants $\equiv_{\langle\alpha\langle\alpha t\rangle\rangle}$, (3) $A_{\langle\beta\alpha\rangle}B_\beta$ or (4) $\lambda X_\beta B_\alpha$.

Interpretations of these expressions on the hierarchy $\mathfrak{PT}$ are recursively defined with the help of assignments, which give values in $\mathfrak{PT}$ to variables of all types. In particular, for a given assignment g, we recursively define the interpretation $\mathsf{V}(A_\alpha, g)$ for any A_α: (1) $\mathsf{V}(X_\alpha, g) = g(X_\alpha)$, (2) $\mathsf{V}(\equiv_{\langle\alpha\langle\alpha t\rangle\rangle}, g)$ is the identity relation on type α, (3) $\mathsf{V}(A_{\langle\beta\alpha\rangle}B_\beta, g)$ is the value of the function $\mathsf{V}(A_{\langle\beta\alpha\rangle}, g)$ for the argument $\mathsf{V}(B_\beta, g)$ and (4) $\mathsf{V}(\lambda X_\alpha B_\beta, g)$ is the function of $\mathsf{D}_{\langle\alpha\beta\rangle}$ whose value for any $\chi \in \mathsf{D}_\alpha$ is the element $\mathsf{V}(B_\beta, g^{\chi}_{X_\alpha})$ of D_β [3].

Classical logical constants as defined operators. As we mentioned already, using only the equality $\equiv_{\langle\alpha\langle\alpha t\rangle\rangle}$ and λ, the remaining connectives as well as the quantifiers $\forall X_\alpha$ —for each propositional variable of any propositional type α— are presented as defined operators.

[1]Henkin uses the reverse notation; namely, $\mathsf{D}_{\langle\alpha\beta\rangle}$ is the set of all functions from D_β to D_α but nowadays it is more common to use the one we have adopted.

[2]Even though $\mathfrak{PT}$ and PT are different sets, it is common to refer to both as types.

[3]Here $g^{\chi}_{X_\alpha}$ is the $X_\alpha-$ variant of g sending X_α to χ and leaving the rest of values as in g.

Definition 1 (Logical constants). *We define the following expressions*[4]

1. $T^{\mathsf{N}} ::= ((\lambda X_t X_t) \equiv (\lambda X_t X_t))$
2. $F^{\mathsf{N}} ::= \big((\lambda X_t X_t) \equiv \big(\lambda X_t T^{\mathsf{N}}\big)\big)$
3. $\neg ::= \big(\lambda X_t \big(F^{\mathsf{N}} \equiv X_t\big)\big)$
4. $\wedge ::= \lambda X_t \big(\lambda Y_t \left(\lambda f_{tt} \left(f_{tt} X_t \equiv Y_t\right)\right) \equiv \big(\lambda f_{tt} \big(f_{tt} T^{\mathsf{N}}\big)\big)\big)$
5. $\forall X_\alpha A_t ::= \big((\lambda X_\alpha A_t) \equiv \big(\lambda X_\alpha T^{\mathsf{N}}\big)\big)$

Nameability Theorem. The power of the combination of lambda plus equality goes much further, as these symbols provide a name for each object in the propositional type hierarchy.

The strategy described in [10] to produce a name for each object in $\mathfrak{PT}$ rests upon the power of the description operator. In order to define the description operator properly, one fixes one element for each propositional type; this element would serve as the denotation of improper descriptions. The definition is done by induction on types: for type t we just take $\mathbf{a}_t = F$; for type $\langle\alpha\beta\rangle$ we take the constant function $\mathbf{f}_{\langle\alpha\beta\rangle}$ with value $\mathbf{b}_\beta$ for every element of D_α, where $\mathbf{b}_\beta$ is the element in D_β already chosen. Thus, $\mathbf{f}_{\langle\alpha\beta\rangle}\chi = \mathbf{b}_\beta$ for each $\chi \in \mathsf{D}_\alpha$. Now, using these elements, an election function $\mathbf{t}^{(\alpha)}$ can be defined for each type. For any arbitrary type α let $\mathbf{t}^{(\alpha)}$ be the function of $\mathsf{D}_{\langle\langle\alpha t\rangle\alpha\rangle}$ such that, for any $\chi_{\langle\alpha,t\rangle} \in \mathsf{D}_{\langle\alpha t\rangle}$, we have that $(\mathbf{t}^{(\alpha)}\chi_{\langle\alpha t\rangle})$ is the unique element $\chi_\alpha \in \mathsf{D}_\alpha$ for which $(\chi_{\langle\alpha t\rangle}\chi_\alpha) = T$, in case there is such a unique element χ_α, or else $(\mathbf{t}^{(\alpha)}\chi_{\langle\alpha t\rangle}) = \mathbf{a}_\alpha$ if there is no such a particular χ_α, or if there are more than one χ_α, such that $(\chi_{\langle\alpha t\rangle}\chi_\alpha) = T$. Having done that Henkin continues: '*We shall show inductively that for each α there is a closed formula $\iota_{\langle\langle\alpha,t\rangle,\alpha\rangle}$ such that $(\iota_{\langle\langle\alpha t\rangle,\alpha\rangle})^d = \mathbf{t}^{(\alpha)}$. Then, for any formula A_t and variable X_α we shall set $(\jmath X_\alpha A_t) = (\iota_{\langle\langle\alpha,t\rangle,\alpha\rangle}(\lambda X_\alpha A_t))$.*'[5]

Having introduced the description operator, we will produce a name χ^{N}_α for each $\chi_\alpha \in \mathsf{D}_\alpha$ and then prove the renowned "nameability theorem".

Theorem 2 (Nameability Theorem). *For each element $\chi \in \mathsf{D}_\alpha$ of any arbitrary type α, there exists a sentence χ^{N} of type α such that $\mathsf{V}(\chi^{\mathsf{N}}, g) = \chi$ for any assignment g.*

---

[4]These definitions are a rewording of the ones in [10], pages 326-327.
The definition of $\wedge$ offered by Andrews in page 160 of [2] is

$$\wedge ::= \lambda X_t \big(\lambda Y_t \left(\lambda f_{ttt}(f_{ttt} X_t Y_t)\right) \equiv \big(\lambda f_{ttt}(f_{ttt} T^{\mathsf{N}} T^{\mathsf{N}})\big)\big)$$

and his explanation reads: '$(\lambda f_{ttt}(f_{ttt} X_t Y_t)$ *can be used to represent the ordered pair* $\langle X_t, Y_t\rangle$ *and the conjunction* $X_t \wedge Y_t$ *is true iff* X_t *and* Y_t *are both true, i.e., iff* $\langle X_t, Y_t\rangle \equiv \langle T^{\mathsf{N}}, T^{\mathsf{N}}\rangle$

[5]Henkin [10], p. 328.

Henkin proves this theorem by induction on the hierarchy's construction. Names for the basic object T and F of type t are given by the previous definition 1. For type $\langle\alpha\beta\rangle$, assuming that the theorem is proven for types α and β, we set a name for every function $\chi_{\langle\alpha\beta\rangle}$ which maps every element $(\chi_\alpha)_i$ of the finite type D_α, say $\mathsf{D}_\alpha = \{(\chi_\alpha)_1, \ldots, (\chi_\alpha)_q\}$, to the corresponding value in D_β, that is, to $(\chi_{\langle\alpha\beta\rangle}(\chi_\alpha)_i)$. To this effect, the names of the objects in D_α and D_β (whose existence is assumed by induction hypothesis) as well as the descriptor operator are used. To introduce $\chi^{\mathsf{N}}_{\langle\alpha\beta\rangle}$ we need to formalize the following: when variable X_α is just the name of an object $(\chi_\alpha)_i$ of type α —that is, $X_\alpha \equiv (\chi_\alpha)^{\mathsf{N}}_i$— function $\chi_{\langle\alpha\beta\rangle}$ matches it to the unique Z_β naming $(\chi_{\langle\alpha\beta\rangle}(\chi_\alpha)_i)$ —that is, $Z_\beta \equiv (\chi_{\langle\alpha\beta\rangle}(\chi_\alpha)_i)^{\mathsf{N}}$. In particular,

$$\begin{aligned}\chi^{\mathsf{N}}_{\langle\alpha\beta\rangle} \quad &:= \lambda X_\alpha.\jmath Z_\beta.\left[(X_\alpha \equiv (\chi_\alpha)^{\mathsf{N}}_1) \wedge (Z_\beta \equiv (\chi_{\langle\alpha\beta\rangle}(\chi_\alpha)_1)^{\mathsf{N}})\right] \vee \ldots \\ &\qquad \ldots \vee \left[(X_\alpha \equiv (\chi_\alpha)^{\mathsf{N}}_q) \wedge (Z_\beta \equiv (\chi_{\langle\alpha\beta\rangle}(\chi_\alpha)_q)^{\mathsf{N}})\right]\end{aligned}$$

2.2. Calculus

For the theory of propositional types Henkin offers a calculus based on λ and equality rules[6]. Let us quote Henkin's seven axioms and *Replacement Rule* (See Henkin [10], page 330).

5.1.1. Axiom Schema 1. $A_\alpha \equiv A_\alpha$
5.1.2. Axiom Schema 2. $(A_t \equiv T^{\mathsf{N}}) \equiv A_t$
5.1.3. Axiom Schema 3. $(T^{\mathsf{N}} \wedge F^{\mathsf{N}}) \equiv F^{\mathsf{N}}$
5.1.4. Axiom Schema 4. $(g_{\langle tt\rangle}T^{\mathsf{N}} \wedge g_{\langle tt\rangle}F^{\mathsf{N}}) \equiv (\forall X_t(g_{\langle tt\rangle}X_t))$
5.1.5. Axiom Schema 5. $(X_\alpha \equiv Y_\alpha) \rightarrow ((Z_{\langle\alpha\beta\rangle} \equiv V_{\langle\alpha,\beta\rangle}) \rightarrow ((Z_{\langle\alpha\beta\rangle}X_\alpha) \equiv (V_{\langle\alpha\beta\rangle}Y_\alpha)))$
5.1.6. Axiom Schema 6. $(\forall X_\alpha((Z_{\langle\alpha\beta\rangle}X_\alpha) \equiv (V_{\langle\alpha\beta\rangle}X_\alpha))) \rightarrow (Z_{\langle\alpha\beta\rangle} \equiv V_{\langle\alpha\beta\rangle})$
5.1.7. Axiom Schema 7. $((\lambda X_\alpha B_\beta)A_\alpha) \equiv C_\beta$, where C_β is obtained from B_β by replacing each occurrence of X_α in B_β by an occurrence of A_α, provided no such occurrence of X_α is within a part of B_β which is a formula beginning 'λY_γ' where Y_γ is a variable free in A_α.

5.2. By the Rule of Replacement we refer to the ternary relation on formulas of type t which holds for $\langle A'_t, C_t, D_t\rangle$ if and only if $A'_t = (A_\alpha \equiv B_\alpha)$ for some formulas A_α and B_α and D_t is obtained from C_t by replacing one occurrence of A_α by an occurrence of B_α. When this

[6] This calculus was improved by Andrews [1]. In particular, Andrews eliminates Axioms 1, 2 and 3 and replaces Axiom 6 for $(\forall X_\alpha((Z_{\langle\alpha\beta\rangle}X_\alpha) \equiv (V_{\langle\alpha\beta\rangle}X_\alpha))) \equiv (Z_{\langle\alpha\beta\rangle} \equiv V_{\langle\alpha\beta\rangle})$.

situation holds for $\langle A_\alpha \equiv B_\alpha, C_t, D_t \rangle$ we shall say that D_t *is obtained by Rule R from* $A_\alpha \equiv B_\alpha$ *and* C_t.

2.2.1. Completeness

This calculus is complete. The method of proof is rather different from Henkin's previous completeness theorems for type theory [8] and first-order logic [7]. The next lemma and proposition synthesize the whole idea of the proof which rest upon the Nameability theorem 2 (cf. [10]). The important result from where the completeness theorem easily follows has the amazing form:

LEMMA 3. *For any formula* A_α*and assignment* g

$$\vdash A_\alpha \frac{(g(X_{\beta_1}))^{\mathsf{N}} ... (g(X_{\beta_m}))^{\mathsf{N}}}{X_{\beta_1} ... X_{\beta_m}} \equiv (\mathsf{V}(A_\alpha, g))^{\mathsf{N}}$$

where $\mathsf{FreeVar}(A_\alpha) = \{X_{\beta_1} ... X_{\beta_m}\}$.

As you see, the formulation resembles the substitution lemma, but instead of being a semantics metatheorem it is a theorem of the calculus. The lemma is proved in [10], pp. 341-343, by induction on the length of A_α.

The obvious question we ask is, *how to prove that* $\models A_t$ *implies* $\vdash A_t$ *for any formula of type* t *?* We will see that Lemma 3 implies completeness.

THEOREM 4 (Completeness). $\models A_t$ *implies* $\vdash A_t$*, for any formula of type* t.

PROOF. If A_t is closed then $\models A_t$ implies $\mathsf{V}(A_t, g) = T$ for any assignment g. Thus the Lemma 3 gives $\vdash A_t \equiv (\mathsf{V}(A_t, g))^{\mathsf{N}}$ which turns to be $\vdash A_t \equiv T^{\mathsf{N}}$, where T^{N} is the name of the truth value true.

But using the calculus, in particular, AXIOM SCHEMA 2 and the *Rule of replacement R,* we obtain the desired result, $\vdash A_t$.

In case A_t were a valid formula, $\models A_t$, but not a sentence, we pass from A_t to the sentence $\forall X_{\gamma_1} ... X_{\gamma_r} A_t$, where $\mathsf{FreeVar}(A_t) = \{X_{\gamma_1}, \ldots, X_{\gamma_r}\}$. Clearly we have that $\models \forall X_{\gamma_1} ... X_{\gamma_r} A_t$,. Now, using the previous argument, we get $\vdash \forall X_{\gamma_1} ... X_{\gamma_r} A_t$. Applying the rules of the calculus, we obtain $\vdash A_t$. ■

2.3. Other important results

Theorem 32 states that all theorems of Henkin's propositional type theory are also provable in our EHPTT calculus. That means that we can just give the reference of Henkin's demonstration, when necessary.

LEMMA 5 ([10], page 340). *For each* $\chi_{\langle\alpha\beta\rangle} \in \mathsf{D}_{\langle\alpha\beta\rangle}$ *and* $\chi_\alpha \in D_\alpha$ *we have*

$$\vdash \chi^{\mathsf{N}}_{\langle\alpha\beta\rangle}\chi^{\mathsf{N}}_{\alpha} \equiv (\chi_{\langle\alpha\beta\rangle}\chi_\alpha)^{\mathsf{N}}$$

LEMMA 6. *For each type* α *and assignment* g*, we have*

$$\mathsf{V}((\exists X_\alpha A_t) \equiv (A_t\frac{\chi_1^{\mathsf{N}}}{X_\alpha} \vee ... \vee A_t\frac{\chi_m^{\mathsf{N}}}{X_\alpha}), g) = T,$$

where $\mathsf{D}_\alpha = \{\chi_1, \ldots, \chi_m\}$ *(as* D_α *is finite).*

PROOF. Let g be any assignment. By definition $V((\exists X_\alpha A_t \equiv A_t\frac{\chi_1^{\mathsf{N}}}{X_\alpha} \vee ... \vee A_t\frac{\chi_m^{\mathsf{N}}}{X_\alpha}), g) = T$ iff $\mathsf{V}(\exists X_\alpha A_t, g) = \mathsf{V}(A_t\frac{\chi_1^{\mathsf{N}}}{X_\alpha} \vee ... \vee A_t\frac{\chi_m^{\mathsf{N}}}{X_\alpha}, g)$

We will see that $\mathsf{V}(\exists X_\alpha A_t, g) = T$ iff $\mathsf{V}(A_t\frac{\chi_1^{\mathsf{N}}}{X_\alpha} \vee ... \vee A_t\frac{\chi_m^{\mathsf{N}}}{X_\alpha}, g) = T$.

($\Longrightarrow$) Let $\mathsf{V}(\exists X_\alpha A_t, g) = T$, then there is an $\chi_p \in \mathsf{D}_\alpha$ such that $\mathsf{V}(A_t, g^{\chi_p}_{X_\alpha}) = T$. Using Lemma 3 above, $\vdash A_t\frac{(g^{\chi_p}_{X_\alpha}(X_\alpha))^{\mathsf{N}}}{X_\alpha} \equiv (\mathsf{V}(A_t, g^{\chi_p}_{X_\alpha}))^{\mathsf{N}}$ and then $\vdash A_t\frac{\chi_p^{\mathsf{N}}}{X_\alpha} \equiv T^{\mathsf{N}}$.

Using the calculus, in particular AXIOM 2 and the rule of replacement we get $\vdash A_t\frac{\chi_p^{\mathsf{N}}}{X_\alpha}$. By soundness we obtain that $\mathsf{V}(A_t\frac{\chi_p^{\mathsf{N}}}{X_\alpha}, g) = T$.

Finally, $\mathsf{V}(A_t\frac{\chi_1^{\mathsf{N}}}{X_\alpha} \vee ... \vee A_t\frac{\chi_m^{\mathsf{N}}}{X_\alpha}, g) = T$.

($\Longleftarrow$) Let $\mathsf{V}(A_t\frac{\chi_1^{\mathsf{N}}}{X_\alpha} \vee ... \vee A_t\frac{\chi_m^{\mathsf{N}}}{X_\alpha}, g) = T$. Therefore, $\mathsf{V}(A_t\frac{\chi_q^{\mathsf{N}}}{X_\alpha}, g) = T$ for at least one of the disjuncts. By Substitution Lemma we get $\mathsf{V}(A_t, g^{\chi_q}_{X_\alpha}) = T$, and so $\mathsf{V}(\exists X_\alpha A_t, g) = T$, as needed. ■

THEOREM 7. *For each type* α*, we have*

$$\vdash (\exists X_\alpha A_t) \equiv (A_t\frac{\chi_1^{\mathsf{N}}}{X_\alpha} \vee \cdots \vee A_t\frac{\chi_n^{\mathsf{N}}}{X_\alpha}),$$

where $\mathsf{D}_\alpha = \{\chi_1, \ldots, \chi_m\}$.

PROOF. We have already proven its validity as Lemma 6. Then we can use Henkin's completeness Theorem 4 for Propositional Type Theory. ■

3. Equational Hybrid Propositional Type Theory (EHPTT)

We are facing the challenge posed by the identity relation and the equality symbol by designing a language with four basic components: propositional type theory, first-order equational logic, modal and hybrid logics.

3.1. Syntax

DEFINITION 8 (Type Symbols). *Let* t *and* 0 *be any fixed objects. The set* $\mathsf{TYPES} = \mathsf{AT} \cup \mathsf{PT}$ *of types of* EHPTT *is defined as follows:*

- ***Algebraic Type Symbols***, $\mathsf{AT} ::= 0 \mid \langle\underbrace{0\ldots0}_{n\ times}\rangle$ *(to simplify notation we will write n for $\langle\underbrace{0,\ldots,0}_{n\ times}\rangle$).*
- ***Propositional Type Symbols***, $\mathsf{PT} ::= t \mid \langle\alpha\beta\rangle$, $\alpha\beta \in \mathsf{PT}$

In the sequel we will use a, b for indiscriminate types.

Definition 9 (Language). *The set of* ***meaningful expressions of*** EHPTT *is built on the* EHPTT *language containing:*

- *a family* $\mathsf{CON} = \langle \mathsf{CON}_n : n \in \mathsf{AT}\rangle$ *of at most denumerably infinite sets of* ***non-logical constants*** *such that for each n, CON_n is a set of functional symbols of type n for $n \neq 0$ and CON_0 is a set of individual constants.*
- *a denumerably infinite set VAR_a of* ***variables*** *V_a, for each type $a \in \mathsf{PT} \cup \{0\}$; we will use $X_\alpha, Y_\alpha, Z_\alpha, \ldots$ for variables of propositional type α, and $v_1, v_2, v_3, \ldots$ for individual variables.*
- *a denumerably infinite set* NOM *of* ***nominals***.
- *the only primitive logical symbols are equality ($\equiv$), lambda (λ), satisfaction operator (@) and the modality ($\Box$).*

Definition 10. *By recursion we define the set ME_a of* ***meaningful expressions of type*** *a.*

- *Algebraic terms (ME_n):*

$$v \in \mathsf{ME}_0 \mid c \in \mathsf{ME}_n \mid \lambda v_1 \cdots v_n \tau \in \mathsf{ME}_n \mid \gamma(\tau_1, \ldots, \tau_n) \in \mathsf{ME}_0 \mid @_i\gamma \in \mathsf{ME}_n$$

for any $v \in \mathsf{VAR}_0$, $c \in \mathsf{CON}_n$ (any $n \in \mathsf{AT}$), $v_1, \cdots, v_n \in \mathsf{VAR}_0$, $v_p \neq v_m$ (for $p \neq m$) and $\tau \in \mathsf{ME}_0$, $\gamma \in \mathsf{ME}_n$ and $\tau_1, \ldots, \tau_n \in \mathsf{ME}_0$. For lambda function we also add the condition that $\mathsf{Free}(\tau) \subseteq \{v_1, \cdots, v_n\}$.

- *Propositional terms (ME_α):*

$$X_\alpha \in \mathsf{ME}_a \mid \lambda X_\delta A_\beta \in \mathsf{ME}_{\langle\delta\beta\rangle} \mid A_{\langle\beta\alpha\rangle} B_\beta \in \mathsf{ME}_\alpha \mid @_i A_\beta \in \mathsf{ME}_\beta$$

for any $X_\alpha \in \mathsf{VAR}_\alpha$, $\alpha \in \mathsf{PT} - \{t\}$, $A_\beta \in \mathsf{ME}_\beta$, $\delta, \beta \in \mathsf{PT} - \{\mathsf{t}\}$. As in the previous case, only lambda terms contain bounded variables.

- *Formulas (ME_t):*

$$X_t \in \mathsf{ME}_t \mid i \in \mathsf{ME}_t \mid A_{\langle\alpha t\rangle} B_\alpha \in \mathsf{ME}_t \mid F_a \equiv G_a \mid \Box\varphi \in \mathsf{ME}_t \mid @_i\varphi \in \mathsf{ME}_n$$

for any $X_t \in \mathsf{VAR}_t$, $i \in$ NOM, $B_\alpha \in \mathsf{ME}_\alpha$, $\alpha \in \mathsf{PT}$, $\{F_a, G_a\} \subseteq \mathsf{ME}_a$, $a \in \mathsf{PT} \cup \mathsf{AT} - \{0\}$, $\varphi \in \mathsf{ME}_t$.

The set $\mathsf{Free}(\epsilon)$ of **free variables** in expression ϵ is defined in the usual way, being lambda functions the only expressions where variables are bounded. ∎

Remark 11.

- *As you probably noticed, formulas like* $F_a \equiv G_a$ *(with* $F_a, G_a \in \mathsf{ME}_a$ *and* $a \in \mathsf{PT} \cup \mathsf{AT} - \{0\}$*) are rather heterogeneous as they can have both algebraic and propositional components —as in* $X_t \equiv (\lambda v_1 \cdots v_n c \equiv \delta) \in \mathsf{ME}_t$ *(with* $X_t \in \mathsf{VAR}_t$*,* $c \in \mathsf{ME}_0$ *and* $\delta \in \mathsf{ME}_n$ *for* $n \neq 0$*); they also can be* pure algebraic*, —like* $\gamma \equiv \lambda vv \in \mathsf{ME}_t$ *(with* $\gamma \in \mathsf{ME}_1$*)— and pure propositional —like* $(\lambda Y_\alpha Y_\alpha) \equiv X_{\langle\alpha\alpha\rangle} \in \mathsf{ME}_t$ *(with* $Y_\alpha \in \mathsf{VAR}_\alpha, X_{\langle\alpha\alpha\rangle} \in \mathsf{VAR}_{\langle\alpha\alpha\rangle}$*,* $\alpha \in \mathsf{PT}$*).*
- *All pure algebraic formulas are closed.*
- *The next definition extends Definition 1 with other logical symbols in terms of* λ *and* $\equiv$*, like algebraic equations and propositional quantifiers for all formulas of* EHPTT*. Having defined* $\neg$ *and* $\wedge$*, we will use the common definitions of* $\vee$ *and* $\rightarrow$ *using them. We will use as well the operator* $\perp$ *instead of* F^{N}*. Moreover,* $\Diamond A_t$ *is defined as* $\neg\Box\neg A_t$ *and* $\exists X_\alpha A_t$ *is defined as* $\neg\forall X_\alpha \neg A_t$*.*

We extend Definition 10 with the two items below:

Definition 12 (Logical operators).

- $\forall X_\alpha A_t ::= \big((\lambda X_\alpha A_t) \equiv \big(\lambda X_\alpha T^{\mathsf{N}}\big)\big)$, *where* A_t *is a formula of* EHPTT*.*
- $\tau \approx \sigma ::= \lambda \bar{v}\tau \equiv \lambda \bar{v}\sigma$*, where* $\mathsf{VarFree}(\tau) \cup \mathsf{VarFree}(\sigma) \subseteq \{v_1, \ldots, v_n\} \neq \varnothing$ *and* $\bar{v} = \langle v_1, \ldots, v_n\rangle$*. It is a closed formula of type* t*. These formulas are called "equations"*[7]*.*

Once we define the semantics for our EHPTT in section 3.2, it is easy to see that all the introduced connectives behave as expected. That is, with equality, abstraction, nominals, $\Box$ and @ we can define all basic standard operators needed for equational hybrid propositional type theory.

It is important to identify the expressions coming from propositional type theory as a basic ingredient of our combined logic, we will call these expressions "strictly propositional". These expressions are actually Henkin's as defined in [10]. They will play a decisive role in our approach to EHPTT. Next we consider those formulas of our system which correspond to formulas of the ordinary untyped propositional logic.

[7]Note: In case τ or σ are constants the corresponding λ-functions are constant functions.

Definition 13. *We say that an expression $A_\beta \in \mathsf{ME}_\beta$ is* ***strictly propositional*** *if it is a meaninful expression recursively built using only the following rules:*

1. $\mathsf{VAR}_\alpha \subseteq \mathsf{ME}_\alpha, \alpha \in \mathsf{PT}$
2. $\lambda X_\alpha A_\beta \in \mathsf{ME}_{\langle\alpha\beta\rangle}$, *if* $X_\alpha \in \mathsf{VAR}_\alpha, A_\beta \in \mathsf{ME}_\beta$, $\alpha\beta \in \mathsf{PT}$
3. $A_{\langle\alpha\beta\rangle} B_\alpha \in \mathsf{ME}_\beta$, *if* $A_{\langle\alpha\beta\rangle} \in \mathsf{ME}_{\langle\alpha\beta\rangle}, B_\alpha \in \mathsf{ME}_\alpha$, $\alpha \in \mathsf{PT}$ *and* $\beta \in \mathsf{PT}$
4. $A_\alpha \equiv B_\alpha \in \mathsf{ME}_t$, *if* $A_\alpha, B_\alpha \in \mathsf{ME}_\alpha$, $\alpha \in \mathsf{PT}$.

Definition 14. *We define the class of P-*formula *to be the least class of formulas containing T^{N} and F^{N} and each variable X_t of type t as members, which is such that whenever A_t and B_t are in the class, then so are $\neg A_t$, $A_t \wedge B_t$, $A_t \vee B_t$, $A_t \rightarrow B_t$ and $A_t \equiv B_t$.*

3.2. Semantics

In EHPTT the semantic is intensional (*i.e.,* every meaningful expression has an intensional interpretation) while the language has no symbols of intensional types. In practice, a meaningful expression of type a receives as interpretation a function from the set of worlds W to the universe D_a, that function is an object of intensional type.

Definition 15. *A structure for* EHPTT *is a tuple $\mathfrak{M} = \langle W, R, \mathcal{A}, \mathfrak{PT}, I\rangle$, where:*

1. *W is the* ***set of worlds****, $W \neq \varnothing$, $R \subseteq W \times W$ is the* ***accessibility relation*** *and $\mathcal{A}$ is a non empty set - the* ***carrier set*** *of the algebras (in order to unify notation we will use $\mathsf{D}_0 = \mathcal{A}$).*
2. *$\mathfrak{PT}$=$\langle \mathsf{D}_\alpha\rangle_{\alpha\in\mathsf{PT}}$, the* ***hierarchy of standard propositional types****, is defined recursively by:*
   $\mathsf{D}_t = \{T, F\}$, $\mathsf{D}_{\langle\alpha\beta\rangle} = {\mathsf{D}_\beta}^{\mathsf{D}_\alpha}$ *($\alpha\beta \in \mathsf{PT}$),*
3. *I is a function whose domain is the union of the set of nominals with the set of all individual constants and the set of functional symbols such that*
   - *If $i \in \mathsf{NOM}$, $I(i) : W \longrightarrow \mathsf{D}_t$ such that $I(i)^{-1}(T)$ is a singleton.*
     We denote by w^i the unique element w of W such that $I(i)(w) = T$.
   - *If $c \in \mathsf{CON}_0$, $I(c) : W \longrightarrow \mathsf{D}_0$*
   - *If $f \in \mathsf{CON}_n$, $I(f) : W \longrightarrow \mathsf{D}_0^{(\mathsf{D}_0)^n}$, with $I(f)(w) : (\mathsf{D}_0)^n \longrightarrow \mathsf{D}_0$*

DEFINITION 16. *An* ***assignment*** g ***of values to variables*** *is a function having as domain the set* $\mathsf{VAR} = \mathsf{VAR}_0 \cup \mathsf{VAR}_{\mathsf{PT}}$ *of all variables, and such that for any variable* $V_a \in \mathsf{VAR}$*, the value is in the appropriate domain, that is,* $g(V_a) \in \mathsf{D}_a$*, for all* $a \in \mathsf{PT} \cup \{0\}$.

Observe that the assignment is extensional since the value of any variable of any type is an object of the same type. The interpretation of a variable of type a is *redefined* below as a constant function of intensional type.

We also define, as usual, that an assignment g' is a V_a-variant of an assignment g if it coincides with g in all values except perhaps in the value assigned to V_a. We will use $g^{\theta}_{V_a}$ to denote the V_a-variant assignment of g whose value for variable V_a is $\theta \in \mathsf{D}_a$. Namely, $g^{\theta}_{V_a}(u) = g(u)$ for any $u \neq V_a$ while $g^{\theta}_{V_a}(V_a) = \theta$.

DEFINITION 17. *An* ***interpretation for*** EHPTT *is a pair* $\mathfrak{I} = \langle \mathfrak{M}, g \rangle$*, where* $\mathfrak{M}$ *is a structure for* EHPTT *and* g *is an assignment of values to variables. We denote the interpretation* $\left\langle \mathfrak{M}, g^{\theta}_{V_a} \right\rangle$ *by* $\mathfrak{I}^{\theta}_{V_a}$.

Given a structure $\mathfrak{M}$ *and an assignment* g *we recursively define for any expression* F_a ***the interpretation of*** F_a ***with respect to*** $\mathfrak{I}$*, denoted by* $(F_a)^{\mathfrak{I}}$.

1. *Algebraic terms. Let* $w \in W$ *be arbitrary*
   - $v^{\mathfrak{I}} : W \longrightarrow \mathsf{D}_0$, $v^{\mathfrak{I}}(w) = g(v)$*, for any* $v \in \mathsf{VAR}_0$
   - $c^{\mathfrak{I}} : W \longrightarrow \mathsf{D}_0$, $c^{\mathfrak{I}}(w) = (I(c))\,(w)$ *for any* $c \in \mathsf{CON}_0$
   - $f^{\mathfrak{I}} : W \longrightarrow \mathsf{D}_n$, $f^{\mathfrak{I}}(w) = (I(f))\,(w)$ *for any* $f \in \mathsf{CON}_n$, $n \neq 0$
   - $(\lambda v_1 \cdots v_n \tau)^{\mathfrak{I}} : W \longrightarrow \mathsf{D}_0^{(\mathsf{D}_0)^n}$, $(\lambda v_1 \cdots v_n \tau)^{\mathfrak{I}}(w) : (\mathsf{D}_0)^n \longrightarrow \mathsf{D}_0$*, mapping* $\overline{a}$ *to* $\tau^{\mathfrak{I}^{\overline{a}}_{\overline{v}}}(w)$*, where* $\overline{a} = (a_1, \ldots, a_n)$, $\overline{v} = (v_1, \ldots, v_n)$ *and* $\mathfrak{I}^{\overline{a}}_{\overline{v}}$ *is the interpretation* $\left\langle \mathfrak{M}, g^{\overline{a}}_{\overline{v}} \right\rangle$ *with* $g^{\overline{a}}_{\overline{v}}$ *the* $\overline{v}$*-variant assignment* $\left(\left(\left(g^{a_1}_{v_1}\right)^{a_2}_{v_2}\right)\cdots\right)^{a_n}_{v_n}$.
   - $(\gamma(\tau_1, \ldots, \tau_n))^{\mathfrak{I}} : W \longrightarrow \mathsf{D}_0$,
     $(\gamma(\tau_1, \ldots, \tau_n))^{\mathfrak{I}}(w) = \gamma^{\mathfrak{I}}(w)(\tau_1^{\mathfrak{I}}(w), \ldots, \tau_n^{\mathfrak{I}}(w)), \gamma \in \mathsf{ME}_n$
   - $(@_i\gamma)^{\mathfrak{I}} : W \longrightarrow \mathsf{D}_n$*, where* $(@_i\gamma)^{\mathfrak{I}}(w) = \gamma^{\mathfrak{I}}(w^i)$ *for any* $w \in W$.
2. *Propositional terms. Let* $w \in W$ *be arbitrary*
   - $(X_\alpha)^{\mathfrak{I}} : W \longrightarrow \mathsf{D}_\alpha$, $(X_\alpha)^{\mathfrak{I}}(w) = g(X_\alpha)$*, for any* $X_\alpha \in \mathsf{VAR}_\alpha$, $\alpha \in \mathsf{PT} - \{\mathsf{t}\}$.
   - $(\lambda X_\alpha A_\beta)^{\mathfrak{I}} : W \longrightarrow \mathsf{D}_{\langle \alpha\beta \rangle}$*, with* $(\lambda X_\alpha A_\beta)^{\mathfrak{I}}(w) : \mathsf{D}_\alpha \longrightarrow \mathsf{D}_\beta$*, mapping* χ *to* $(A_\beta)^{\mathfrak{I}^{\chi}_{X_\alpha}}(w)$.

- $\left(A_{\langle\alpha\beta\rangle}B_\alpha\right)^{\Im} : W \longrightarrow \mathsf{D}_\beta$, $\left(A_{\langle\alpha\beta\rangle}B_\alpha\right)^{\Im}(w) = \left((A_{\langle\alpha\beta\rangle})^{\Im}(w)\right)\left(B_\alpha^{\Im}(w)\right)$, *for* $\beta \neq t$
- $(@_i A_\alpha)^{\Im} : W \longrightarrow \mathsf{D}_\alpha$, *where* $(@_i A_\alpha)^{\Im}(w) = (A_\alpha)^{\Im}(w^i)$ *for any* $w \in W$ *and* $\alpha \in \mathsf{PT} - \{\mathsf{t}\}$.

3. *Formulas. Let* $w \in W$ *be arbitrary*
   - $(X_t)^{\Im} : W \longrightarrow \mathsf{D}_t$, $(X_t)^{\Im}(w) = g(X_t)$, *for any* $X_t \in \mathsf{VAR}_t$.
   - $i^{\Im} = I(i)$.
   - $(A_{\langle\alpha t\rangle}B_\alpha)^{\Im} : W \longrightarrow \mathsf{D}_t$, $\left(A_{\langle\alpha t\rangle}B_\alpha\right)^{\Im}(w) = \left((A_{\langle\alpha t\rangle})^{\Im}(w)\right)\left(B_\alpha^{\Im}(w)\right)$
   - $(F_a \equiv G_a)^{\Im} : W \longrightarrow \mathsf{D}_t$ *mapping any* w *to* T *if and only if* $F_a^{\Im}(w) = G_a^{\Im}(w), a \in \mathsf{AT} \cup \mathsf{PT} - \{0\}$.
   - $(\Box\varphi)^{\Im} : W \longrightarrow \mathsf{D}_t$ *mapping* w *to* T *if for all* $w' \in W$ *such that* $(w, w') \in R$ *then* $\varphi^{\Im}(w') = T$ *and to* F *otherwise.*
   - $(@_i\varphi)^{\Im} : W \longrightarrow \mathsf{D}_t$ *mapping* w *to* T *if* $\varphi^{\Im}(w^i) = T$ *and to* F *otherwise.*

Definition 18 (Validity, consequence and tautology).

1. *We say that a formula* A_t *is* true at the world w *under the interpretation* $\Im$, *if* $(A_t)^{\Im}(w) = T$.
2. *A formula* A_t *is* valid, *if for each interpretation* $\Im$ *and any* $w \in W$, *we have that* $(A_t)^{\Im}(w) = T$. *As usual, we write,* $\models A_t$.
3. *A formula* A_t *is said to be a* ***local consequence*** *of a set of formulas* Γ, *in symbols* $\Gamma \models A_t$ *if for each interpretation* $\Im$ *and any* $w \in W$, $(A_t)^{\Im}(w) = T$ *whenever* $(\Gamma)^{\Im}(w) = T$ *(i.e.,* $(B_t)^{\Im}(w) = T$, *for any* $B_t \in \Gamma$*).*
4. *A formula* A_t *is said to be a* ***global consequence*** *of a set of formulas* Γ, *in symbols* $\Gamma \models_{Glob} A_t$ *if* A_t *is valid whenever* $(\Gamma)^{\Im}(w) = T$ *for each interpretation* $\Im$ *and any* $w \in W$.
5. *A* P-formula *(definition 14) which is valid is called a* tautology.

Lemma 19. *Let* A_α *be any strictly propositional expression (Definition 13) and* $\Im = \langle\mathfrak{M}, g\rangle$ *an interpretation, then*

$$(A_\alpha)^{\Im}(w) = \mathsf{V}(A_\alpha, g_{\mathsf{PT}})$$

where w *is any element of* W *and* g_{PT} *is the restriction of the assignment* g *to variables of propositional types.* $\mathsf{V}(A_\alpha, g_{\mathsf{PT}})$ *is the classical evaluation as defined 2.1, which corresponds to the definition offered in Henkin [10] page 326.*

Proof. The proof can be done by induction. ■

Lemma 20 (Coincidence Lemma for nominals). *Let* $\mathfrak{I} = \langle \mathfrak{M}, g \rangle$ *and* $\mathfrak{I}^* = \langle \mathfrak{M}^*, g \rangle$ *be two interpretations such that* $\mathfrak{M} = \langle W, R, \mathcal{A}, \mathfrak{PT}, I \rangle$ *and* $\mathfrak{M}^* = \langle W, R, \mathcal{A}, \mathfrak{PT}, I^* \rangle$. *I agress with* I^* *for all arguments except, possibly, the nominal* i. *Let* F_a *be a meaningful expression of type* a *s.t.* i *does not occur in* F_a. *Then*

$$(F_a)^{\mathfrak{I}} = (F_a)^{\mathfrak{I}^*}$$

for any $a \in \mathsf{PT} \cup \mathsf{AT}$.

Proof. Straightforward. ■

Lemma 21 (Coincidence Lemma for variables). *Let* F_a *be a meaningful expression of type* a *and* V_b *a variable such that* $V_b \notin \mathsf{Free}(F_a)$ *and* $\mathfrak{I}$ *an interpretation. Then*

$$(F_a)^{\mathfrak{I}} = (F_a)^{\mathfrak{I}^{\theta}_{V_b}}$$

for any $a \in \mathsf{PT} \cup \mathsf{AT}$, $b \in \mathsf{PT} \cup \{0\}$ *and* $\theta \in \mathsf{D}_a$.

Proof. Straightforward by induction on the construction of the meaninful expressions. ■

3.3. Variables, substitution, and rigidity

Next we define one of the paper's key concept, that of ***rigid expressions***. These expressions are introduced recursively and we will prove that they have the same value at all worlds. Good examples 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.

Definition 22 (Rigid meaningful expressions). *The set* RIGIDS *of* ***rigid meaningful expressions*** *is defined inductively as follows:*

$$\mathsf{RIGIDS} ::= V_a \,|\, @_i F_b \,|\, \lambda \overline{v} \tau \,|\lambda X_\alpha A_\beta \,|\, \gamma(\sigma_1, \dots, \sigma_n) \,|\, B_{\langle \alpha\beta \rangle} A_\beta \,|\, G_d \equiv H_d$$

where $a \in \mathsf{PT} \cup \{0\}$, $b \in \mathsf{AT} \cup \mathsf{PT}$, $\alpha \in \mathsf{PT}$, $\beta \in \mathsf{PT}$, $d \in \mathsf{AT} \cup \mathsf{PT} - \{0\}$ *; and with* $F_b, \tau, B_{\langle \alpha\beta \rangle}, A_\beta, G_d, H_d, \gamma, \sigma_i$ *being* RIGIDS.

Remark 23. *Previous definition introduces the set* RIGIDS *of rigids expressions by recursion. Expressions of the form* $@_i F_a$ *are called* rigidified expressions. *Next lemma proves that all the elements in* RIGIDS *receives a rigid interpretation, namely the interpretation is independent of the evaluation's* *world. In particular, all strictly propositional expressions (Definition 13) are rigids.*

LEMMA 24 (Rigids are rigid). *Let $\mathfrak{I}$ be a an interpretation. If $A \in \mathsf{RIGIDS}$ then $(A)^{\mathfrak{I}}(w) = (A)^{\mathfrak{I}}(w')$ for all $w, w' \in W$.*

PROOF. By induction on the construction of rigid expressions.

We give the case for: $\lambda v_1 \cdots v_n \tau$ with $\tau \in \mathsf{RIGIDS}$. Let $\mathfrak{I} = \langle \mathfrak{M}, g \rangle$ and $w, w' \in W$. .

$(\lambda v_1 \cdots v_n \tau)^{\mathfrak{I}}(w)$ and $(\lambda v_1 \cdots v_n \tau)^{\mathfrak{I}}(w')$ are extensional functions.

By definition, for any $(a_1, \ldots, a_n) \in (\mathsf{D}_0)^n$:

- $(\lambda v_1 \cdots v_n \tau)^{\mathfrak{I}}(w) = \tau^{\mathfrak{I}^{a_1,\ldots,a_n}_{v_1,\ldots,v_n}}(w)$
- $(\lambda v_1 \cdots v_n \tau)^{\mathfrak{I}}(w') = \tau^{\mathfrak{I}^{a_1,\ldots,a_n}_{v_1,\ldots,v_n}}(w')$

Using the induction hypothesis for τ, we get $\tau^{\mathfrak{I}^{a_1,\ldots,a_n}_{v_1,\ldots,v_n}}(w) = \tau^{\mathfrak{I}^{a_1,\ldots,a_n}_{v_1,\ldots,v_n}}(w')$, and thus $(\lambda v_1 \cdots v_n \tau)^{\mathfrak{I}}(w) = (\lambda v_1 \cdots v_n \tau)^{\mathfrak{I}}(w')$ ■

DEFINITION 25 (Variable substitution). *For all $F_a \in \mathsf{ME}_a$, $b \in \mathsf{PT} \cup \{0\}$ the* ***substitution of*** *G_b* ***for a variable*** *V_b* ***in*** *F_a, written $F_a \frac{G_b}{V_b}$, is defined by induction in the usual way, the only binder operator is lambda abstractor.*

For example:

$$(\lambda X_\rho F_a)\frac{G_b}{V_b} = \begin{cases} \lambda X_\rho F_a & \text{, if } V_b \notin \mathsf{Free}(\lambda X_\rho F_a) \\ \lambda X_\rho (F_a \frac{G_b}{V_b}) & \text{, if } V_b \in \mathsf{Free}(\lambda X_\rho (F_a)) \text{ and } X_\rho \notin \mathsf{Free}(G_b) \\ (\lambda Y_\rho (F_a \frac{Y_\rho}{X_\rho}))\frac{G_b}{V_b} & \text{, if } V_b \in \mathsf{Free}(\lambda(X_\rho F_a)),\ X_\rho \in \mathsf{Free}(G_b) \text{ and } Y_\rho \text{ new} \end{cases}$$

Rigid expressions are well-behaved with respect to substitution, as stated in the following lemma

LEMMA 26 (Rigid substitution). *Let $\mathfrak{I}$ be an interpretation and w a world on it:*

$$\left(F_a \frac{G_b}{V_b}\right)^{\mathfrak{I}}(w) = (F_a)^{\mathfrak{I}^{(G_b)^{\mathfrak{I}}(w)}_{V_b}}(w)$$

for all F_a basic expression of type $a \in \mathsf{AT} \cup \mathsf{PT}$, $G_b \in \mathsf{RIGIDS}_b$ and V_b any variable of type $b \in \mathsf{PT} \cup \{0\}$.

PROOF. Straightforward by induction on the construction of meaningful expressions, with the help of *Coincidence Lemma* 21. ■

COROLLARY 27. *Let B_t be a formula.*

$$\text{If } \models B_t \text{ then } \models B_t \frac{A^1_{\alpha_1}, \ldots, A^n_{\alpha_n}}{X^1_{\alpha_1}, \ldots, X^n_{\alpha_n}}$$

provided that the variables $X^1_{\alpha_1}, \ldots, X^n_{\alpha_n}$ are distinct from one another and that $A^i_{\alpha_i}$ is rigid and free for $X^i_{\alpha_i}$ in φ for all $i = 1, \ldots, n$.

3.4. Nameability in EHPTT

As a consequence of our Lemma 19 we obtain nameability for the strict propositional fragment of our EHPTT logic. In the first place, our Definition 12 extends the Definition 1 of all common logical operators using lambda and propositional equality to cover the needs of of our equational system. The next theorem, proved in [13], states that all the introduced operators behave as usual. That is, with equality, abstraction, nominals, $\Box$ and @ we can define all basic standard operators needed for equational hybrid propositional type theory.

THEOREM 28. *For every interpretation $\mathfrak{I}$ the following holds:*

1. $\left(T^{\mathsf{N}}\right)^{\mathfrak{I}} : W \longrightarrow \mathsf{D}_t$, *with* $\left(T^{\mathsf{N}}\right)^{\mathfrak{I}}(w) = T$.
2. $\left(F^{\mathsf{N}}\right)^{\mathfrak{I}} : W \longrightarrow \mathsf{D}_t$, *with* $\left(F^{\mathsf{N}}\right)^{\mathfrak{I}}(w) = F$.
3. $(\neg)^{\mathfrak{I}} : W \longrightarrow \mathsf{D}_t^{\mathsf{D}_t}$ *such that* $(\neg)^{\mathfrak{I}}(w)$ *is the Boolean "negation".*
4. $(\wedge)^{\mathfrak{I}} : W \longrightarrow \mathsf{D}_{tt}^{\mathsf{D}_t}$ *such that* $(\wedge)^{\mathfrak{I}}(w)$ *is the Boolean "conjunction".*
5. $(\forall X_\alpha A_t)^{\mathfrak{I}} : W \longrightarrow \mathsf{D}_t$, *mapping w to T only if* $(A_t)^{\mathfrak{I}^x_{X_\alpha}}(w) = T$ *for all* $x \in D_\alpha$.
6. $(\tau \approx \sigma)^{\mathfrak{I}} : W \longrightarrow \mathsf{D}_t$ *mapping w to T iff* $\left\{\bar{a} \mid \tau^{\mathfrak{I}^{\bar{a}}_{\bar{v}}}(w) = \sigma^{\mathfrak{I}^{\bar{a}}_{\bar{v}}}(w)\right\} = \mathcal{A}^n$, *where* $\mathsf{Free}(\tau) \cup \mathsf{Free}(\sigma) \subseteq \{v_1, \ldots, v_n\} \neq \varnothing$ *and* $\bar{v} = \langle v_1, \ldots, v_n \rangle$.

In the sequel we will use the usual quantifiers as well as the connectives $\bot, \neg, \vee, \wedge, \rightarrow$ and $\leftrightarrow$ as abreviations. Recall that $\leftrightarrow$ is a special case of $\equiv$ for type t.

The Nameability Theorem also applies for EHPTT:

THEOREM 29 (Nameability Theorem). *We shall associate, with each element χ of an arbitrary type D_α, a closed formula χ^{N} of type α such that $(\chi^{\mathsf{N}})^{\mathfrak{I}}(w) = \chi$ for any interpretation $\mathfrak{I}$ and world w.*

PROOF. The proof is obvious using Lemma 19 and Henkin's Nameability Theorem[8]. ■

4. Rules and Axioms (EHPTT)

Rules

[8] The proof is in [10], section §4, pages 326-3229.

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) **Pseudo-Gen$_{Alg}$:** If $\Delta \vdash \gamma(@_{k_1}c_1, \ldots, @_{k_n}c_n) \approx \delta(@_{k_1}c_1, \ldots @_{k_n}c_n)$ then $\Delta \vdash \gamma \equiv \delta$, where $\gamma, \delta \in \mathsf{ME}_n$ and $c_1, \ldots, c_n \in \mathsf{CON}_0$ do not occur in Δ.
3. (a) **Rigid Substitution/Replacement:** If $\vdash \varphi$ then $\vdash \varphi'$, where φ' is obtained from φ by uniformly replacing nominals by nominals, or $\varphi' := \varphi\frac{A^1_{\alpha_1},\ldots,A^n_{\alpha_n}}{X^1_{\alpha_1},\ldots,X^n_{\alpha_n}}$ provided that $A^i_{\alpha_i}$ is rigid and variables $X^i_{\alpha_i}$ are distinct from one another and free in φ for all $i = 1, \ldots, n$.
   (b) **Henkin's Rule:** Let φ, A_α and B_α be strictly propositional. From φ and $(A_\alpha \equiv B_\alpha)$ to infer the result of replacing one occurrence of A_α in φ by an occurrence of B_α, provided that the occurrence of A_α in φ is not (an occurrence of a variable) immediately preceded by λ.
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 [6]. 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 EHPTT instances of tautologies [9].
2. **Henkin PTT Axiom 4:** $\vdash (g_{\langle t,t\rangle}T^{\mathsf{N}} \wedge g_{\langle t,t\rangle}F^{\mathsf{N}}) \equiv (\forall X_t(g_{\langle t,t\rangle}X_t))$
3. **Distributivity Axioms:**

[9] An EHPTT instance of a tautology is any EHPTT formula which is obtained by a substitution of propositional variables by formulas in a tautology.

   (a) **$\Box$-distributivity:** $\vdash \Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi)$
   (b) **@-distributivity:** $\vdash @_i(\varphi \to \psi) \to (@_i\varphi \to @_i\psi)$
4. **Quantifier Axiom:**
   $\vdash (\exists X_\alpha \varphi) \equiv (\varphi\frac{\chi_1^{\mathsf{N}}}{X_\alpha} \vee ... \vee \varphi\frac{\chi_n^{\mathsf{N}}}{X_\alpha})$, where φ is rigid and $\mathsf{D}_\alpha = \{\chi_1, \ldots, \chi_n\}$
5. **Equality Axioms:**
   (a) **Reflexivity:** $\vdash F_a \equiv F_a$, where $a \in \mathsf{PT} \cup \mathsf{AT} - \{0\}$
   (b) **Symmetry:** $\vdash F_a \equiv G_a \to G_a \equiv F_a$, where $a \in \mathsf{PT} \cup \mathsf{AT} - \{0\}$
   (c) **Transitivity:** $\vdash (F_a \equiv G_a \wedge G_a \equiv H_a) \to F_a \equiv H_a$, $a \in \mathsf{PT} \cup \mathsf{AT} - \{0\}$
   (d) **Substitution Prop:** $\vdash A_\alpha \equiv B_\alpha \to F_\beta\frac{A_\alpha}{X_\alpha} \equiv F_\beta\frac{B_\alpha}{X_\alpha}$, wiht $\alpha\beta \in \mathsf{PT}$ and A_α, B_α are rigids.
   (e) **Equality-at-i:** $\vdash @_i(F_a \equiv G_a) \equiv (@_iF_a \equiv @_iG_a)$
6. **Functional Axioms:**
   (a) **Extensionality:**
      i. **Prop:** $\vdash (\forall X_\alpha(A_{\langle\alpha\beta\rangle}X_\alpha \equiv B_{\langle\alpha\beta\rangle}X_\alpha)) \to A_{\langle\alpha\beta\rangle} \equiv B_{\langle\alpha\beta\rangle}$, X_α does not occur free in $A_{\langle\alpha\beta\rangle}$ or $B_{\langle\alpha\beta\rangle}$.
      ii. **Alg:** $\vdash \gamma(v_1, \ldots, v_n) \approx \delta(v_1, \ldots, v_n) \to \gamma \equiv \delta$
   (b) **β-conversion:**
      i. **Prop:** For rigid B_β, $\vdash (\lambda X_\beta A_\alpha)B_\beta \equiv A_\alpha\frac{B_\beta}{X_\beta}$
      ii. **Alg:** For rigids $\sigma_1, \ldots, \sigma_n$, $\vdash (\lambda\bar{v}\tau)(\sigma_1, \ldots, \sigma_n) \approx \tau\frac{\sigma_1,\ldots,\sigma_n}{v_1,\ldots,v_n}$
   (c) **λ-conversion:**
      i. **Prop:** $\vdash (\lambda X_\beta A_{\langle\alpha\beta\rangle}X_\beta) \equiv A_{\langle\alpha\beta\rangle}$, where X_β is not free in $A_{\langle\alpha\beta\rangle}$
      ii. **Alg:** $\vdash (\lambda\bar{v}\tau)(v_1, \ldots, v_n) \approx \tau$, where v_i are not free in $\tau \in \mathsf{ME}_0$
   (d) **Algebraic functionality**
      i. $\vdash (\tau_1 \approx \tau_1' \wedge \cdots \wedge \tau_n \approx \tau_n') \to \gamma(\tau_1, \ldots, \tau_n) \approx \gamma(\tau_1', \ldots, \tau_n')$, $\gamma \in \mathsf{ME}_n$ and $\tau_i, \tau_i' \in \mathsf{ME}_0$ for $i = 1, \ldots, n$
      In primitive symbols: $\vdash (\lambda\bar{x}_1\tau_1 \equiv \lambda\bar{x}_1\tau_1' \wedge \cdots \wedge \lambda\bar{x}_n\tau_n \equiv \lambda\bar{x}_n\tau_n') \to \lambda\bar{x}_1 \ldots \bar{x}_n\gamma(\tau_1, \ldots, \tau_n) \equiv \lambda\bar{x}_1 \ldots \bar{x}_n\gamma(\tau_1', \ldots, \tau_n')$
      ii. $\vdash \gamma \equiv \delta \to \gamma(\tau_1, \ldots, \tau_n) \approx \delta(\tau_1, \ldots, \tau_n)$ $\gamma, \delta \in \mathsf{ME}_n$ and $\tau_i \in \mathsf{ME}_0$ for $i = 1, \ldots, n$
      In primitive symbols: $\vdash \gamma \equiv \delta \to \lambda\bar{v}\gamma(\tau_1, \ldots, \tau_n) \equiv \lambda\bar{v}\delta(\tau_1, \ldots, \tau_n)$
   (e) **Rigid function application:**
      i. $\vdash @_i(A_{\langle\alpha\beta\rangle}B_\alpha) = (@_iA_{\langle\alpha\beta\rangle})(@_iB_\alpha)$
      ii. $\vdash @_i(\gamma(\tau_1, \ldots, \tau_n)) \approx (@_i\gamma)(@_i\tau_1, \ldots, @_i\tau_n)$, $\gamma \in \mathsf{ME}_n$ and $\tau_i \in \mathsf{ME}_0$ for $i = 1, \ldots, n$
7. **Axioms for @:**
   (a) **Selfdual:** $\vdash @_i\varphi \equiv \neg @_i\neg\varphi$

(b) **Intro:** $\vdash i \to (\varphi \equiv @_i\varphi)$
(c) **Back:** $\vdash \Diamond @_i\varphi \to @_i\varphi$
(d) **Ref:** $\vdash @_i i$
(e) **Agree:**
  i. $\vdash @_i @_j F_a \equiv @_j F_a$
  ii. $\vdash @_i @_j \tau \approx @_j \tau$
(f) **Rigids are rigid:**
  i. $\vdash @_i F_a \equiv F_a$, where F_a is rigid.
  ii. $\vdash @_i \tau \approx \tau$, where τ is rigid.

8. **Barcan Axioms:**
(a) **Prop:** $\vdash \lambda X_\alpha @_i A_\beta \equiv @_i \lambda X_\alpha A_\beta$
(b) **Alg:** $\vdash \lambda \bar{v} @_i \tau \equiv @_i \lambda \bar{v} \tau$

REMARK 30. *This set of axioms exhibits that the combination of the three logics (propositional type theory, equational logic and hybrid modal logic) is not obtained just by putting together the axioms of each one. There are axioms that come from a specific logic (e.g.,* Henkin PTT axiom 4*, distributive axioms, algebraic functionality, etc), as well as axioms that integrate attributes from more than one logic (λ-conversion axioms, equality axioms, etc.). There are also others that are just reformulations for the combined language (e.g., tautologies).*
One interesting axiom is the quantifier axiom that substantiates the importance of the Henkin's nameability and the finiteness of D_α in the hierarchy of propositional types. The Barcan axioms are also well known in first order modal logic, they are connected with the fact that in our semantics the algebras at each world are over the same set A, that is, we are dealing with constants domains.
Since our semantics is intensional, some axioms have to be restricted to rigid (or rigidified) terms and/or formulas.

DEFINITION 31. *A **deduction** of φ is a finite sequence $\xi_1, \ldots, \xi_n$ of expressions in ME_t such that $\xi_n := \varphi$ and for every $1 \leq i \leq n$, either:*
– ξ_i is an axiom or
– ξ_i is obtained from previous expressions in the sequence using the rules.

We will write $\vdash \varphi$ whenever we have such a sequence and we will say that φ is an EHPTT*-**theorem** of the calculus.*

*If $\Gamma \cup \{\varphi\}$ is a set of meaningful expressions of type t, a **deduction of** φ **from** Γ — written $\Gamma \vdash \varphi$ — is a deduction of $\gamma_1 \wedge \cdots \wedge \gamma_n \to \varphi$ where for every $1 \leq i \leq n$, $\gamma_i \in \Gamma$.*

4.1. EHPTT contains PTT

It is easy to see that the calculus of our EHPTT logic contains that of Henkin's PTT.

THEOREM 32. *If* $\vdash_{\mathsf{PTT}} A_t$ *then* $\vdash_{\mathsf{EHPTT}} A_t$ *for any* A_t *formula of* PTT.

PROOF. In our calculus for EHPTT we include as axioms the numbers 1, 4, 6 and 7 of Henkin's calculus. They are our Axioms for **Reflexivity of equality** (5a), **Henkin PTT Axiom 4** (2), **Extensionality** (6a) and **β-conversion** (6(b)i). Henkin's rule of replacement is just a rule in our calculus (**Henkin's Rule**).

The proofs of Henkin's Axioms 2, 3 and 5 in our EHPTT calculus are in the appendix (Theorems 68, 70 and 69). ■

5. Maximal consistent sets

In this section we define and explore maximal consistent sets of EHPTT sentences with various useful properties, prove the variant of Lindenbaum's Lemma we shall require and then introduce several equivalence relations using a saturated maximal consistent set.

DEFINITION 33. *A set* $\Delta \subseteq \mathsf{ME}_t$ *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 lemmas state some well known consequences of the definitions and rules of the calculus.

LEMMA 34. *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 35. *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. $\varphi \wedge \psi \in \Delta$ *iff* $\varphi \in \Delta$ *and* $\psi \in \Delta$.
5. $\varphi \vee \psi \in \Delta$ *iff* $\varphi \in \Delta$ *or* $\psi \in \Delta$.
6. $\varphi \equiv \psi \in \Delta$ *iff* $(\varphi \in \Delta$ *and* $\psi \in \Delta)$ *or* $(\varphi \notin \Delta$ *and* $\psi \notin \Delta)$
7. *If* $\Delta \cup \{\varphi\} \vdash \psi$ *and* $\Delta \cup \{\psi\} \vdash \varphi$ *then* $\varphi \in \Delta$ *iff* $\psi \in \Delta$.

5.1. Maximal consistent, named, $\Diamond-$saturated and extensionally algebraic-saturated sets

Definition 36. *Let* Γ *be a set of sentences.*

1. Γ *is* named *iff one of its elements is a nominal.*
2. Γ *is* $\Diamond$-saturated *iff for all expressions* $@_i\Diamond\varphi \in \Gamma$ *there is a nominal* $j \in \mathsf{NOM}$ *such that* $@_i\Diamond j \in \Gamma$ *and* $@_j\varphi \in \Gamma$.
3. Γ *is* extensionally algebraic-saturated *if for all expressions* $@_i\gamma \equiv @_j\delta$, *with* $\gamma, \delta \in \mathsf{ME}_n$, *there are rigid terms* $@_{k_1}\tau_1, \ldots, @_{k_n}\tau_n \in \mathsf{ME}_0$ *such that*

$$@_i\gamma(@_{k_1}\tau_1, \ldots, @_{k_n}\tau_n) \approx @_j\delta(@_{k_1}\tau_1, \ldots, @_{k_n}\tau_n) \rightarrow @_i\gamma \equiv @_j\delta \in \Gamma$$

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 37 (Extended Lindenbaum). *Let* Γ *be a consistent set of sentences. Then* Γ *can be extended to a maximal consistent set* Γ^ω *which is named,* $\Diamond$*-saturated and extensionally algebraic-saturated.*

Proof. Let $\{i_n\}_{n\in\omega}$ be an enumeration of a countably infinite set of new nominals, $\{c_n\}_{n\in\omega}$ an enumeration of a countably infinite set of new constants of type 0, and $\{\varphi_n\}_{n\in\omega}$ an enumeration of the sentences of the extended language. We will build $\{\Gamma^n\}_{n\in\omega}$, a family of subsets of ME_t, by induction:

- $\Gamma^0 = \Gamma \cup \{i_0\}$.
- Assume that Γ^n has already been built. To define Γ^{n+1} we distinguish four cases:
  1. $\Gamma^{n+1} = \Gamma^n$, if $\Gamma^n \cup \{\varphi_n\}$ is inconsistent.

2. $\Gamma^{n+1} = \Gamma^n \cup \{\varphi_n\}$, if $\Gamma^n \cup \{\varphi_n\}$ is consistent and φ_n is neither of the form $@_i\Diamond\psi$ nor $\neg(@_i\gamma \equiv @_j\delta)$.
3. $\Gamma^{n+1} = \Gamma^n \cup \{\varphi_n, @_i\Diamond i_m, @_{i_m}\psi\}$, if $\Gamma^n \cup \{\varphi_n\}$ is consistent, $\varphi_n := @_i\Diamond\psi$ and i_m is the first nominal not in Γ^n or φ_n.
4. $\Gamma^{n+1} =$
$\Gamma^n \cup \{\varphi_n, \neg(@_i\gamma(@_{k_1}c_1, \dots, @_{k_m}c_m) \approx @_j\delta(@_{k_1}c_1, \dots, @_{k_m}c_m))\}$,
if $\Gamma^n \cup \{\varphi_n\}$ is consistent, $\varphi_n := \neg(@_i\gamma \equiv @_j\delta)$ and $c_1, \dots, c_m$ are the first constants of type 0 not in Γ^n or φ_n and $k_1, \dots, k_n$ are the first nominals not in Γ^n or φ_n.

First we show by induction that each Γ^n is consistent. For the base case, suppose Γ^0 is inconsistent. Hence $\Gamma \cup \{i_0\} \vdash \bot$, then, by Theorem 54, $\Gamma \vdash i_0 \to \bot$ and by Name$'^*$ (Theorem 59), $\Gamma \vdash \bot$, which is impossible.

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

1. $\Gamma^{n+1} = \Gamma^n$ is consistent by the induction hypothesis.
2. $\Gamma^{n+1} = \Gamma^n \cup \{\varphi_n\}$ is consistent by construction.
3. Suppose $\Gamma^{n+1} = \Gamma^n \cup \{\varphi_n, @_i\Diamond i_m, @_{i_m}\psi\}$, where $\varphi_n := @_i\Diamond\psi$ and i_m is the first new nominal that does not occur in Γ^n or φ_n. By construction, $\Gamma^n \cup \{\varphi_n\}$ is consistent. Suppose that $\Gamma^n \cup \{\varphi_n, @_i\Diamond i_m, @_{i_m}\psi\} \vdash \bot$. Then, $\Gamma^n \cup \{\varphi_n\} \vdash @_i\Diamond i_m \wedge @_{i_m}\psi \to \bot$, hence $\Gamma^n \cup \{\varphi_n\} \vdash @_i\Diamond\psi \to \bot$, by using Paste$^*_\Diamond$ (Theorem 59) and the fact that $i_m \neq i$ and i_m does not occur in ψ or $\bot$. Thus, $\Gamma^n \cup \{\varphi_n\} \vdash \bot$, which is impossible.
4. Suppose
$\Gamma^{n+1} = \Gamma^n \cup \{\varphi_n, \neg(@_i\gamma(@_{k_1}c_1, \dots, @_{k_m}c_m) \approx @_j\delta(@_{k_1}c_1, \dots, @_{k_m}c_m))\}$,
where $\varphi_n := \neg(@_i\gamma \equiv @_j\delta)$ and $c_1, \dots, c_m$ are the first constants of type 0 not in Γ^n or φ_n and $k_1, \dots, k_m$ are the first nominals not in Γ^n or φ_n. By construction, $\Gamma^n \cup \{\varphi_n\}$ is consistent. Suppose that
$\Gamma^n \cup \{\varphi_n, \neg(@_i\gamma(@_{k_1}c_1, \dots, @_{k_m}c_m) \approx @_j\delta(@_{k_1}c_1, \dots, @_{k_m}c_m))\} \vdash \bot$
Then,
$\Gamma^n \cup \{\varphi_n\} \vdash (\neg(@_i\gamma(@_{k_1}c_1, \dots, @_{k_m}c_m) \approx @_j\delta(@_{k_1}c_1, \dots, @_{k_m}c_m)) \to \bot$.
Hence, $\Gamma^n \cup \{\varphi_n\} \vdash @_i\gamma(@_{k_1}c_1, \dots, @_{k_m}c_m) \approx @_j\delta(@_{k_1}c_1, \dots, @_{k_m}c_m)$.
Therefore, by the rule Pseudo-Gen$_{Alg}$, we have $\Gamma^n \cup \{\varphi_n\} \vdash @_i\gamma \equiv @_j\delta$ and, consequently, $\Gamma^n \cup \{\varphi_n\} \vdash \bot$, which is impossible.
Now, let $\Gamma^\omega = \bigcup_{n\in\omega}\Gamma^n$. Γ^ω is maximal consistent. Moreover, it is named, $\Diamond$-saturated and extensionally algebraic-saturated. ■

5.2. Equivalence relations using a saturated maximal consistent set

Let Δ be a maximal consistent named, $\Diamond$-saturated and extensionally algebraic-saturated. We define an equivalence relation on the set NOM of nominals.

DEFINITION 38. *Let Δ be a maximal consistent named, $\Diamond$-saturated and extensionally algebraic-saturated. Define for all $i, j \in \mathrm{NOM}$ the relation $\sim$:*

$$i \sim j \textit{ iff } @_i j \in \Delta$$

PROPOSITION 39. *The relation $i \sim j$ is an equivalence on the set* NOM *of nominals.*

PROOF. It is trivial to prove that $\sim$ is an equivalence relation by using the theorems of maximal consistency and applying reflexivity (Axiom 7d) and theorems of symmetry (Theorem 63) and transitivity (Theorem 64). ■

And then, we define the equivalence class $[i] = \{j \in \mathsf{NOM} : i \sim j\}$.

DEFINITION 40. *Let Δ be a maximal consistent which is named, $\Diamond$ saturated and extensionally algebraic-saturated. The accessibility relation is a binary relation on W defined by*

$$R = \{\langle [i], [j] \rangle \mid @_i \Diamond j \in \Delta\},$$

where $W = \{[i] \mid i \text{ is a nominal}\}$.

PROPOSITION 41. *R is well defined.*

PROOF. To prove that the definition is independent of the representatives, we use Theorem 60 (**Nom**) and Theorem 66, the definition of the equivalence relation and that Δ is maximal consistent. ■

Next, we define another equivalence relation $\sim$ for expressions of algebraic types.

DEFINITION 42. *Let Δ be a maximal consistent which named, $\Diamond$ saturated and extensionally algebraic-saturated. We define a relation $\sim$ on rigidified algebraic expressions of any type n:*

- *Equivalence relation on type 0: $@_i\tau \sim @_j\tau'$ iff $@_i\tau \approx @_j\tau' \in \Delta$, for all rigid terms of the form $@_i\tau$ or $@_j\tau'$.*

- *Equivalence relation on type $n \neq 0$: $@_i\gamma \sim @_j\gamma'$ iff $@_i\gamma \equiv @_j\gamma' \in \Delta$, for all rigid terms of the form $@_i\gamma$ or $@_j\gamma'$.*

PROPOSITION 43. *The relation $\sim$ is an equivalence on the set of rigid algebraic expressions of any type n.*

PROOF. In both cases, it is trivial to prove that $\sim$ is indeed an equivalence relation. For terms of type $n \neq 0$, it follows from Axioms 5a, 5b and 5c. For type 0 we do not need to prove that the relation $\sim$

$$@_i\tau \sim @_j\tau' \text{ iff } @_i\tau \approx @_j\tau' \in \Delta$$

is an equivalence relation, as equations of this form $@_i\tau \approx @_j\tau'$ are defined as $\lambda\bar{v}@_i\tau \equiv \lambda\bar{v}@_j\tau'$. Using Barcan Axiom 8 we see that this very relation could also be defined with $@_i\lambda\bar{v}\tau \equiv @_j\lambda\bar{v}\tau'$ and we have already proved that the relation $@_i\lambda\bar{v}\tau \sim @_j\lambda\bar{v}\tau'$ is an equivalence. ■

And then, we define the equivalence classes for all algebraic types. In the next section we will build a model using the set of all classes of rigidified terms of type 0 as the domain of individuals.

6. Completeness

In this section we will build a structure using the information in the set Δ which is maximal consistent, named, $\Diamond$-saturated and extensionally algebraic-saturated. The domain W and the relation R are defined (Definition 40) in the usual way based on the equivalence relation $\sim$ on nominals (Definition 38).

To define the domain of individuals and the hierarchy of propositional types, as well as the interpretation of the non-logical constants, we define a function Φ acting on expressions of the form $@_iF$ and, simultaneously, we define the domains D^*_a for each type, even for propositional types. In this particular case, instead of taking the standard hierarchy as it, we build it through the names of the objects in the hierarchy. These names play a very relevant role, similar to the one played in Henkin's original proof. The decision was made in order to easy our final objective, namely, the Completeness Theorem.

In essence, we shall take equivalence classes as elements of the domains, but for type $\langle\alpha\beta\rangle$ we need functions from D^*_α to D^*_β. So we define Φ as a map which corresponds, in a proper sense, to the equivalence classes.

6.1. Definition of the function Φ and associated domains.

Function Φ for algebraic types. On the first place we define a function Φ on rigidified algebraic expressions, as well as the domain of individuals of the Δ-model to be defined afterwards.

DEFINITION 44 (**Function Φ for algebraic types**). *Let* $D_0^* = \{[@_i\tau] : \tau \in \mathsf{ME}_0, i \in \text{NOM}\}$*. The equivalence relation used in this definition is the one introduced at the end of the previous section, namely, Definition 42.*

- *Type* 0. $\Phi(@_i\tau) = [@_i\tau]$
- *Type* $n \neq 0$.

$$\begin{array}{lccc} \Phi(@_i\gamma): & \mathsf{D}_0^* \times \cdots \times \mathsf{D}_0^* & \longrightarrow & \mathsf{D}_0^* \\ & \langle [@_{i_1}\tau_1], \ldots, [@_{i_n}\tau_n] \rangle & \mapsto & \Phi(@_i\gamma(@_{i_1}\tau_1, \ldots, @_{i_n}\tau_n)) \end{array}$$

LEMMA 45 (Algebraic types). Φ *is well defined for all algebraic types.*

PROOF. **Type** 0. Obviously $\Phi(@_i\tau) = \Phi(@_j\tau')$ iff $@_i\tau \approx @_j\tau' \in \Delta$.

Type n. In order to see that it is well defined we have to show that

1. Φ is independent of the representatives.
2. Φ respects the equivalence relation on type n, i.e.,

$$\Phi(@_i\gamma) = \Phi(@_j\delta) \text{ iff } @_i\gamma \equiv @_j\delta \in \Delta$$

We prove both items for $n = 1$, for the remaining algebraic types it is similar.

- Proof of 1. Let $@_k\tau \approx @_{k'}\tau' \in \Delta$. Then $@_i\gamma(@_k\tau) \approx @_i\gamma(@_{k'}\tau') \in \Delta$, from Axiom 6d and properties of maximal consistent sets.
- Proof of 2.
  (⇒) Let $\Phi(@_i\gamma) = \Phi(@_j\delta)$ and $@_i\gamma \equiv @_j\delta \notin \Delta$. Being Δ extensionally algebraic-saturated, there is a rigidified term $@_k\tau \in \mathsf{ME}_0$ such that $@_i\gamma(@_k\tau) \approx @_j\delta(@_k\tau) \to @_i\gamma \equiv @_j\delta \in \Delta$. Then $\neg @_i\gamma(@_k\tau) \approx @_j\delta(@_k\tau) \in \Delta$. However, $\Phi(@_i\gamma) = \Phi(@_j\delta)$ implies $\Phi(@_i\gamma)([@_k\tau]) = \Phi(@_i\delta)([@_k\tau])$. And, according with our definition

$$\Phi(@_i\gamma)([@_k\tau]) = \Phi(@_i\gamma(@_k\tau)) = [@_i\gamma(@_k\tau)]$$

and

$$\Phi(@_j\delta)([@_k\tau]) = \Phi(@_j\delta(@_k\tau)) = [@_j\delta(@_k\tau)]$$

Then

$$@_i\gamma(@_k\tau) \approx @_j\delta(@_k\tau) \in \Delta$$

Impossible, as Δ is maximal consistent.
($\Leftarrow$) Let $@_i\gamma \equiv @_j\delta \in \Delta$. Then for all rigids $@_k\tau \in \mathsf{ME}_0$, $@_i\gamma(@_k\tau) \approx @_j\delta(@_k\tau) \in \Delta$, by Axiom 6d and properties of maximal consistent sets. Then $\Phi(@_i\gamma) = \Phi(@_j\delta)$.
■

Remark 46. *Obviously,* $\mathsf{D}^*_0 = \{[@_i\tau] : \tau \in \mathsf{ME}_0, i \in \mathrm{NOM}\} = \{\Phi(@_i\tau) : \tau \in \mathsf{ME}_0, i \in \mathrm{NOM}\}$*. The function* Φ *on type* 0 *characterized the universe of individuals. As we do not have variables of type* $n \neq 0$*, we are not defining universes* D^*_n*. On type* n *the function will be used to define the interpretation of functional symbols as the value under* Φ*, guaranteeing that such an interpretation is a* n*-ary function on the domain of individuals.*

Definition of Φ for propositional types.

Lemma 47 (Propositional types). *Given a maximal consistent set* Δ*, which is named,* $\Diamond$ *saturated and extensionally algebraical-saturated, there exists a family of domains* $\langle \mathsf{D}^*_\alpha \rangle_{\alpha \in PT}$ *and a function* Φ *satisfying:*

1. Φ *is well defined over the set* $\mathsf{CME}_\alpha \cap \mathsf{RIGIDS}$ *and so we define:*

$$\mathsf{D}^*_\alpha = \{\Phi(A_\alpha) : A_\alpha \in \mathsf{CME}_\alpha \cap \mathsf{RIGIDS}\}$$

2. Φ *respects the equivalent relation based on* Δ*; namely,* $\Phi(A_\alpha) = \Phi(B_\alpha)$ *iff* $A_\alpha \equiv B_\alpha \in \Delta$ *for all* $A_\alpha, B_\alpha \in \mathsf{CME}_\alpha \cap \mathsf{RIGIDS}$*.*
3. $\chi_\alpha = \Phi(\chi^{\mathsf{N}}_\alpha)$ *for each type* $\alpha \in \mathsf{PT}$ *and any* $\chi_\alpha \in \mathsf{D}_\alpha$ *(where* D_α *are the standard echelons of the hierarchy built on* $\mathsf{D}_t = \{T, F\}$*)*
4. $\mathsf{D}^*_\alpha = \{\Phi(\chi^{\mathsf{N}}_\alpha) : \chi_\alpha \in \mathsf{D}_\alpha\} = \mathsf{D}_\alpha$*.*

Proof. The proof is by induction on propositional types by simultaneously defining the function Φ and the hierarchy $\langle \mathsf{D}^*_\alpha \rangle_{\alpha \in PT}$

Type t. Let us define $\Phi(A_t) = T$ if $A_t \in \Delta$, otherwise $\Phi(A_t) = F$, for all $A_t \in \mathsf{CME}_t \cap \mathsf{RIGIDS}$. Note that, $A_t \notin \Delta$ iff $\neg A_t \in \Delta$.

Let us see that the 4 conditions above are satisfied.

1. Since Δ is maximal consistent, exactly one of this conditions $A_t \in \Delta$ or $\neg A_t \in \Delta$ holds, for all $A_t \in \mathsf{CME}_t \cap \mathsf{RIGIDS}$. That proves the first requirement.

2. ($\Rightarrow$) Let $\Phi(A_t) = \Phi(B_t)$. There are two possibilities: either both are T or both are F.
   If $\Phi(A_t) = T = \Phi(B_t)$ then $A_t \in \Delta$ and $B_t \in \Delta$.
   Hence, $A_t \equiv B_t \in \Delta$, by Lemma 35 on maximal consistent sets.
   If $\Phi(A_t) = F = \Phi(B_t)$ then $A_t \notin \Delta$ and $B_t \notin \Delta$.
   Hence, $A_t \equiv B_t \in \Delta$, by Lemma 35 on maximal consistent sets.
   ($\Leftarrow$) Let $A_t \equiv B_t \in \Delta$. Then either ($A_t \in \Delta$ and $B_t \in \Delta$) or ($A_t \notin \Delta$ and $B_t \notin \Delta$), by Lemma 35 on maximal consistent sets. Thus either $\Phi(A_t) = T = \Phi(B_t)$ or $\Phi(A_t) = F = \Phi(B_t)$, by definition of Φ.

3. $\mathsf{D}_t = \{T, F\}$, according to the standard definition. We need to prove that $\Phi(T^{\mathsf{N}}) = T$ and $\Phi(F^{\mathsf{N}}) = F$. As already defined, $T^{\mathsf{N}} := ((\lambda X_t X_t) \equiv (\lambda X_t X_t))$ and $F^{\mathsf{N}} := ((\lambda X_t X_t) \equiv (\lambda X_t T^n))$. It is obvious that both are in the set $\mathsf{CME}_t \cap \mathsf{RIGIDS}$ as both are strictly propositional (Definition 13) sentences.$T^{\mathsf{N}} \in \Delta$, since $\vdash T^{\mathsf{N}}$, using our Axiom 5a. Hence, $\Phi(T^{\mathsf{N}}) = T$. $F^{\mathsf{N}} \notin \Delta$, since $\vdash F^{\mathsf{N}} \equiv \neg T^{\mathsf{N}}$ (Theorem 67) and Lemma 35 on maximal consistent sets. Therefore , $\Phi(F^{\mathsf{N}}) = F$.

4. Clearly, $\mathsf{D}^*_t = \{\Phi(\chi^{\mathsf{N}}_t) : \chi_t \in \{T, F\}\} = \mathsf{D}_t$.

Type $\langle\alpha\beta\rangle$, $\alpha\beta \in \mathsf{PT}$. Assume the two domains D^*_α and D^*_β are defined as well as the function Φ acting on $\mathsf{CME}_\alpha \cap \mathsf{RIGIDS}$ and $\mathsf{CME}_\beta \cap \mathsf{RIGIDS}$. Assume as well the induction hypothesis. In particular:

- Φ is well defined and $\mathsf{D}^*_\alpha = \{\Phi(A_\alpha) : A_\alpha \in \mathsf{CME}_\alpha \cap \mathsf{RIGIDS}\}$ and $\mathsf{D}^*_\beta = \{\Phi(A_\beta) : A_\beta \in \mathsf{CME}_\beta \cap \mathsf{RIGIDS}\}$.
- $\Phi(A_\alpha) = \Phi(A'_\alpha)$ iff $A_\alpha \equiv A'_\alpha \in \Delta$ and $\Phi(B_\beta) = \Phi(B'_\beta)$ iff $B_\beta \equiv B'_\beta \in \Delta$ for all elements $A_\alpha, A'_\alpha \in \mathsf{CME}_\alpha \cap \mathsf{RIGIDS}$ and $B_\beta, B'_\beta \in \mathsf{CME}_\beta \cap \mathsf{RIGIDS}$.
- $\chi_\alpha = \Phi(\chi^{\mathsf{N}}_\alpha)$ and $\chi_\beta = \Phi(\chi^{\mathsf{N}}_\beta)$ for any $\chi_\alpha \in \mathsf{D}_\alpha$ and $\chi_\beta \in \mathsf{D}_\beta$.
- $\mathsf{D}^*_\alpha = \{\Phi(\chi^{\mathsf{N}}_\alpha) : \chi_\alpha \in \mathsf{D}_\alpha\} = \mathsf{D}_\alpha$ and $\mathsf{D}^*_\beta = \{\Phi(\chi^{\mathsf{N}}_\beta) : \chi_\beta \in \mathsf{D}_\beta\} = \mathsf{D}_\beta$.

1. Let $A_{\langle\alpha\beta\rangle} \in \mathsf{CME}_{\langle\alpha\beta\rangle} \cap \mathsf{RIGIDS}$. We define $\Phi(A_{\langle\alpha\beta\rangle})$ as a function

$$\begin{array}{rccc} \Phi(A_{\langle\alpha\beta\rangle}) : & \mathsf{D}^*_\alpha & \longrightarrow & \mathsf{D}^*_\beta \\ & \Phi(\chi^{\mathsf{N}}_\alpha) & \mapsto & \Phi(A_{\langle\alpha\beta\rangle}\chi^{\mathsf{N}}_\alpha) \end{array}$$

   We define $\mathsf{D}^*_{\langle\alpha\beta\rangle} = \{\Phi(A_{\langle\alpha\beta\rangle}) : A_{\langle\alpha\beta\rangle} \in \mathsf{CME}_{\langle\alpha\beta\rangle} \cap \mathsf{RIGIDS}\}$ and we need to prove that Φ is independent of the particular representatives chosen. Namely, if $\Phi(\chi^{\mathsf{N}}_\alpha) = \Phi(C_\alpha)$ then $\Phi(A_{\langle\alpha\beta\rangle})(\Phi(\chi^{\mathsf{N}}_\alpha)) = \Phi(A_{\langle\alpha\beta\rangle})(\Phi(C_\alpha))$ for any $C_\alpha \in \mathsf{CME}_\alpha \cap \mathsf{RIGIDS}$.

To prove it, let $\Phi(\chi^{\mathsf{N}}_\alpha) = \Phi(C_\alpha)$. Then $\chi^{\mathsf{N}}_\alpha \equiv C_\alpha \in \Delta$ (induction hypothesis for type α) and then $A_{\langle\alpha\beta\rangle}\chi^{\mathsf{N}}_\alpha \equiv A_{\langle\alpha\beta\rangle}C_\alpha \in \Delta$ by Theorem 56, modus ponens and properties of maximal consistent sets.
And so, $\Phi(A_{\langle\alpha\beta\rangle}\chi^{\mathsf{N}}_\alpha) = \Phi(A_{\langle\alpha\beta\rangle}C_\alpha)$, by induction hypothesis. Using the definition of $\Phi(A_{\langle\alpha\beta\rangle})$ we finally obtain that $\Phi(A_{\langle\alpha\beta\rangle})(\Phi(\chi^{\mathsf{N}}_\alpha)) = \Phi(A_{\langle\alpha\beta\rangle})(\Phi(C_\alpha))$, the desired result.

2. The second condition reads as follows: $\Phi(A_{\langle\alpha\beta\rangle}) = \Phi(B_{\langle\alpha\beta\rangle})$ iff $A_{\langle\alpha\beta\rangle} \equiv A_{\langle\alpha\beta\rangle} \in \Delta$.
($\Rightarrow$) Let $\Phi(A_{\langle\alpha\beta\rangle}) = \Phi(B_{\langle\alpha\beta\rangle})$ and $A_{\langle\alpha\beta\rangle} \equiv B_{\langle\alpha\beta\rangle} \notin \Delta$. By Propositional Extensionality (6a)

$$\vdash (\forall X_\alpha(A_{\langle\alpha\beta\rangle}X_\alpha \equiv B_{\langle\alpha\beta\rangle}X_\alpha)) \to (A_{\langle\alpha\beta\rangle} \equiv B_{\langle\alpha\beta\rangle})$$

therefore, $\neg\forall X_\alpha(A_{\langle\alpha\beta\rangle}X_\alpha \equiv B_{\langle\alpha\beta\rangle}X_\alpha) \in \Delta$, using the properties of maximal consistent sets.
Thus, $\exists X_\alpha\neg(A_{\langle\alpha\beta\rangle}X_\alpha \equiv B_{\langle\alpha\beta\rangle}X_\alpha) \in \Delta$.

Since in propositional type theory all types are finite, D_α is finite. Let $\mathsf{D}_\alpha = \{\chi_{1,\alpha}, \ldots, \chi_{n,\alpha}\}$. By Axiom 4 we have
$\vdash (\exists X_\alpha\neg(A_{\langle\alpha\beta\rangle}X_\alpha) \equiv B_{\langle\alpha\beta\rangle}X_\alpha) \equiv (\neg(A_{\langle\alpha\beta\rangle}\chi^{\mathsf{N}}_{1,\alpha} \equiv$
$B_{\langle\alpha\beta\rangle}\chi^{\mathsf{N}}_{1,\alpha}) \vee ... \vee \neg(A_{\langle\alpha\beta\rangle}\chi^{\mathsf{N}}_{n,\alpha} \equiv B_{\langle\alpha\beta\rangle}\chi^{\mathsf{N}}_{n,\alpha}))$
Thus for at least one $\chi_{p,\alpha} \in \mathsf{D}_\alpha$ we have, $\neg(A_{\langle\alpha\beta\rangle}\chi^{\mathsf{N}}_{p,\alpha} \equiv B_{\langle\alpha\beta\rangle}\chi^{\mathsf{N}}_{p,\alpha}) \in \Delta$, using properties of maximal consistent sets (Lemma 35).
But $\Phi(A_{\langle\alpha\beta\rangle}) = \Phi(B_{\langle\alpha\beta\rangle})$ and so $\Phi(A_{\langle\alpha\beta\rangle})(\Phi(\chi^{\mathsf{N}}_{p,\alpha})) = \Phi(B_{\langle\alpha\beta\rangle})(\Phi(\chi^{\mathsf{N}}_{p,\alpha}))$ and so $\Phi(A_{\langle\alpha\beta\rangle}\chi^{\mathsf{N}}_{p,\alpha}) = \Phi(B_{\langle\alpha\beta\rangle}\chi^{\mathsf{N}}_{p,\alpha})$, by definition of Φ.
In this case, $A_{\langle\alpha\beta\rangle}\chi^{\mathsf{N}}_{p,\alpha} \equiv B_{\langle\alpha\beta\rangle}\chi^{\mathsf{N}}_{p,\alpha} \in \Delta$. (By induction hypothesis). Being Δ consistent, this is impossible.
($\Leftarrow$) Let $A_{\langle\alpha\beta\rangle} \equiv B_{\langle\alpha\beta\rangle} \in \Delta$. Then for all $\chi^{\mathsf{N}}_\alpha \in \mathsf{ME}_\alpha$, $A_{\langle\alpha\beta\rangle}\chi^{\mathsf{N}}_\alpha \equiv B_{\langle\alpha\beta\rangle}\chi^{\mathsf{N}}_\alpha \in \Delta$, using Teorem 56 and properties of maximal consistent sets. So, by induction hypothesis (item 2), $\Phi(A_{\langle\alpha\beta\rangle}\chi^{\mathsf{N}}_\alpha) \equiv \Phi(B_{\langle\alpha\beta\rangle}\chi^{\mathsf{N}}_\alpha)$. And by definition of Φ, $\Phi(A_{\langle\alpha\beta\rangle})(\Phi(\chi^{\mathsf{N}}_\alpha)) = \Phi(B_{\langle\alpha\beta\rangle})(\Phi(\chi^{\mathsf{N}}_\alpha)$. Thus, $\Phi(A_{\langle\alpha\beta\rangle}) = \Phi(B_{\langle\alpha\beta\rangle})$, because $\mathsf{D}^*_\alpha = \{\Phi(\chi^{\mathsf{N}}_\alpha) : \chi_\alpha \in \mathsf{D}_\alpha\}$ by induction hypothesis.

3. $\chi_{\langle\alpha\beta\rangle} = \Phi(\chi^{\mathsf{N}}_{\langle\alpha\beta\rangle})$ for all $\chi_{\langle\alpha\beta\rangle} \in \mathsf{D}_{\langle\alpha\beta\rangle}$. In fact, for type t it was already proved that $\Phi(T^{\mathsf{N}}) = T$ and $\Phi(F^{\mathsf{N}}) = F$. For type $\langle\alpha\beta\rangle$ we have:
On one hand:

$$\begin{array}{rccc} \Phi(\chi^{\mathsf{N}}_{\langle\alpha\beta\rangle}): & \mathsf{D}^*_\alpha & \longrightarrow & \mathsf{D}^*_\beta \\ & \Phi(\chi^{\mathsf{N}}_\alpha) & \mapsto & \Phi(\chi^{\mathsf{N}}_{\langle\alpha\beta\rangle}\chi^{\mathsf{N}}_\alpha) \end{array}$$

On the other hand:

$$\begin{array}{rccc} \chi_{\langle\alpha\beta\rangle}: & \mathsf{D}_\alpha & \longrightarrow & \mathsf{D}_\beta \\ & \chi_\alpha & \mapsto & \chi_{\langle\alpha\beta\rangle}\chi_\alpha \end{array}$$

By induction hypothesis, $\Phi(\chi_\alpha^{\mathsf{N}}) = \chi_\alpha$ and $\chi_{\langle\alpha\beta\rangle}\chi_\alpha = \Phi((\chi_{\langle\alpha\beta\rangle}\chi_\alpha)^{\mathsf{N}})$. By Lemma 5, $\vdash \chi^{\mathsf{N}}_{\langle\alpha\beta\rangle}\chi^{\mathsf{N}}_\alpha \equiv (\chi_{\langle\alpha\beta\rangle}\chi_\alpha)^{\mathsf{N}}$. Thus, $\chi^{\mathsf{N}}_{\langle\alpha\beta\rangle}\chi^{\mathsf{N}}_\alpha \equiv (\chi_{\langle\alpha\beta\rangle}\chi_\alpha)^{\mathsf{N}} \in \Delta$. So, $\Phi((\chi_{\langle\alpha\beta\rangle}\chi_\alpha)^{\mathsf{N}}) = \Phi(\chi^{\mathsf{N}}_{\langle\alpha\beta\rangle}\chi^{\mathsf{N}}_\alpha) = \chi_{\langle\alpha\beta\rangle}\chi_\alpha$. Therefore, both functions, $\chi_{\langle\alpha\beta\rangle}$ and $\Phi(\chi^{\mathsf{N}}_{\langle\alpha\beta\rangle})$, are the same.

4. Now we have to prove: $\mathsf{D}^*_{\langle\alpha\beta\rangle} = \{\Phi(\chi^{\mathsf{N}}_{\langle\alpha\beta\rangle}) : \chi^{\mathsf{N}}_{\langle\alpha\beta\rangle} \in \mathsf{D}_{\langle\alpha\beta\rangle}\} = \mathsf{D}_{\langle\alpha\beta\rangle}$
   Clearly, $\{\Phi(\chi^{\mathsf{N}}_{\langle\alpha\beta\rangle}) : \chi^{\mathsf{N}}_{\langle\alpha\beta\rangle} \in \mathsf{D}_{\langle\alpha\beta\rangle}\} \subseteq \mathsf{D}^*_{\langle\alpha\beta\rangle}$, since $\chi^{\mathsf{N}}_{\langle\alpha\beta\rangle} \in \mathsf{CME}_{\langle\alpha\beta\rangle}$ and all strictly propositional meaningful expressions are rigids.
   To see that $\mathsf{D}^*_{\langle\alpha\beta\rangle} \subseteq \mathsf{D}_{\langle\alpha\beta\rangle}$, let $\Phi(A_{\langle\alpha\beta\rangle}) \in \mathsf{D}^*_{\langle\alpha\beta\rangle}$, $A_{\langle\alpha\beta\rangle} \in \mathsf{CME}_{\langle\alpha\beta\rangle} \cap \mathsf{RIGIDS}$. By definition, $\Phi(A_{\langle\alpha\beta\rangle}) : \mathsf{D}^*_\alpha \to \mathsf{D}^*_\beta$. By induction hypothesis, $\mathsf{D}^*_\alpha = \mathsf{D}_\alpha$ and $\mathsf{D}^*_\beta = \mathsf{D}_\beta$. Therefore, $\Phi(A_{\langle\alpha\beta\rangle}) \in \mathsf{D}_\beta^{\mathsf{D}_\alpha}$. Now we use item 3. above and prove that $\mathsf{D}_{\langle\alpha\beta\rangle} \subseteq \{\Phi(\chi^{\mathsf{N}}_{\langle\alpha\beta\rangle}) : \chi^{\mathsf{N}}_{\langle\alpha\beta\rangle} \in \mathsf{D}_{\langle\alpha\beta\rangle}\}$. ■

6.2. Defining the Δ-structure

With the domains already defined it is straightforward to complete the definition of the required structure by defining $\langle W, R\rangle$ and I. The domain W and the relation R were introduced in Section 5.2.

DEFINITION 48. *Let Δ be a maximal consistent set which is named, $\Diamond$- saturated and extensionally algebraic-saturated. The* EHPTT *structure* $\mathcal{M}_\Delta = \langle W, R, D^*_0, (D^*_\alpha)_{\alpha\in\mathsf{PT}}, I\rangle$ *over Δ is made up of:*

- $W = \{[i] : i \in \mathrm{NOM}\}$
- $R = \{\langle [i][j]\rangle : @_i\Diamond j \in \Delta\}$
- $D^*_0 = \{[@_i\tau] : \tau \in \mathsf{ME}_0, i \in \mathrm{NOM}\}$
- $\mathsf{D}^*_\alpha = \{\Phi(A_\alpha) : A_\alpha \in \mathsf{CME}_\alpha \cap \mathsf{RIGIDS}\}$, *for each* $\alpha \in \mathsf{PT}$
- $I(c)([i]) = \Phi(@_ic) = [@_ic]$, *for* $c \in \mathsf{CON}_0$
- $I(f)([i]) = \Phi(@_if)$, *for* $f \in \mathsf{CON}_n$
- $I(j)([i]) = T$ *iff* $j \in [i]$, *for* $i \in \mathrm{NOM}$

CLAIM 49. $\mathcal{M}_\Delta = \langle W, R, D^*_0, (D^*_\alpha)_{\alpha\in\mathsf{PT}}, I\rangle$ *is a well defined* EHPTT *structure.*

PROOF. Since NOM is nonempty (Δ is named), $W \neq \varnothing$. R is well defined, as we saw in Section 5.2. Moreover, $\mathsf{D}_0^* \neq \varnothing$ since the set of rigidified terms of type 0 is nonempty. Moreover, $(D^*_\alpha)_{\alpha \in \mathsf{PT}} = \mathfrak{PT}$. ■

LEMMA 50 (Extended substitution). *Let Δ be a maximal consistent set which is also named, $\Diamond$-saturated and extensionally algebraic-saturated. Let $\mathcal{M}_\Delta$ be the structure built on Δ and $[i] \in W$. Then*

$$F_a^{\mathfrak{I}}([i]) = \Phi(@_i F_a \frac{(g(X^1_{\alpha_1}))^{\mathsf{N}} ... (g(X^n_{\alpha_n}))^{\mathsf{N}}}{X^1_{\alpha_1} ... X^n_{\alpha_n}})$$

where $\mathsf{Free}(F_a) \subseteq \{v_1, \ldots, v_m, X^1_{\alpha_1}, \ldots, X^n_{\alpha n}\}$, *(propositional variables are all different but their types not necessarily different), g is any assignment such that $g(v_l) = \Phi(@_i v_l)$, for any $l \in \{1, \ldots, m\}$ and $\mathfrak{I} = (\mathcal{M}_\Delta, g)$.*

PROOF. **Algebraic terms.**
Note that since in algebraic terms there are no occurrences of propositional variables we can simple write $\Phi(@_i\gamma)$ instead of $\Phi(@_i\gamma \frac{(g(X^1_{\alpha_1}))^{\mathsf{N}} ... (g(X^n_{\alpha_n}))^{\mathsf{N}}}{X^1_{\alpha_1} ... X^n_{\alpha_n}})$.

- $\tau \in \mathsf{ME}_0$
  - Variables: $v^{\mathfrak{I}}([i]) = g(v) = \Phi(@_i v) = [@_i v]$, by definition of D_0^*.
  - Individual constants: $c^{\mathfrak{I}}([i]) = I(c)([i]) = \Phi(@_i c)$, by $\mathcal{M}_\Delta$ definition.
  - Constants of type n: $f^{\mathfrak{I}}([i]) = I(f)([i]) = \Phi(@_i f)$, by $\mathcal{M}_\Delta$ definition.
  - Complex terms (for simplicity, we prove just for binary terms, but the argument is similar for the other cases):
    - ∗ Assume that it holds for τ, σ and γ, i.e., $\gamma^{\mathfrak{I}}([i]) = \Phi(@_i\gamma)$, $\tau^{\mathfrak{I}}([i]) = \Phi(@_i\tau)$ and $\sigma^{\mathfrak{I}}([i]) = \Phi(@_i\sigma)$. Then, $(\gamma(\tau,\sigma))^{\mathfrak{I}}([i]) =$
      $= \gamma^{\mathfrak{I}}([i])(\tau^{\mathfrak{I}}([i]), \sigma^{\mathfrak{I}}([i]))$, by definition
      $= \Phi(@_i\gamma)(\Phi(@_i\tau), \Phi(@_i\sigma))$, by ind. hypothesis
      $= \Phi(@_i\gamma)([@_i\tau], [@_i\sigma])$, by definition of Φ for type 0
      $= \Phi(@_i\gamma(@_i\tau, @_i\sigma))$, by definition of Φ for type 2
      $= [@_i\gamma(@_i\tau, @_i\sigma)]$, by definition of Φ for type 0
      $= [@_i(\gamma(\tau,\sigma))]$, by Axiom 6e (rigid fun. app.)
      $= \Phi(@_i(\gamma(\tau,\sigma)))$, by definition of Φ for type 0
    - ∗ Assume that it holds for τ.
      $(\lambda v_1 v_2 \tau)^{\mathfrak{I}}([i])$ is an arbitrary function on D_0^* such that
      $(\lambda v_1 v_2 \tau)^{\mathfrak{I}}([i])([@_j\tau_1], [@_k\tau_2]) = \tau^{\mathfrak{I}^{[@_j\tau_1][@_k\tau_2]}_{v_1 v_2}}([i]) =$
      $= (\tau \frac{@_j\tau_1 @_k\tau_2}{v_1 v_2})^{\mathfrak{I}}([i]) = \Phi(@_i\tau \frac{@_j\tau_1 @_k\tau_2}{v_1 v_2})$, using Lemma 26 and the inductive hypothesis.

On the other hand, $\Phi(@_i\lambda v_1 v_2\tau)$ is a binary function on D_0^* s.t.
$\Phi(@_i\lambda v_1 v_2\tau)([@_j\tau_1], [@_k\tau_2]) =$
$= \Phi((@_i\lambda v_1 v_2\tau)(@_j\tau_1, @_k\tau_2))$, by definition of Φ for type 2
$= [@_i\lambda v_1 v_2\tau(@_j\tau_1, @_k\tau_2)]$, by definition of Φ for type 0
$= [(\lambda v_1 v_2 @_i\tau)(@_j\tau_1, @_k\tau_2)]$, by equation Barcan Axiom 8
$= [@_i\tau\frac{@_j\tau_1 @_k\tau_2}{v_1 v_2}]$, by $\beta -$ conversion 6b
Therefore, $(\lambda v_1 v_2\tau)^{\mathfrak{I}}([i]) = \Phi(@_i\lambda v_1 v_2\tau)$

* Assume that the theorem holds for $\gamma \in \mathsf{ME}_n (n \geq 0)$. We have that $(@_j\gamma)^{\mathfrak{I}} =$
$= \gamma^{\mathfrak{I}}([j])$, by definition
$= \Phi(@_j\gamma)$, by ind. hypothesis
$= \Phi(@_i @_j\gamma)$
The last equality holds for type 0 since $@_j\gamma \approx @_i @_j\gamma \in \Delta$, and for type n since $@_j\gamma \equiv @_i @_j\gamma \in \Delta$.

Propositional expressions.

- For variables of any propositional type in $\mathsf{PT} - \{t\}$, say α. We want to see that $X_\alpha^{\mathfrak{I}}([i]) = \Phi(@_i X_\alpha\frac{(g(X_\alpha))^{\mathsf{N}}}{X_\alpha})$. $X_\alpha^{\mathfrak{I}}([i]) =$
$= g(X_\alpha)$, by def.
$= \Phi(g(X_\alpha))^{\mathsf{N}})$, by def. of Φ
$= \Phi(@_i(g(X_\alpha))^{\mathsf{N}})$, by Axiom 7f, max. consist. and props. of Φ
$= \Phi(@_i X_\alpha\frac{(g(X_\alpha))^{\mathsf{N}}}{X_\alpha})$
- For lambda expressions $\lambda X_\alpha A_\beta$ of propositional type $\langle\alpha\beta\rangle$ we want to prove that

$$(\lambda X_\alpha A_\beta)^{\mathfrak{I}}([i]) = \Phi((@_i\lambda X_\alpha A_\beta)\frac{(g(X^1_{\alpha_1}))^{\mathsf{N}}...(g(X^n_{\alpha_n}))^{\mathsf{N}}}{X^1_{\alpha_1}...X^n_{\alpha_n}})$$

where $\mathsf{Free}(A_\beta) \subseteq \{X_{\alpha_1}, \ldots, X_{\alpha_p}\}$.
We will prove that both functions $(\lambda X_\alpha A_\beta)^{\mathfrak{I}}([i])$ and $\Phi((@_i\lambda X_\alpha A_\beta)\frac{(g(X^1_{\alpha_1}))^{\mathsf{N}}...(g(X^n_{\alpha_n}))^{\mathsf{N}}}{X^1_{\alpha_1}...X^n_{\alpha_n}})$ give the same value for each argument.

1. For any element $\Phi({\chi_\alpha}^{\mathsf{N}}) \in \mathcal{D}_\alpha$ the value under function $(\lambda X_\alpha A_\beta)^{\mathfrak{I}}([i])$ is

$$(\lambda X_\alpha A_\beta)^{\mathfrak{I}}([i])(\Phi({\chi_\alpha}^{\mathsf{N}})) = (A_\beta)^{\mathfrak{I}^{\Phi({\chi_\alpha}^{\mathsf{N}})}_{X_\alpha}}([i])$$

Since $\Phi({\chi_{\alpha_k}}^{\mathsf{N}}) = \Phi(@_i{\chi_{\alpha_k}}^{\mathsf{N}})$ (by Axiom 7f, the properties of Δ as a maximal consistent set and the properties of Φ respecting equivalent relation based on Δ), we can use the induction hypothesis to get $\Phi(@_i{\chi_\alpha}^{\mathsf{N}}) = ({\chi_\alpha}^{\mathsf{N}})^{\Im}([i])$, therefore

$$(A_\beta)^{\Im^{\Phi({\chi_\alpha}^{\mathsf{N}})}_{X_\alpha}}([i]) = (A_\beta)^{\Im^{({\chi_\alpha}^{\mathsf{N}})^{\Im}([i])}_{X_\alpha}}([i])$$

2. For every element $\Phi(\chi_\alpha^{\mathsf{N}}) \in \mathsf{D}_\alpha$ the value under function $\Phi((@_i\lambda X_\alpha A_\beta)\frac{(g(X^1_{\alpha_1}))^{\mathsf{N}}...(g(X^n_{\alpha_n}))^{\mathsf{N}}}{X^1_{\alpha_1}...X^n_{\alpha_n}}))$ is given by

$$\begin{aligned}
&\Phi((@_i\lambda X_\alpha A_\beta)\tfrac{(g(X^1_{\alpha_1}))^{\mathsf{N}}...(g(X^n_{\alpha_n}))^{\mathsf{N}}}{X^1_{\alpha_1}...X^n_{\alpha_n}})(\Phi(\chi^{\mathsf{N}}_\alpha)\\
&= \Phi((@_i\lambda X_\alpha A_\beta)\tfrac{(g(X^1_{\alpha_1}))^{\mathsf{N}}...(g(X^n_{\alpha_n}))^{\mathsf{N}}}{X^1_{\alpha_1}...X^n_{\alpha_n}})(\chi^{\mathsf{N}}_\alpha)) && \text{, by def of } \Phi\\
&= \Phi((\lambda X_\alpha @_i A_\beta)\tfrac{(g(X^1_{\alpha_1}))^{\mathsf{N}}...(g(X^n_{\alpha_n}))^{\mathsf{N}}}{X^1_{\alpha_1}...X^n_{\alpha_n}})(\chi^{\mathsf{N}}_\alpha)) && \text{, by Barcan Axiom 8}\\
&= \Phi(@_i A_\beta \tfrac{(g(X^1_{\alpha_1}))^{\mathsf{N}}...(g(X^n_{\alpha_n}))^{\mathsf{N}}\ \ \chi^{\mathsf{N}}_\alpha}{X^1_{\alpha_1}...X^n_{\alpha_n}\ \ X_\alpha}) && \text{, } \beta \text{ conversion (6b)}\\
&= (A_\beta \tfrac{(g(X^1_{\alpha_1}))^{\mathsf{N}}...(g(X^n_{\alpha_n}))^{\mathsf{N}}\ \ \chi^{\mathsf{N}}_\alpha}{X^1_{\alpha_1}...X^n_{\alpha_n}\ \ X_\alpha})^{\Im}([i]) && \text{, by ind. hypothesis}\\
&= (A_\beta \tfrac{(g(X^1_{\alpha_1}))^{\mathsf{N}}...(g(X^n_{\alpha_n}))^{\mathsf{N}}}{X^1_{\alpha_1}...X^n_{\alpha_n}})^{\Im^{({\chi_\alpha}^{\mathsf{N}})^{\Im}([i])}_{X_\alpha}}([i]) && \text{, by lemma 26}
\end{aligned}$$

By using the induction hypothesis and the properties of Φ we get

$$((g(X^1_{\alpha_1}))^{\mathsf{N}})^{\Im}([i]) = \Phi((@_i(g(X^1_{\alpha_1}))^{\mathsf{N}})) = g(X^1_{\alpha_1})$$

Finally using n-times Lemma 26,

$$(A_\beta \tfrac{(g(X^1_{\alpha_1}))^{\mathsf{N}}...(g(X^n_{\alpha_n}))^{\mathsf{N}}}{X^1_{\alpha_1}...X^n_{\alpha_n}})^{\Im^{({\chi_\alpha}^{\mathsf{N}})^{\Im}([i])}_{X_\alpha}}([i]) = (A_\beta)^{\Im^{({\chi_\alpha}^{\mathsf{N}})^{\Im}([i])}_{X_\alpha}}([i])$$

Therefore, both functions

$(\lambda X_\alpha A_\beta)^{\Im}([i])$ and $\Phi((@_i\lambda X_\alpha A_\beta)\frac{(g(X^1_{\alpha_1}))^{\mathsf{N}}...(g(X^n_{\alpha_n}))^{\mathsf{N}}}{X^1_{\alpha_1}...X^n_{\alpha_n}})$ are the same.

- For any expression $A_{\langle\alpha\beta\rangle}B_\alpha$, with $\mathsf{Free}(A_{\langle\alpha\beta\rangle}B_\alpha) \subseteq \{X^1_{\alpha_1}, \ldots, X^n_{\alpha_n}\}$, of propositional type $\beta \neq t$

$$\begin{aligned}
&(A_{\langle\alpha\beta\rangle}B_\alpha)^{\Im}([i])\\
&= (A_{\langle\alpha\beta\rangle})^{\Im}([i])((B_\alpha)^{\Im}([i])) && (1)\\
&= \Phi((@_i A_{\langle\alpha\beta\rangle})\tfrac{(g(X^1_{\alpha_1}))^{\mathsf{N}}...(g(X^n_{\alpha_n}))^{\mathsf{N}}}{X^1_{\alpha_1}...X^n_{\alpha_n}})\Phi((@_i B_\alpha)\tfrac{(g(X^1_{\alpha_1}))^{\mathsf{N}}...(g(X^n_{\alpha_n}))^{\mathsf{N}}}{X^1_{\alpha_1}...X^n_{\alpha_n}}) && (2)\\
&= \Phi((@_i A_{\langle\alpha\beta\rangle})\tfrac{(g(X^1_{\alpha_1}))^{\mathsf{N}}...(g(X^n_{\alpha_n}))^{\mathsf{N}}}{X^1_{\alpha_1}...X^n_{\alpha_n}}(@_i B_\alpha)\tfrac{(g(X^1_{\alpha_1}))^{\mathsf{N}}...(g(X^n_{\alpha_n}))^{\mathsf{N}}}{X^1_{\alpha_1}...X^n_{\alpha_n}}) && (3)\\
&= \Phi((@_i A_{\langle\alpha\beta\rangle})B_\alpha)\tfrac{(g(X^1_{\alpha_1}))^{\mathsf{N}}...(g(X^n_{\alpha_n}))^{\mathsf{N}}}{X^1_{\alpha_1}...X^n_{\alpha_n}}) && (4)
\end{aligned}$$

Step (1) holds by definition, (2) by induction hypothesis and (3) is by definition of Φ. Using Axiom 6e (Rigid function application) and the properties of maximal consistent sets we get

$$(@_i A_{\langle\alpha\beta\rangle})\frac{(g(X^1_{\alpha_1}))^{\mathsf{N}} \ldots (g(X^n_{\alpha_n}))^{\mathsf{N}}}{X^1_{\alpha_1} \ldots X^n_{\alpha_n}}(@_i B_\alpha)\frac{(g(X^1_{\alpha_1}))^{\mathsf{N}} \ldots (g(X^n_{\alpha_n}))^{\mathsf{N}}}{X^1_{\alpha_1} \ldots X^n_{\alpha_n}} \equiv (@_i A_{\langle\alpha\beta\rangle})B_\alpha)\frac{(g(X^1_{\alpha_1}))^{\mathsf{N}} \ldots (g(X^n_{\alpha_n}))^{\mathsf{N}}}{X^1_{\alpha_1} \ldots X^n_{\alpha_n}} \in \Delta$$

that justifies the last step in our proof, as Φ respects the equivalence based on Δ.

- For any expression $@_j A_\alpha$ of propositional type $\alpha \neq t$ we want to prove that

$$(@_j A_\alpha)^{\mathfrak{I}}([i]) = \Phi((@_i @_j A)\frac{(g(X^1_{\alpha_1}))^{\mathsf{N}} \ldots (g(X^n_{\alpha_n}))^{\mathsf{N}}}{X^1_{\alpha_1} \ldots X^n_{\alpha_n}})$$

where $\mathsf{Free}(A_\alpha \equiv B_\alpha) \subseteq \{X^1_{\alpha_1} \ldots X^n_{\alpha_n}\}$.

$$\begin{aligned}
(@_j A_\alpha)^{\mathfrak{I}}([i]) &= (A_\alpha)^{\mathfrak{I}}([j]) && \text{, by def.}\\
&= \Phi((@_j A)\tfrac{(g(X^1_{\alpha_1}))^{\mathsf{N}} \ldots (g(X^n_{\alpha_n}))^{\mathsf{N}}}{X^1_{\alpha_1} \ldots X^n_{\alpha_n}}) && \text{, by ind. hyp.}\\
&= \Phi((@_i @_j A)\tfrac{(g(X^1_{\alpha_1}))^{\mathsf{N}} \ldots (g(X^n_{\alpha_n}))^{\mathsf{N}}}{X^1_{\alpha_1} \ldots X^n_{\alpha_n}}) && \text{, using Axiom 7e}
\end{aligned}$$

Formulas.

- For variables of type t we prove that $X_t^{\mathfrak{I}}([i]) = \Phi(@_i X_t \frac{(g(X_t))^{\mathsf{N}}}{X_t})$ using the same argument used in the previuos case for propositional variables of type $\alpha \neq t$.
- $j^{\mathfrak{I}}([i]) = T$ iff $j \in [i]$ iff $@_i j \in \Delta$ iff $\Phi(@_i j) = T$.
- For expression $A_{\langle\alpha\beta\rangle} B_\alpha$ of propositional type $\beta = t$, the proof is similar to the corresponding case above for type $\langle\alpha\beta\rangle$ with $\beta \neq t$.
- For any expression $F_a \equiv G_a$, with $a \neq 0$, where $\mathsf{Free}(F_a \equiv G_a) \subseteq \{X^1_{\alpha_1}, \ldots, X^n_{\alpha_n}\}$ we have that $(F_a \equiv G_a)^{\mathfrak{I}}([i]) = T$ iff

$$\begin{aligned}
&\text{iff } F_a^{\mathfrak{I}}([i]) = G_a^{\mathfrak{I}}([i]) && (1)\\
&\text{iff } \Phi((@_i F_a)\tfrac{(g(X^1_{\alpha_1}))^{\mathsf{N}} \ldots (g(X^n_{\alpha_n}))^{\mathsf{N}}}{X^1_{\alpha_1} \ldots X^n_{\alpha_n}}) = \Phi((@_i G_a)\tfrac{(g(X^1_{\alpha_1}))^{\mathsf{N}} \ldots (g(X^n_{\alpha_n}))^{\mathsf{N}}}{X^1_{\alpha_1} \ldots X^n_{\alpha_n}}) && (2)\\
&\text{iff } (@_i F_a \tfrac{(g(X^1_{\alpha_1}))^{\mathsf{N}} \ldots (g(X^n_{\alpha_n}))^{\mathsf{N}}}{X^1_{\alpha_1} \ldots X^n_{\alpha_n}}) \equiv (@_i G_a \tfrac{(g(X^1_{\alpha_1}))^{\mathsf{N}} \ldots (g(X^n_{\alpha_n}))^{\mathsf{N}}}{X^1_{\alpha_1} \ldots X^n_{\alpha_n}}) \in \Delta && (3)\\
&\text{iff } @_i(F_a \tfrac{(g(X^1_{\alpha_1}))^{\mathsf{N}} \ldots (g(X^n_{\alpha_n}))^{\mathsf{N}}}{X^1_{\alpha_1} \ldots X^n_{\alpha_n}} \equiv G_a \tfrac{(g(X^1_{\alpha_1}))^{\mathsf{N}} \ldots (g(X^n_{\alpha_n}))^{\mathsf{N}}}{X^1_{\alpha_1} \ldots X^n_{\alpha_n}}) \in \Delta && (4)\\
&\text{iff } @_i(F_a \equiv G_a)\tfrac{(g(X^1_{\alpha_1}))^{\mathsf{N}} \ldots (g(X^n_{\alpha_n}))^{\mathsf{N}}}{X^1_{\alpha_1} \ldots X^n_{\alpha_n}} \in \Delta && (5)\\
&\text{iff } \Phi((@_i F_a \equiv G_a)\tfrac{(g(X^1_{\alpha_1}))^{\mathsf{N}} \ldots (g(X^n_{\alpha_n}))^{\mathsf{N}}}{X^1_{\alpha_1} \ldots X^n_{\alpha_n}}) = T && (6)
\end{aligned}$$

Step (1) holds by definition, (2) by induction hypothesis, (3) and (6) by definition of Φ, (4) by Ax. Equality-at-i and (5) by definition of substitution.

- For expressions $\Diamond\varphi$ we have:

  $(\Diamond\varphi)^{\Im}([i]) = T$

  iff there is $j \in W$ s.t. $@_i\Diamond j \in \Delta$, $\varphi^{\Im}([j]) = T$

  iff there is $j \in W$ s.t. $@_i\Diamond j \in \Delta$, $\Phi(@_j\varphi\frac{(g(X^1_{\alpha_1}))^{\mathsf{N}}...(g(X^n_{\alpha_n}))^{\mathsf{N}}}{X^1_{\alpha_1}...X^n_{\alpha_n}} = T$ (1)

  iff there is $j \in W$ s.t. $@_i\Diamond j \in \Delta$, $(@_j\varphi\frac{(g(X^1_{\alpha_1}))^{\mathsf{N}}...(g(X^n_{\alpha_n}))^{\mathsf{N}}}{X^1_{\alpha_1}...X^n_{\alpha_n}} \in \Delta$

  iff $(@_i\Diamond\varphi\frac{(g(X^1_{\alpha_1}))^{\mathsf{N}}...(g(X^n_{\alpha_n}))^{\mathsf{N}}}{X^1_{\alpha_1}...X^n_{\alpha_n}} \in \Delta$ (2)

  iff $\Phi(@_i\Diamond\varphi\frac{(g(X^1_{\alpha_1}))^{\mathsf{N}}...(g(X^n_{\alpha_n}))^{\mathsf{N}}}{X^1_{\alpha_1}...X^n_{\alpha_n}}) = T$

  Step (1) holds by Induction hypothesis. The equivalence (2) holds using the Bridge Theorem (Theorem 65) in one direction, and the fact that Δ is $\Diamond$-saturated in the other direction.

- $(@_j\varphi)^{\Im}([i])$ can be proved as we did for similar expressions above. ■

Theorem 51 (Henkin's Theorem). *Every consistent set of sentences has a model.*

Proof. Let Γ be a consistent set of sentences. By Extended Lindenbaum 37 there exists a maximal consistent extension Δ of Γ which is named, $\Diamond$-saturated and and extensionally algebraic-saturated. We build the Δ-structure $\mathcal{M}_\Delta$ according to Definition 48. As Δ is named, there is a nominal $i \in \Delta$. By Extended Substitution Lemma (Lemma 50) for any formula φ we have that

$$\varphi^{\Im}([i]) = \Phi(@_i\varphi\frac{(g(X^1_{\alpha_1}))^{\mathsf{N}}...(g(X^n_{\alpha_n}))^{\mathsf{N}}}{X^1_{\alpha_1}...X^n_{\alpha_n}}),$$

where g is any assignment, $[i] \in W$ and $\Im = (\mathcal{M}_\Delta, g)$. When φ is a sentence, by using Coincidence and Substitution Lemmas (21 and 26), we get

$$\varphi^{\Im}([i]) = \Phi(@_i\varphi),$$

and so $\varphi^{\Im}([i]) = T$ iff $@_i\varphi \in \Delta$.

Let now $\varphi \in \Gamma$. Clearly, $@_i\varphi \in \Delta$ by Axiom Intro 7b and Modus Ponens and the maximality of Δ. Therefore, $\varphi^{\Im}([i]) = T$. ■

Theorem 52 (Completeness). *For all* Γ *and* φ *in* CME_t *the following holds:* $\Gamma \models \varphi$ *implies* $\Gamma \vdash \varphi$.

Proof. Standard. ■

Conclusions and future work

Identity and equality can be considered in different contexts with a diversity of meanings; for example: (1) in an algebraic context, equational identity is used to build equational theories and equational classes which are the basis of Universal Algebra; (2) in the context of propositional type theory, identity takes the form of the biconditional connective at the first level and it plays an important role in order to define other connectives and quantifiers with the help of lambda operator; (3) in Hybrid logic, identity between worlds can be defined by the formula $@_i j$, which is key in order to have Robinson Diagrams.

The logic we discuss in this paper, EHPTT, combines these three different logics, namely, propositional type theory, equational logic and hybrid modal logic. These logics have the three identities we describe above, and more. This combined nature is rather obvious in the syntax, as well as in the list of axioms and rules we include in the proof system we propose. In most sentences the three components are present and we have to create an harmonious unity. In the new logic the equational terms and formulas receive intensional interpretations, and that change is the origin of a completely novel environment. In the new landscape, most of the axioms and rules dealing with equality are restricted to rigid expressions. Intentionally, we did not make any effort to get a smaller set of axioms; on the contrary, we group the axioms in order to explicitly show the heterogeneous nature of our logic and to help the reader identify the origin of each axiom (however, it will be an interesting exercise to find an independent set of axioms for our logic).

The formal achievement of this paper is the axiomatic calculus and the proof that such axiomatization is sound and complete with respect to the intensional Kripke style semantics presented in [13]. Our completeness proof for EHPTT is inspired in the work of Henkin on completeness, namely in the three proofs Henkin published last century (see [7], [8] and [10])

There are still open questions that we would like to address in the near future. First, we would like to study the application of Henkin's method employed in the proof of completeness for Propositional Type Theory to a Many-valued Propositional Type Theory. We think that one can build the type hierarchy from the enlarged finite set of truth values as Henkin did from $\{T, F\}$, and use a similar nameability theorem to achieve completeness. In our logic EHPTT we assume that the universe of individuals is the same in each world; however, there are circumstances where some elements do not exist in some worlds. We think all this could be accommodated in a specific logic, by developing a propositional type theory allowing more truth values, and so partial functions appearing in the algebraic part could receive an adequate treatment.

Finally, the proliferation of logical systems used in mathematics, computer science, philosophy and linguistics (to what we also have contributed with our system EHPTT) makes their relationships between and their possible translations into one another a pressing issue. Translation between logics has been scarcely formulated as an ambitious paradigm whose tools would serve for handling the multiplicity of logics. From a strictly pragmatic perspective, the translation can allow, for example, borrowing proof systems or completeness theorems from one to another. We would like to study translation from the logics we have been studying into well behaved and studied logics, like many-sorted logic, general logics or labelled deduction systems.

Acknowledgments. This research has been possible thanks to two research projects sustained by MINECO (Spain) with references FFI2013-47126-P and FFI2017-82554, respectively. We also thanks the partial support by the Portuguese Foundation for Science and Technology (FCT) through CIDMA within project UID/MAT/04106/2013 and Dalí project POCI-01-0145-FEDER-016692. Special thanks to the reviewers of this article for their valuable comments and recommendations.

References

[1] Andrews, P., *A reduction of the axioms for the theory of propositional types*, Fundam. Math. **52** (1963), pp. 345–350.

[2] Andrews, P., An Introduction to Mathematical Logic and Type Theory: To Truth Through Proof. Academic press. 1986

[3] Areces, C. Blackburn, P. Huertas, A. Manzano, M. *Completeness in Hybrid Type Theory*, DOI 10.1007/s10992-012-9260-4. J Philos Logic (Journal of Philosophical Logic) 43 (2-3) (2014), pp 209-238. Springer

[4] Barbosa, L. S. Martins, Manuel A. Carreteiro, M. . A Hilbert-Style Axiomatisation for Equational Hybrid Logic. *Journal of Logic, Language and Information*, 23 (1) (2014), pp 31-52.

[5] Blackburn, P. van Benthem, J. *Modal Logic: A Semantic Perspective.* Handbook of Modal Logic. Elsevier, 2007.

[6] Blackburn, P. ten Cate, B. . *Pure Extensions, Proof Rules and Hybrid Axiomatics.* Studia Logica. vol. 84, (2006) pp. 277-322.

[7] Henkin, L. *The completeness of the first order functional calculus.* The Journal of Symbolic Logic. vol. 14 (1949), pp. 159-166.

[8] Henkin, L. *Completeness in the theory of types.*The Journal of Symbolic Logic. vol. 15 (1950), pp. 81-91.

[9] Henkin, L. *Some Notes on Nominalism.* The Journal of Symbolic Logic. vol. 18 (1953), pp. 19-29.

[10] Henkin, L. *A theory of propositional types*, Fundam. Math. **52** (1963), pp. 323–344.

[11] Henkin, L. *Identity as a logical primitive*, Philosophia **5** (1975), pp. 31–45.
[12] Manzano, M. Extensions of first order logic. Cambridge Univ. Press., Cambridge, 1996.
[13] Manzano, M. Martins, M. A. Huertas, A. *A Semantics for Equational Hybrid Propositional Type Theory*, Bulletin of the Section of Logic, **43**, (3-4) (2014), pp. 121–138.
[14] *Henkin on Completeness.* In The Life and Work of Leon Henkin. Essays on His Contributions. Manzano, M. Sain, I. and Alonso, E. (eds). (2014), pp. 149-176. Springer International Publishing.
[15] Manzano, M. and Moreno, M. C. *Identity, Equality, nameability and completeness.* Bulletin of the Section of Logic, Volume 46:3/4 (2017), pp. 169–195.
[16] Manzano, M. and Moreno, M. C. *Identity, Equality, nameability and completeness - Part II.* Bulletin of the Section of Logic. To appear.
[17] Quine, W. *Logic based on inclusion and abstraction.*, J. Symb. Log. **2** (1937), pp. 145–152.
[18] Ramsey, F. P. *The foundations of mathematics*, Proceedings L. M. S. (2) **25** (1926), pp. 338–384.
[19] Tarski, A. *Sur le terme primitif de la logistique*, Fundamenta math. **4** (1923), pp. 196–200.

María Manzano
Dep. of Philosophy, University of Salamanca, Spain
mara@usal.es

Manuel A. Martins
CIDMA, Dep. Mathematics, Universidade de Aveiro, Portugal
martins@ua.pt

Antonia Huertas
Dep. of Comp. Science, Universitat Oberta de Catalunya, Spain
mhuertass@uoc.edu

7. Appendix - Helpful results

THEOREM 53. *If* $\alpha \in \Gamma$ *then* $\Gamma \vdash \alpha$.

THEOREM 54 (Deduction Theorem). *If* $\Gamma \cup \{\varphi\} \vdash \psi$ *then* $\Gamma \vdash \varphi \to \psi$ *where* $\varphi, \psi \in \mathsf{ME}_t$.

THEOREM 55 (Modus Ponens with hypothesis).

$$\Delta \vdash \varphi \to \psi \textit{ and } \Delta \vdash \varphi \textit{ then } \Delta \vdash \psi.$$

THEOREM 56. *The two theorems below are easily obtained in our calculus:*

1. $\vdash \chi_\alpha^{\mathsf{N}} \equiv C_\alpha \to A_{\langle\alpha\beta\rangle}\chi_\alpha^{\mathsf{N}} \equiv A_{\langle\alpha\beta\rangle}C_\alpha$
2. $\vdash A_{\langle\alpha\beta\rangle} \equiv B_{\langle\alpha\beta\rangle} \to A_{\langle\alpha\beta\rangle}\chi_\alpha^{\mathsf{N}} \equiv B_{\langle\alpha\beta\rangle}\chi_\alpha^{\mathsf{N}}$.

THEOREM 57. $\mathbf{K}_@^{-1}$

$$\vdash (@_i\varphi \to @_i\psi) \to @_i(\varphi \to \psi)$$

THEOREM 58. *The following are derivable*

Name′ *If* $\vdash i \to \varphi$ *then* $\vdash \varphi$ *where* $i \in \mathsf{NOM}$ *does not occur in* φ

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* $\vdash @_i\Diamond\varphi \to \psi$

COROLLARY 59. *The following is also provable*

Name $'^*$ *If* $\Delta \vdash i \to \varphi$ *then* $\Delta \vdash \varphi$ *where* $i \in \mathsf{NOM}$ *does not occur in* $\Delta \cup \{\varphi\}$.

Paste$^*_\Diamond$ *If* $\Delta \vdash (@_i\Diamond j \wedge @_j\varphi) \to \psi$ *and* $j \neq i$ *and* j *does not occur in* $\Delta \cup \{\psi, \varphi\}$*, then* $\Delta \vdash @_i\Diamond\varphi \to \psi$

THEOREM 60 (Nom). $\vdash @_ij \to (@_j\varphi \equiv @_i\varphi)$.

THEOREM 61. $\vdash @_i(\tau \approx \sigma) \equiv (@_i\sigma \approx @_i\sigma)$

THEOREM 62. $\Delta \vdash A_t \equiv B_t$ *and* $\Delta \vdash A_t$ *then* $\Delta \vdash B_t$.

THEOREM 63. $\vdash @_ij \to @_ji$

THEOREM 64. $\vdash @_ij \to (@_jk \to @_ik)$.

THEOREM 65 (Bridge). $\vdash @_i\Diamond j \wedge @_j\varphi \to @_i\Diamond\varphi$.

THEOREM 66. $\vdash @_i\Diamond j \wedge @_jj' \to @_i\Diamond j'$.

THEOREM 67. $\vdash F^{\mathsf{N}} \equiv \neg T^{\mathsf{N}}$

7.1. Axioms and rules of Henkin's system are EHPTT-theorems

THEOREM 68 (Axiom Schema 2). $\vdash (A_t \equiv T^{\mathsf{N}}) \equiv A_t$ *(where* $T^{\mathsf{N}} ::= ((\lambda X_tX_t) \equiv (\lambda X_tX_t))$*).* ■

THEOREM 69 (Axiom Schema 5). $\vdash (X_\alpha \equiv Y_\alpha) \to ((Z_{\langle\alpha\beta\rangle} \equiv V_{\langle\alpha,\beta\rangle}) \to ((Z_{\langle\alpha\beta\rangle}X_\alpha) \equiv (V_{\langle\alpha\beta\rangle}Y_\alpha)))$.

THEOREM 70 (Axiom Schema 3). $\vdash (T^{\mathsf{N}} \wedge F^{\mathsf{N}}) \equiv F^{\mathsf{N}}$