

María Manzano and Manuel A. Martins and Antonia Huertas

## A SEMANTICS FOR EQUATIONAL HYBRID PROPOSITIONAL TYPE THEORY\*

### Abstract

The definition of identity in terms of other logical symbols is a recurrent issue in logic. In particular, in FOL there is no way of defining the global relation of identity, while in standard SOL this definition is not only possible, but widely used. In this paper, the reverse question is posed and affirmatively answered: Can we define with only equality and abstraction the remaining logical symbols? Our present work is developed in the context of an equational hybrid logic (i.e. a modal logic with equations as propositional atoms enlarged with the hybrid expressions: nominals and the @ operator). Our logical base is propositional type theory. We take the propositional equality,  $\lambda$  abstraction, nominals,  $\Diamond$  and @ operators as primitive symbols and we demonstrate that all of the remaining logical symbols can be defined, including propositional quantifiers and equational equality.

### 1. Introduction

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. Due to the central role this notion plays in logic, you can either be interested in how to define it using other logical concepts, or

---

\*This research has been possible thanks to the research project sustained by Ministerio de Ciencia e Innovación of Spain with reference FFI-2009-09345MICINN. M. Martins also acknowledges partial financial assistance by project FP7-PEOPLE-2012-IRSES (GetFun) and by the Portuguese Foundation for Science and Technology (FCT), under projects FCOMP-01-0124-FEDER-028923 (NASONI) and PEst-OE/MAT/UI4106/2014 (CIDMA)

else, in the opposite scheme. In the first case, you investigate what kind of logic is required. In the second one, you become interested in the definition of the other logical concepts (connectives and quantifiers) in terms of the identity relation, using also abstraction.

In first order logic (FOL), identity cannot be defined in equality-free FOL. This is true even in the best scenario of a formal language with only a finite set of predicate symbols in which a binary relation obeying the usual rules for equality can be defined. Even though we can express that two objects cannot be distinguished in the formal language, we can find models where the interpretation of this equality is not the identity (see [11], pages 55-56).

In second order logic (SOL), identity can be defined by Leibniz's principle. Thus, in equality-free SOL there is a formula to define equality for individuals and the relation defined by this formula is 'genuine' identity in any standard second order structure. SOL with the standard semantics has an extraordinary expressive power but poor logical properties, and, therefore, non-standard semantics has to be introduced. Within non-standard structures, there is no guarantee that the equivalence relation defined by Leibniz's principle is identity (see [11] page 159). In general, equality for relations is neither introduced as a primitive logical symbol nor defined using the rest of the symbols in SOL, since to follow Leibniz's pattern we would need third order variables.

We have analyzed the question of how to define equality in terms of other logical symbols. In this paper, the reverse question is posed and affirmatively answered: can we define, with only equality and abstraction, the remaining logical symbols?

It is known that the identity relation on the set of truth values,  $T$  and  $F$ , serves as the denotation of the biconditional and is usually defined using other connectives, but our question here is how to use identity to obtain the rest. We know that in propositional logic we are not able to define connectives, such as conjunction, whose truth table shows a value  $T$  on an odd number of lines, not even with equality and negation. We can allow quantification over propositional variables of all types (including second order propositional variables) and then all connectives are defined with equality and quantifier. Theories of this kind were studied by Leśniewski and received the name of protothetic. But what about quantifiers? Can they be defined with equality? Quine was the first to observe that this is possible in a system with the lambda abstractor operator. The idea of reducing

the other concepts to identity is an old one which was tackled with some success in 1923 by Tarski [15], who solved the case for connectors; three years later, Ramsey [14] raised the whole subject; it was Quine [12] who introduced quantifiers in 1937. It was finally answered in 1963 by Henkin [6], where he developed a system of propositional type theory (followed by Andrews' improvement [1]). Later, in 1975, Henkin wrote a whole paper [7] on this subject in a volume completely devoted to identity.

Equality in first-order modal logic and intensional logic has given rise to heated philosophical debates [3]. These were initiated by Quine's criticism of quantified modal logics [13]. A central part of the debate is the substitution of equals for equals in modal/intensional contexts; in [4] there is a detailed account of this discussion. The key point is that classic problems such as Frege's original morning star/evening star puzzle often give rise to analogous (but harder) problems when modal reasoning is involved. For example, the sentence "If the morning star is identical to the evening star, then it is necessary that the morning star is identical to the evening star", is true in many first-order modal logics. Various solutions have been attempted, such as generalizing the models to encompass so-called counterpart relations [10], but the problem is difficult and demands closer investigation.

General intensional logics are developed to accommodate the difference between what a term designates and what it means. The meaning of a formula in a formal representation system is a proposition or declarative sentence and the denotation of a term is an object in the represented domain. A number of versions of first order intensional logic can be found in the literature. They all face the following fundamental challenges: (1) Extensionalist tendencies in quantification; (2) De re and de dicto classical ambiguity; (3) Troubles with identity and (4) The need to introduce logical elements for tracing individuals across worlds/cases/times.

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

Our present work is developed in the context of equational hybrid logic (i.e. a modal logic with equations as propositional atoms enlarged with

the hybrid features: nominals and the @ operator). The goal of this paper is to investigate whether basic hybridization, equational logic and propositional type theory taken together also leads to the definitions of connectives, propositional quantifiers and the equality over individuals (of the algebra) in terms of lambda and propositional equality, and the answer will be shown as affirmative.

We define as well one important concept in modal logic, that of *rigid expressions*. These are expressions that 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). We are interested in the interconnection between the @ operator and equality and provide a variety of validities on equality.

The present paper comes after our own investigations in equational hybrid logic and in hybrid type theory, as described below. Equational hybrid logic is a fragment of first-order hybrid logic which is more expressive than propositional hybrid logic. Concerning applications in Computer Science, hybrid logics have proved worth to express the requirements of complex reconfigurable systems (see [9] and [8]). The basic idea is to model systems configurations in a suitable logical system and the reconfiguration by hybrid logic. Equational hybrid logic is one of these examples: equational logic is used to express data and functional properties, and, on top, the hybrid logic is employed to reason about change of configurations in response to varying context conditions. The worlds of the underlying Kripke frame are algebras modeling the system's functionality at the respective mode. Modal formulas, on the other hand, express the dynamic view of the system's evolution. Finally, nominals allow for reference to specific configurations. In [5], a version of (simple) equational hybrid logic is presented and a Hilbert axiomatization is proposed.

In this paper, we go further by presenting a semantic for a propositional type theory version of equational hybrid logic that aims to address some of the problems concerning intensional logic.

Our language is a fragment of the (full) higher-order hybrid logic presented in the recent publication [2] (by Areces et al.) where completeness is discussed. In that paper, we show that basic hybridization (adding nominals and @ operators) makes it possible to give straightforward Henkin-style completeness proofs even when the modal logic being hybridized is higher-order.

## 2. Equational Hybrid Propositional Type Theory ( $\mathcal{EHP\!TT}$ )

In our approach, to deal with equations we consider an equational hybrid propositional type theory with two different kinds of types: *algebraic* and *propositional types*. Besides, the set of atoms consists of nominals, variables or constants and the other meaningful expressions are defined using the lambda operator, the equality symbol  $\equiv$  and the modal operators  $\Diamond$  and  $\@$ . The hierarchy of types is mixed, based on algebraic types and on Henkin's hierarchy of propositional types [6].

### 2.1. Syntax

DEFINITION 2.1. **Type Symbols:** Let  $t$  be any fixed object. The set  $\text{TYPES} = \text{AT} \cup \text{PT}$  of types of  $\mathcal{EHP\!TT}$  is defined as follows:

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

DEFINITION 2.2. The set  $\text{ME}$  of *meaningful expressions of  $\mathcal{EHP\!TT}$*  is formed by:

- a family  $\text{CON} = \langle \text{CON}_n : n \in \text{AT} \rangle$  of denumerable (finite or infinite) sets of **non-logical constants** such that for each  $n \neq 0$ ,  $\text{CON}_n$  is a set of function symbols  $f$  of type  $n$ ;  $\text{CON}_0$  is a set of non-logical constants  $c$ ;
- a denumerably infinite set  $\text{VAR}_\alpha$  of **variables**  $X_\alpha$ , for each type  $\alpha \in \text{AT} \cup \text{PT}$ , and
- a denumerably infinite set  $\text{NOM}$  of **nominals**.

The set  $\text{ME}_a$  of *meaningful expressions of type  $a$*  is defined by:

- **Algebraic expressions -  $\text{ME}_n$ :**
  - \*  $\text{VAR}_n \cup \text{CON}_n \subseteq \text{ME}_n, n \in \text{AT}$ ;
  - \*  $\lambda v_1 \dots v_n \tau \in \text{ME}_n$ , if  $v_1 \dots v_n \in \text{VAR}_0, v_i \neq v_j$  for all  $i, j \leq n, i \neq j, \tau \in \text{ME}_0$  (we also write  $\lambda \bar{v} \tau$  for  $\lambda v_1 \dots v_n \tau$ );
  - \*  $\gamma(\tau_1, \dots, \tau_n) \in \text{ME}_0$ , if  $\gamma \in \text{ME}_n$  and  $\tau_1, \dots, \tau_n \in \text{ME}_0$ ;

- \*  $@_i\gamma$ , if  $i \in \text{NOM}$  and  $\gamma \in \text{ME}_n$ .
- *Propositional expressions* -  $\text{ME}_\alpha$ :
  - \*  $\text{VAR}_\alpha \subseteq \text{ME}_\alpha$ ,  $\alpha \in \text{PT}$ ;
  - \*  $\lambda X_\alpha A_\beta \in \text{ME}_{\langle\alpha\beta\rangle}$ , if  $X_\alpha \in \text{VAR}_\alpha$ ,  $A_\beta \in \text{ME}_\beta$ ,  $\alpha, \beta \in \text{PT}$ ;
  - \*  $A_{\langle\alpha\beta\rangle} B_\alpha \in \text{ME}_\beta$ , if  $A_{\langle\alpha\beta\rangle} \in \text{ME}_{\langle\alpha\beta\rangle}$ ,  $B_\alpha \in \text{ME}_\alpha$ ,  $\alpha, \beta \in \text{PT}$ .
- *Formulas* -  $\text{ME}_t$ :
  - \*  $\text{VAR}_t \cup \text{NOM} \subseteq \text{ME}_t$ ;
  - \*  $\Diamond\varphi \in \text{ME}_t$ , if  $\varphi \in \text{ME}_t$ ;
  - \*  $@_i\varphi \in \text{ME}_t$ , if  $i \in \text{NOM}$ ,  $\varphi \in \text{ME}_t$ ;
  - \*  $A_\alpha \equiv B_\alpha$ , if  $A_\alpha, B_\alpha \in \text{ME}_\alpha$ ,  $\alpha \in \text{AT} \cup \text{PT} - \{0\}$ .

Given a meaningful expression  $A$ , the **set of free variables occurring in  $A_\alpha$**  (notation  $\text{FreeVar}(A_\alpha)$ ) is defined recursively as follows:

- $\text{FreeVar}(A) = \emptyset$ , for  $A \in \text{CON} \cup \text{NOM}$ ;
- $\text{FreeVar}(X) = \{X\}$ , for  $X \in \text{VAR}$ ;
- $\text{FreeVar}(\lambda v_1, \dots, v_n \tau) = \text{FreeVar}(\tau) \setminus \{v_1, \dots, v_n\}$ ;
- $\text{FreeVar}(\gamma(\tau_1, \dots, \tau_n)) = \text{FreeVar}(\gamma) \cup \text{FreeVar}(\tau_1) \cup \dots \cup \text{FreeVar}(\tau_n)$ ;
- $\text{FreeVar}(A_\alpha \equiv B_\alpha) = \text{FreeVar}(A_\alpha) \cup \text{FreeVar}(B_\alpha)$ ;
- $\text{FreeVar}(A_{\langle\alpha\beta\rangle} B_\alpha) = \text{FreeVar}(A_{\langle\alpha\beta\rangle}) \cup \text{FreeVar}(B_\alpha)$ ;
- $\text{FreeVar}(\Diamond\varphi) = \text{FreeVar}(\varphi)$ ;
- $\text{FreeVar}(@_i\varphi) = \text{FreeVar}(\varphi)$ ;
- $\text{FreeVar}(\lambda X_\alpha A_\beta) = \text{FreeVar}(A_\beta) \setminus \{X_\alpha\}$ .

A meaningful expression  $A_t$  of type  $t$  is called a **sentence** if  $\text{FreeVar}(A_t) = \emptyset$ .

## 2.2. Semantics

In  $\mathcal{EHPTT}$  the semantic is intensional (i.e., every meaningful expression has an intensional interpretation) while the language has no intensional symbols. In practice, a meaningful expression of type  $\alpha$  receives as the interpretation a function from the set of worlds  $W$  to the extensional universe  $D_\alpha$ , the function being an object of intensional type.

**DEFINITION 2.3.** A structure for  $\mathcal{EHPTT}$  is a tuple  $\mathfrak{M} = \langle W, R, \mathfrak{PT}, \mathcal{A}, I \rangle$ , where:

1.  $W$  is the **set of worlds**,  $W \neq \emptyset$ ,  $R \subseteq W \times W$  is the **accessibility relation** and  $\mathcal{A}$  is a non empty set - the **carrier set** of the algebras;
2.  $\mathfrak{PT} = \langle D_\alpha \rangle_{\alpha \in \text{PT} \cup \text{AT}}$ , the **hierarchy of extensional algebraic and propositional types**, is defined recursively by:  
 $D_t = \{T, F\}$ ,  $D_{\langle \alpha, \beta \rangle} = D_\beta^{D_\alpha}$  ( $\alpha, \beta \in \text{PT}$ ),  $D_0 = \mathcal{A}$  and  $D_n = \mathcal{A}^{A^n}$ ;
3.  $I$  is a function whose domain is the union of the set of nominals with the set of all non-logical constants and the set of functional symbols such that
  - If  $i \in \text{NOM}$ ,  $I(i) : W \longrightarrow D_t$  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  $c \in \text{CON}_0$ ,  $I(c) : W \longrightarrow D_0$ ;
  - If  $f \in \text{CON}_n$ ,  $I(f) : W \longrightarrow D_n$ , with  $I(f)(w) : \mathcal{A}^n \longrightarrow \mathcal{A}$ .

DEFINITION 2.4. An **assignment of values to variables**,  $g$ , is a function having as domain the set  $\text{VAR}$  of all variables such that for any variable  $X_\alpha \in \text{VAR}$ ,  $g(X_\alpha) \in D_\alpha$ , for any  $\alpha \in \text{AT} \cup \text{PT}$ .

Observe that the assignment is extensional since the value of any variable of any type is an object of the same type. We also define the interpretation of a variable of type  $\alpha$  as a constant function of intensional type.

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

DEFINITION 2.5. An **interpretation for  $\mathcal{EHP\mathcal{T}\mathcal{T}}$**  is a pair  $\mathfrak{I} = \langle \mathfrak{M}, g \rangle$ , where  $\mathfrak{M}$  is a structure for  $\mathcal{EHP\mathcal{T}\mathcal{T}}$  and  $g$  is an assignment of values to variables. We denote the interpretation  $\langle \mathfrak{M}, g_X^\theta \rangle$  by  $\mathfrak{I}_X^\theta$ .

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

1. *Algebraic expressions.*

- $(X_n)^\mathfrak{I} : W \longrightarrow D_n$ , where  $(X_n)^\mathfrak{I}(w) = g(X_n)$  for any  $w \in W$ ;

- $c^{\mathfrak{J}} : W \longrightarrow D_0$ ,  $c^{\mathfrak{J}}(w) = (I(c))(w)$  for any  $c \in \text{CON}_0$ ;
- $f^{\mathfrak{J}} : W \longrightarrow D_n$ ,  $f^{\mathfrak{J}}(w) = (I(f))(w)$  for any  $f \in \text{CON}_n$ ;
- $(f(\tau_1 \dots \tau_n))^{\mathfrak{J}} : W \longrightarrow D_0$ ,  
 $(f(\tau_1 \dots \tau_n))^{\mathfrak{J}}(w) = f^{\mathfrak{J}}(w)(\tau_1^{\mathfrak{J}}(w) \dots \tau_n^{\mathfrak{J}}(w))$ ,  
 $f \in \text{ME}_n$  and  $\tau_1, \dots, \tau_n \in \text{ME}_0$ ;
- $(\lambda v_1 \dots v_n \tau)^{\mathfrak{J}} : W \longrightarrow D_n$ ,  $(\lambda v_1 \dots v_n \tau)^{\mathfrak{J}}(w) : \mathcal{A}^n \longrightarrow \mathcal{A}$ ,  
mapping  $\bar{a}$  to  $\tau^{\mathfrak{J}_{\bar{a}}}(w)$ , where  $\bar{a} = (a_1, \dots, a_n)$ ,  $\bar{v} = (v_1, \dots, v_n)$  and  
 $\mathfrak{J}_{\bar{a}}$  is the interpretation  $\langle \mathfrak{M}, g_{\bar{a}} \rangle$  with  $g_{\bar{a}}$  the  $\bar{v}$ -variant assignment  
 $\left( \left( (g_{v_1}^{a_1})_{v_2}^{a_2} \right) \dots \right)_{v_n}^{a_n}$ ;
- $(@_i \gamma)^{\mathfrak{J}} : W \longrightarrow D_n$ , where  $(@_i \gamma)^{\mathfrak{J}}(w) = \gamma^{\mathfrak{J}}(w^i)$  for any  $w \in W$ .

## 2. Propositional expressions

- $(X_{\alpha})^{\mathfrak{J}} : W \longrightarrow D_{\alpha}$ ,  $(X_{\alpha})^{\mathfrak{J}}(w) = g(X_{\alpha})$ , for any  $X_{\alpha} \in \text{VAR}_{\alpha}$ ,  
 $\alpha \in \text{PT}$ ;
- $(\lambda X_{\alpha} A_{\beta})^{\mathfrak{J}} : W \longrightarrow D_{\langle \alpha \beta \rangle}$ , with  $(\lambda X_{\alpha} A_{\beta})^{\mathfrak{J}}(w) : D_{\alpha} \longrightarrow D_{\beta}$ ,  
mapping  $x$  to  $(A_{\beta})^{\mathfrak{J}_{x_{\alpha}}}(w)$ ;
- $(A_{\langle \alpha \beta \rangle} B_{\alpha})^{\mathfrak{J}} : W \longrightarrow D_{\beta}$ ,  $(A_{\langle \alpha \beta \rangle} B_{\alpha})^{\mathfrak{J}}(w) = ((A_{\langle \alpha \beta \rangle})^{\mathfrak{J}}(w))(B_{\alpha}^{\mathfrak{J}}(w))$ .

## 3. Formulas.

- $(X_t)^{\mathfrak{J}} : W \longrightarrow D_{\alpha}$ , where  $(X_t)^{\mathfrak{J}}(w) = g(X_t)$  for any  $w \in W$ ;
- $i^{\mathfrak{J}} = I(i)$ ;
- $(\Diamond \varphi)^{\mathfrak{J}} : W \longrightarrow D_t$  mapping  $w$  to  $T$  if there exists  $w' \in W$ ,  
 $(w, w') \in R$  and  $\varphi^{\mathfrak{J}}(w') = T$  and to  $F$  otherwise;
- $(@_i \varphi)^{\mathfrak{J}} : W \longrightarrow D_t$  mapping  $w$  to  $T$  if  $\varphi^{\mathfrak{J}}(w^i) = T$  and to  $F$   
otherwise;
- $(A_{\alpha} \equiv B_{\alpha})^{\mathfrak{J}} : W \longrightarrow D_t$  mapping any  $w$  to  $T$  if and only if  
 $A_{\alpha}^{\mathfrak{J}}(w) = B_{\alpha}^{\mathfrak{J}}(w)$ ,  $\alpha \in \text{AT} \cup \text{PT} - \{0\}$ .

LEMMA 2.6. Let  $A_{\alpha}$  be a meaningful expression of type  $\alpha$  and  $\mathfrak{J} = \langle \mathfrak{M}, g \rangle$  and  $\mathfrak{J}' = \langle \mathfrak{M}, g' \rangle$  such that  $g$  and  $g'$  coincides in  $\text{FreeVar}(A_{\alpha})$ . Then

$$(A_{\alpha})^{\mathfrak{J}} = (A_{\alpha})^{\mathfrak{J}'}.$$

PROOF: By induction on the construction of expressions. □



Now, following Henkin [6], we can define all logical operators, quantifiers and (algebraic) equations using lambda and propositional equality, in the present case for equational hybrid propositional type theory. Finally, we prove that the new operators have the expected meaning.

DEFINITION 2.7. *We define the following expressions:*

1.  $T^n ::= ((\lambda X_t X_t) \equiv (\lambda X_t X_t))$  is a sentence of type  $t$ ;
2.  $F^n ::= ((\lambda X_t X_t) \equiv (\lambda X_t T^n))$  is a sentence of type  $t$ ;
3.  $\neg^n ::= (\lambda X_t (F^n \equiv X_t))$  of type  $\langle tt \rangle$ ;
4.  $\wedge^n ::= \lambda X_t (\lambda Y_t (\lambda f_{tt} (f_{tt} X_t \equiv Y_t)) \equiv (\lambda f_{tt} (f_{tt} T^n)))$  of type  $\langle t \langle tt \rangle \rangle$ ;
5.  $\forall X_\alpha A_t := ((\lambda X_\alpha A_t) \equiv (\lambda X_\alpha T^n))$  is a sentence of type  $t$ ;
6.  $\tau \approx \sigma ::= (\lambda \bar{v} \tau \equiv \lambda \bar{v} \sigma)$  is a sentence of type  $t$ , where  $\bar{v} = \langle v_1, \dots, v_n \rangle$  and  $\text{VarFree}(\tau) \cup \text{VarFree}(\sigma) = \{v_1, \dots, v_n\}$ .

The next theorem states that all the introduced connectives behave as usual. That is, we can define all basic standard operators within equational hybrid propositional type theory. The item 6 of the following theorem shows that the equational apparatus of the logic, which is not visible in the syntax, can be defined using lambda abstraction and equality between functions over individuals.

THEOREM 2.8. *For every interpretation  $\mathfrak{J}$  the following holds:*

1.  $(T^n)^\mathfrak{J} : W \longrightarrow D_t$ , with  $(T^n)^\mathfrak{J}(w) = T$ .
2.  $(F^n)^\mathfrak{J} : W \longrightarrow D_t$ , with  $(F^n)^\mathfrak{J}(w) = F$ .
3.  $(\neg^n)^\mathfrak{J} : W \longrightarrow D_t^{\text{D}_t}$  such that  $(\neg^n)^\mathfrak{J}(w)$  is the Boolean “negation”.
4.  $(\wedge^n)^\mathfrak{J} : W \longrightarrow D_{tt}^{\text{D}_t}$  such that  $(\wedge^n)^\mathfrak{J}(w)$  is the Boolean “conjunction”.
5.  $(\forall X_\alpha A_t)^\mathfrak{J} : W \longrightarrow D_t$ , mapping  $w$  to  $T$  only if  $(A_t)^{\mathfrak{J}_{\bar{x}_\alpha}}(w) = T$  for all  $x \in D_\alpha$ .
6.  $(\tau \approx \sigma)^\mathfrak{J} : W \longrightarrow D_t$  mapping  $w$  to  $T$  iff  $\left\{ \bar{a} \mid \tau^{\mathfrak{J}_{\bar{a}}}(w) = \sigma^{\mathfrak{J}_{\bar{a}}}(w) \right\} = A^n$ ,  
where  $\text{VarFree}(\tau) \cup \text{VarFree}(\sigma) = \{v_1, \dots, v_n\}$  and  $\bar{v} = \langle v_1, \dots, v_n \rangle$ .

PROOF: Suppose  $\mathfrak{J} = \langle \mathfrak{M}, g \rangle$  an arbitrary interpretation.

$$1. (T^n)^{\mathfrak{J}} = ((\lambda X_t X_t) \equiv (\lambda X_t X_t))^{\mathfrak{J}} : W \longrightarrow D_t$$

$$w \mapsto \begin{cases} T, & (\lambda X_t X_t)^{\mathfrak{J}}(w) = (\lambda X_t X_t)^{\mathfrak{J}}(w); \\ F, & \text{otherwise.} \end{cases}$$

Since for all  $w \in W$ ,  $(\lambda X_t X_t)^{\mathfrak{J}}(w) = (\lambda X_t X_t)^{\mathfrak{J}}(w)$ ,

$$(T^n)^{\mathfrak{J}} : W \longrightarrow D_t$$

$$w \mapsto T$$

$$2. (F^n)^{\mathfrak{J}} = ((\lambda X_t X_t) \equiv (\lambda X_t T^n))^{\mathfrak{J}} :$$

$$W \longrightarrow D_t$$

$$w \mapsto \begin{cases} T, & (\lambda X_t X_t)^{\mathfrak{J}}(w) = (\lambda X_t T^n)^{\mathfrak{J}}(w); \\ F, & \text{otherwise.} \end{cases}$$

We have,

$$(\lambda X_t X_t)^{\mathfrak{J}}(w) : D_t \longrightarrow D_t \text{ is the identity function, and}$$

$$x \mapsto x$$

$$(\lambda X_t T^n)^{\mathfrak{J}}(w) : D_t \longrightarrow D_t \text{ is the constant function that}$$

$$x \mapsto (T^n)^{\mathfrak{J}} x_t = T$$

maps every  $x$  into  $T$ .

Therefore, these two maps are different and consequently

$$(F^n)^{\mathfrak{J}} : W \longrightarrow D_t$$

$$w \mapsto F$$

$$3. (\neg^n)^{\mathfrak{J}} = (\lambda X_t (F^n \equiv X_t))^{\mathfrak{J}} : W \longrightarrow D_{tt} \text{ is defined by}$$

$$(\lambda X_t (F^n \equiv X_t))^{\mathfrak{J}}(w) : D_t \longrightarrow D_t$$

$$x \mapsto (F^n \equiv X_t)^{\mathfrak{J}x_t}(w)$$

We have

$$(F^n \equiv X_t)^{\mathfrak{J}x_t}(w) = \begin{cases} T, & (F^n)^{\mathfrak{J}x_t}(w) = (X_t)^{\mathfrak{J}x_t}(w) \\ F, & \text{otherwise.} \end{cases}$$

$$= \begin{cases} T, & F = x \\ F, & \text{otherwise.} \end{cases}$$

$$\begin{aligned}
4. \text{ Let } (\wedge^n)^{\mathfrak{J}} &= (\lambda X_t (\lambda Y_t (\lambda f_{tt} (f_{tt} X_t \equiv Y_t)) \equiv (\lambda f_{tt} (f_{tt} T^n))))^{\mathfrak{J}}. \\
(\wedge^n)^{\mathfrak{J}} : W &\longrightarrow D_{tt}^{D_t} \\
w &\mapsto \underbrace{(\lambda X_t (\lambda Y_t (\lambda f_{tt} (f_{tt} X_t \equiv Y_t)) \equiv (\lambda f_{tt} (f_{tt} T^n))))^{\mathfrak{J}}(w)}_{(*)}
\end{aligned}$$

with

$$\begin{aligned}
(*) : D_t &\longrightarrow D_{tt} \\
x &\mapsto f_x : D_t \longrightarrow D_t \\
y &\mapsto \underbrace{((\lambda f_{tt} (f_{tt} X_t \equiv Y_t)) \equiv (\lambda f_{tt} (f_{tt} T^n)))^{\mathfrak{J}_{X_t, Y_t}}(w)}_{(**)}
\end{aligned}$$

where

$$(**) = \begin{cases} T, & (\lambda f_{tt} (f_{tt} X_t \equiv Y_t))^{\mathfrak{J}_{X_t, Y_t}}(w) = (\lambda f_{tt} (f_{tt} T^n))^{\mathfrak{J}_{X_t, Y_t}}(w) \\ F, & \text{otherwise.} \end{cases}$$

On the one hand,

$$\begin{aligned}
&(\lambda f_{tt} (f_{tt} X_t \equiv Y_t))^{\mathfrak{J}_{X_t, Y_t}}(w) : D_{tt} \longrightarrow D_t \text{ is defined by} \\
&(f_{tt} X_t \equiv Y_t)^{\mathfrak{J}_{X_t, Y_t, f_{tt}}}(w) = \begin{cases} T, & (f_{tt} X_t)^{\mathfrak{J}_{X_t, Y_t, f_{tt}}}(w) = (Y_t)^{\mathfrak{J}_{X_t, Y_t, f_{tt}}}(w) \\ F, & \text{otherwise.} \end{cases} \\
&= \begin{cases} T, & ((f_{tt})^{\mathfrak{J}_{X_t, Y_t, f_{tt}}}(w))((X_t)^{\mathfrak{J}_{X_t, Y_t, f_{tt}}}(w)) = y \\ F, & \text{otherwise.} \end{cases} \\
&= \begin{cases} T, & f(x) = y \\ F, & \text{otherwise.} \end{cases}
\end{aligned}$$

On the other hand,

$$\begin{aligned}
&(\lambda f_{tt} (f_{tt} T^n))^{\mathfrak{J}_{X_t, Y_t}}(w) : D_{tt} \longrightarrow D_t \text{ is defined by} \\
&(\lambda f_{tt} (f_{tt} T^n))^{\mathfrak{J}_{X_t, Y_t}}(w)(f) = (f_{tt} T^n)^{\mathfrak{J}_{X_t, Y_t, f_{tt}}}(w) \\
&= ((f_{tt})^{\mathfrak{J}_{X_t, Y_t, f_{tt}}}(w))((T^n)^{\mathfrak{J}_{X_t, Y_t, f_{tt}}}(w)) \\
&= f(T)
\end{aligned}$$

Hence,

$$\begin{aligned}
 (**) &= \begin{cases} T, & \text{for any } f \in D_{tt}, f(x) = y \text{ iff } f(T) \\ F, & \text{otherwise.} \end{cases} \\
 &\stackrel{(1)}{=} \begin{cases} T, & (f_I(x) = y \text{ iff } T) \text{ and } (f_{\neg}(x) = y \text{ iff } F) \text{ and} \\ & (f_F(x) = y \text{ iff } F) \text{ and } (f_T(x) = y \text{ iff } T) \\ F, & \text{otherwise.} \end{cases} \\
 &= \begin{cases} T, & x = y = T \\ F, & \text{otherwise.} \end{cases}
 \end{aligned}$$

Note that (1) holds since there are just 4 elements in  $D_{tt}$ , namely the identity function  $f_I$ , the negation function  $f_{\neg}$ , the constant function  $f_F$  sending all elements to  $F$  and the constant function  $f_T$  sending all elements to  $T$ .

5.  $((\lambda X_{\alpha} A_t) \equiv (\lambda X_{\alpha} T^n))^{\mathcal{J}} : W \longrightarrow D_t$  is defined by
- $$((\lambda X_{\alpha} A_t) \equiv (\lambda X_{\alpha} T^n))^{\mathcal{J}}(w) = \begin{cases} T, & (\lambda X_{\alpha} A_t)^{\mathcal{J}}(w) = (\lambda X_{\alpha} T^n)^{\mathcal{J}}(w); \\ F, & \text{otherwise.} \end{cases}$$

On the one hand,

$$\begin{aligned}
 (\lambda X_{\alpha} A_t)^{\mathcal{J}}(w) &: D_{\alpha} \longrightarrow D_t \\
 x &\mapsto (A_t)^{\mathcal{J}_{X_{\alpha}}^x}(w)
 \end{aligned}$$

On the other hand,

$$\begin{aligned}
 (\lambda X_{\alpha} T^n)^{\mathcal{J}}(w) &: D_{\alpha} \longrightarrow D_t \\
 x &\mapsto (T^n)^{\mathcal{J}_{X_{\alpha}}^x}(w) = T
 \end{aligned}$$

Therefore,

$$(\forall X_{\alpha} A_t)^{\mathcal{J}}(w) = \begin{cases} T, & (A_t)^{\mathcal{J}_{X_{\alpha}}^x}(w) = T \text{ for all } x \in D_{\alpha}; \\ F, & \text{otherwise.} \end{cases}$$

6.  $(\tau \approx \sigma)^{\mathcal{J}} ::= (\lambda \bar{v} \tau \equiv \lambda \bar{v} \sigma)^{\mathcal{J}} : W \rightarrow D_t$ , where

$$\begin{aligned}
 (\lambda \bar{v} \tau \equiv \lambda \bar{v} \sigma)^{\mathcal{J}}(w) &= T \text{ iff} \\
 (\lambda \bar{v} \tau)^{\mathcal{J}}(w) &= (\lambda \bar{v} \sigma)^{\mathcal{J}}(w) \text{ iff} \\
 \text{for all } \bar{a} \in \mathcal{A}^n & \tau^{\mathcal{J}_{\bar{v}}^{\bar{a}}}(w) = \sigma^{\mathcal{J}_{\bar{v}}^{\bar{a}}}(w).
 \end{aligned}$$

Therefore,

$$(\tau \approx \sigma)^{\mathcal{J}}(w) = T \text{ iff } \left\{ \bar{a} \mid \tau^{\mathcal{J}_{\bar{v}}^{\bar{a}}}(w) = \sigma^{\mathcal{J}_{\bar{v}}^{\bar{a}}}(w) \right\} = \mathcal{A}^n$$

□

Next corollary states that the connectives  $\neg^n$  and  $\wedge^n$ , introduced by the definition above, behave as the negation and conjunction in modal logic.

**COROLLARY 2.9.** *Let  $A, B \in \mathbf{ME}_t$ . For every interpretation  $\mathfrak{I}$  the following holds:*

1.  $(\neg^n A)^\mathfrak{I} : W \longrightarrow \mathbf{D}_t$  and  $(\neg^n A)^\mathfrak{I}$  is mapping  $w \in$  to  $T$  if  $A^\mathfrak{I}(w) = F$  and to  $F$  if  $A^\mathfrak{I}(w) = T$ .
2.  $((\wedge^n A) B)^\mathfrak{I} : W \longrightarrow \mathbf{D}_t$  and  $((\wedge^n A) B)^\mathfrak{I}(w) = T$  iff  $A^\mathfrak{I}(w) = B^\mathfrak{I}(w) = T$ .

### 3. Some validities on equality

In this section we show the power of the lambda operator, the equality symbol  $\equiv$  and the hybrid operators  $@$  and  $\diamond$  in terms of expressing intuitive properties of the equality.

First, we define one important concept in modal logic: **rigid expressions**. These are expressions that 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).

**DEFINITION 3.1** (Rigid meaningful expressions). *The set **RIGIDS** of **rigid meaningful expressions** is defined inductively as follows:*

- $X_\alpha$ , for  $\alpha \in \mathbf{AT} \cup \mathbf{PT}$  ;
- $@_i \gamma$ , for  $\gamma \in \mathbf{ME}_n$ ;
- $@_i \varphi$ , for  $\varphi \in \mathbf{ME}_t$ ;
- $\lambda v_1 \cdots v_n \tau$ , for  $\tau \in \mathbf{ME}_0$  rigid expression and  $v_1 \cdots v_n \in \mathbf{VAR}_0$ ,  $v_i \neq v_j$ ;
- $\lambda X_\alpha A_\beta$ , for  $A_\beta \in \mathbf{ME}_\beta$  rigid expression and  $\alpha, \beta \in \mathbf{PT}$ ;
- $\gamma(\tau_1, \dots, \tau_n)$ , for rigid expressions  $\gamma \in \mathbf{ME}_n$  and  $\tau_1, \dots, \tau_n \in \mathbf{ME}_0$ ;
- $A_{\langle \alpha \beta \rangle} B_\alpha$ , for  $A_{\langle \alpha \beta \rangle}$  and  $B_\alpha$  rigid expressions and  $\alpha, \beta \in \mathbf{PT}$
- $A_\alpha \equiv B_\alpha$ , for  $A_\alpha$  and  $B_\alpha$  rigid expressions and  $\alpha \in \mathbf{AT} \cup \mathbf{PT} - \{0\}$ .

**LEMMA 3.2.** *Let  $\mathfrak{I}$  be a an interpretation. If  $A \in \mathbf{RIGIDS}$  then  $(A)^\mathfrak{I}(w) = (A)^\mathfrak{I}(v)$  for all  $w, v \in W$ .*

PROOF: By induction on the construction of rigid expressions.  $\square$

DEFINITION 3.3. *Given an interpretation  $\mathfrak{I}$  for  $\mathcal{EHP\mathcal{T}\mathcal{T}}$ , we say that a propositional expression  $A_t$  is valid in  $\mathfrak{I}$ , in symbols:  $\mathfrak{I} \models A_t$ , iff  $(A_t)^\mathfrak{I} = \mathbb{T}$  where  $\mathbb{T} : W \rightarrow D_t$  is the constant function such that  $\mathbb{T}(w) = T$  for all  $w \in W$ . We write  $\models A_t$  if  $\mathfrak{I} \models A_t$  for any interpretation  $\mathfrak{I}$  for  $\mathcal{EHP\mathcal{T}\mathcal{T}}$ .*

Next, we discuss the validity of some intuitively true statements about equality.

- **Reflexivity, symmetry and transitivity.**

To show reflexivity, i.e.,  $\models A_\alpha \equiv A_\alpha$ , let  $\mathfrak{I}$  be an interpretation for  $\mathcal{EHP\mathcal{T}\mathcal{T}}$ . Since  $(A_\alpha)^\mathfrak{I}(w) = (A_\alpha)^\mathfrak{I}(w)$  for all  $w \in W$ , we have  $(A_\alpha)^\mathfrak{I} \equiv (A_\alpha)^\mathfrak{I} = \mathbb{T}$ .

Symmetry and transitivity can be shown in a similar way.

Thus, the standard properties of equality are valid.

- **$\eta$ -conversion:**  $\models (\lambda X_\beta \gamma_{\langle \beta, \alpha \rangle} X_\beta) \equiv \gamma_{\langle \beta, \alpha \rangle}$ , where  $X_\beta$  is not free in  $\gamma_{\langle \beta, \alpha \rangle}$ .

Let  $\mathfrak{I}$  be an interpretation for  $\mathcal{EHP\mathcal{T}\mathcal{T}}$ .

$(\lambda X_\beta \gamma_{\langle \beta, \alpha \rangle} X_\beta)^\mathfrak{I}(w) : D_\beta \rightarrow D_\alpha$  is defined by

$$(\lambda X_\beta \gamma_{\langle \beta, \alpha \rangle} X_\beta)^\mathfrak{I}(w)(x) = (\gamma_{\langle \beta, \alpha \rangle} X_\beta)^\mathfrak{I} x_\beta(w) = (\gamma_{\langle \beta, \alpha \rangle})^\mathfrak{I} x_\beta(w) (X_\beta)^\mathfrak{I} x_\beta(w) = (\gamma_{\langle \beta, \alpha \rangle})^\mathfrak{I}(x).$$

On the other hand,

$$\begin{array}{ccc} \gamma_{\langle \beta, \alpha \rangle}^\mathfrak{I}(w) : D_\beta & \rightarrow & D_\alpha \\ x & \mapsto & \gamma_{\langle \beta, \alpha \rangle}^\mathfrak{I}(w)(x) \end{array}.$$

Hence,  $(\lambda X_\beta \gamma_{\langle \beta, \alpha \rangle} X_\beta)^\mathfrak{I}(w) = (\gamma_{\langle \beta, \alpha \rangle})^\mathfrak{I}(w)$  for all  $w \in W$ .

Then, we can create functions with equations and variables.

- **Total function:**  $\models @_i \lambda \bar{v} \tau \equiv \lambda \bar{v} @_i \tau$ .

Let  $\mathfrak{I}$  be an interpretation for  $\mathcal{EHP\mathcal{T}\mathcal{T}}$ .

On the one hand,  $(@_i \lambda \bar{v} \tau)^\mathfrak{I}(w) = (\lambda \bar{v} \tau)^\mathfrak{I}(w^i) : D_n \rightarrow D_0$ , mapping  $(a_1, \dots, a_n)$  to  $\tau^{\bar{a}}(w^i)$ , where  $\bar{a} = (a_1, \dots, a_n)$ .

On the other hand,  $(\lambda \bar{v} @_i \tau)^\mathfrak{I}(w) : D_n \rightarrow D_0$ , mapping  $(a_1, \dots, a_n)$  to  $@_i \tau^{\bar{a}}(w) = \tau^{\bar{a}}(w^i)$ .

Hence,  $(@_i \lambda \bar{v} \tau)^\mathfrak{I}(w) = (\lambda \bar{v} @_i \tau)^\mathfrak{I}(w)$ , for all  $w$  world of  $\mathfrak{I}$ .

Thus,  $(@_i \lambda \bar{v} \tau \equiv \lambda \bar{v} @_i \tau)^{\mathfrak{J}} = \mathbb{T}$   
 Therefore, functions are total at every world.

- **Equality-at-i:**  $\models @_i(\tau \approx \sigma) \equiv (@_i \tau \approx @_i \sigma)$ .

$\mathfrak{J}$  be an interpretation for  $\mathcal{EHP\!TT}$  and  $\bar{v} = \text{FreeVar}(\tau) \cup \text{FreeVar}(\sigma)$ .  
 On the one hand,  $(@_i(\tau \approx \sigma))^{\mathfrak{J}}(w) = (@_i(\lambda \bar{v} \tau \equiv \lambda \bar{v} \sigma))^{\mathfrak{J}}(w) =$   
 $(\lambda \bar{v} \tau \equiv \lambda \bar{v} \sigma)^{\mathfrak{J}}(w^i) = T$  iff  $(\lambda \bar{v} \tau)^{\mathfrak{J}}(w^i) = (\lambda \bar{v} \sigma)^{\mathfrak{J}}(w^i)$ .

On the other hand,  $(@_i \tau \approx @_i \sigma)^{\mathfrak{J}}(w) = (\lambda \bar{v} @_i \tau \equiv \lambda \bar{v} @_i \sigma)^{\mathfrak{J}}(w) = T$   
 iff  $(\lambda \bar{v} @_i \tau)^{\mathfrak{J}}(w^i) = (\lambda \bar{v} @_i \sigma)^{\mathfrak{J}}(w^i)$  iff  $(\lambda \bar{v} \tau)^{\mathfrak{J}}(w^i) = (\lambda \bar{v} \sigma)^{\mathfrak{J}}(w^i)$ .

The last equivalence holds because, by the Domain axiom, we have  
 $(\lambda \bar{v} @_i \tau)^{\mathfrak{J}}(w^i) = (@_i \lambda \bar{v} \tau)^{\mathfrak{J}}(w^i)$  and  $(\lambda \bar{v} @_i \sigma)^{\mathfrak{J}}(w^i) = (@_i \lambda \bar{v} \sigma)^{\mathfrak{J}}(w^i)$ .

Thus,  $(@_i(\tau \approx \sigma))^{\mathfrak{J}}(w) = (@_i \tau \approx @_i \sigma)^{\mathfrak{J}}(w)$  for all  $w$  in  $W$ .

Then, the equation formula defined from the lambda operator and the equality symbol  $\equiv$  express equality at every world.

- **Rigid function application:**  $\models @_i \gamma(\tau_1, \dots, \tau_n) \equiv (@_i \gamma)(@_i \tau_1, \dots, @_i \tau_n)$ , for  $\gamma \in \text{ME}_n$  and  $\tau_1, \dots, \tau_n \in \text{ME}_0$ .

Let  $\mathfrak{J}$  be an interpretation for  $\mathcal{EHP\!TT}$ . We have

$$\begin{aligned} (@_i \gamma(\tau_1, \dots, \tau_n) \equiv (@_i \gamma)(@_i \tau_1, \dots, @_i \tau_n))^{\mathfrak{J}} &= \mathbb{T} \\ \text{iff} \\ (@_i \gamma(\tau_1, \dots, \tau_n))^{\mathfrak{J}}(w) &= ((@_i \gamma)(@_i \tau_1, \dots, @_i \tau_n))^{\mathfrak{J}}(w) \end{aligned}$$

Moreover,

$$\begin{aligned} (@_i \gamma(\tau_1, \dots, \tau_n))^{\mathfrak{J}}(w) &= (\gamma(\tau_1, \dots, \tau_n))^{\mathfrak{J}}(w^i) = T \\ \text{iff} \\ (\gamma^{\mathfrak{J}}(w^i)(\tau_1^{\mathfrak{J}}(w^i), \dots, \tau_n^{\mathfrak{J}}(w^i)))^{\mathfrak{J}} &= T \\ \text{iff} \\ (@_i \gamma(@_i \tau_1, \dots, @_i \tau_n))^{\mathfrak{J}}(w) &= T \end{aligned}$$

Thus, the "ridigification" of a function can be expressed with the equality symbol  $\equiv$  and  $@$ .

- **Rigids are rigid:**  $\models @_i \tau \approx \tau$  if  $\tau$  is a rigid expression of type 0.

Let  $\mathfrak{J}$  be an interpretation for  $\mathcal{EHP\!TT}$ . We have

$(@_i\tau \approx \tau)^{\mathfrak{J}}(w) = (\lambda \bar{v} @_i\tau \equiv \lambda \bar{v}\tau)^{\mathfrak{J}}(w) = T$  iff  $(\lambda \bar{v} @_i\tau)(w) = (\lambda \bar{v}\tau)^{\mathfrak{J}}(w)$ . This equivalence holds because, by the Domain axiom,  $(\lambda \bar{v} @_i\tau)(w) = (\lambda \bar{v}\tau)(w^i)$  and, since  $\tau$  is rigid,  $(\lambda \bar{v}\tau)(w^i) = (\lambda \bar{v}\tau)(w)$ .

So, to be a rigid expression can be expressed with abstraction, the equality symbol and  $@$ .

There are other facts involving  $@$  and equality that can be easily shown to be valid. We just list some of them without a proof.

- **Back:**  $\models \Diamond @_i\varphi \rightarrow @_i\varphi$ .
- **Ref:**  $\models @_ii$ .
- **Sym:**  $\models @_ij \rightarrow @_ji$ .
- **Trans:**  $\models @_ij \wedge @_jk \rightarrow @_ik$ .
- **Agree:**  $\models @_i@_j\alpha_a \equiv @_j\alpha_a$ .
- $\models @_i(\varphi \equiv \psi) \equiv (@_i\varphi \equiv @_i\psi)$ .
- $\models @_i\varphi \equiv \varphi$  if  $\varphi$  is a rigid expression of type  $t$ .

#### 4. Conclusions and future work

In this paper, we have developed an intensional semantics for the equational hybrid type theory. This semantics is inspired by Henkin's work [6] about the definability of connectives using equality and  $\lambda$  abstraction.

We would like to stress that formulas of form  $@_ij$ , together with  $@_i\Diamond j$  are extremely important in hybrid logic.

*Why is this?* Well, in the case where  $\varphi$  is a nominal, we have a handy way of expressing equality. After all, the formula

$$@_ij$$

asserts that  $i$  and  $j$  name the same point. To put it in another way,  $@_ij$  is a modal way of expressing what  $i = j$  would express in classical logic.

Indeed, it is easy to see that the basic hybrid formulas, which express the *reflexivity*, *symmetry* and *transitivity* of equality, respectively, are all validities. *And when the  $\varphi$  (in  $@_i\varphi$ ) is of the form  $\Diamond i$ ?* The formula  $@_i\Diamond j$  says the point named  $i$  is related (by whatever the relation of interest is: perhaps temporal, perhaps epistemic, perhaps something else) to the



point  $j$ . In temporal hybrid logic we can express the property of trichotomy of the order relation:

$$@_i j \vee @_i Pj \vee @_j Pi.$$

This tells us that either  $i$  and  $j$  name the same point, or that at  $i$  we have that  $j$  lies in the past, or that at  $j$  we have that  $i$  lies in the past.

There is a great deal more that could be said (both technical and non-technical) about formulas of the form  $@_i j$  and  $@_i \Diamond j$ .

The next step in our work will be the development of a complete calculus for the equational hybrid propositional type theory. We intend to merge the calculus for the theory of propositional types provided by Henkin together with several axiomatic systems studied in a variety of hybrid logics.

## References

- [1] P. Andrews, *A reduction of the axioms for the theory of propositional types*, **Fundam. Math.** 52 (1963), pp. 345–350.
- [2] C. Areces, P. Blackburn, A. Huertas, M. Manzano, *Completeness in Hybrid Type Theory*, **Journal of Philosophical Logic** (2013).
- [3] T. Braüner, S. Ghilardi, *First Order Modal Logic*, [in:] P. Blackburn, J. Van Benthem, F. Wolter, **Handbook of Modal Logic. Studies in Logic and Practical Reasoning**, vol. 3, Elsevier, 2007.
- [4] M. C. Fitting and R. Mendelsohn, **First-Order Modal Logic**, Dordrecht: Kluwer, 1998.
- [5] L. S. Barbosa, Manuel A. Martins, M. Carreteiro, *A Hilbert-Style Axiomatization for Equational Hybrid Logic*, **Journal of Logic, Language and Information** (1)23 (2014), pp. 31–52.
- [6] L. Henkin, *A theory of propositional types*, **Fundam. Math.** 52 (1963), pp. 323–344.
- [7] L. Henkin, *Identity as a logical primitive*, **Philosophia** 5 (1975), pp. 31–45.
- [8] A. Madeira, *Foundations and techniques for software reconfigurability*, PhD thesis, Universidades do Minho, Aveiro and Porto (Joint MAP-i Doctoral Programme), July 2013.
- [9] A. Madeira, J. M. Faria, M. A. Martins, and L. S. Barbosa, *Hybrid specification of reactive systems: An institutional approach*, [in:] **9th International Conference on Software Engineering and Formal Methods (SEFM’11)**, Springer Lect. Notes Comp. Sci. 7041, 2011, pp. 250–259.

- [10] D. Lewis, *Counterpart theory of quantified modal logic*, **Journal of Philosophy** 65 (1968), pp. 113–126.
- [11] M. Manzano, **Extensions of first order logic**, Cambridge Univ. Press., Cambridge, 1996.
- [12] W. Quine, *Logic based on inclusion and abstraction*, **J. Symb. Log.** 2 (1937), pp. 145–152.
- [13] W. Quine, *Three grades of modal involvements*, [in:] **The Ways of Paradox and Other Essays**, Random House, 1953, pp. 156–174.
- [14] F. P. Ramsey, *The foundations of mathematics*, **Proceedings L. M. S.** (2)25 (1926), pp. 338–384.
- [15] A. Tarski, *Sur le terme primitif de la logistique*, **Fundamenta math.** 4 (1923), pp. 196–200.

Department of Philosophy  
University of Salamanca, Spain  
e-mail: mara@usal.es

CIDMA - Center for R&D in Mathematics and Applications  
Dep. Mathematics  
University of Aveiro, Portugal  
e-mail: martins@ua.pt

Department of Computer Science  
Universitat Oberta de Catalunya, Spain  
e-mail: mhuertass@uoc.edu