

Hybrid Logic with operations on nominals

Evangelos Tzanis

University College of London
Department of Computer Science
`e.tzanis@cs.ucl.ac.uk`

Abstract. The *basic hybrid logic* $\mathcal{H}(@)$ ([5]) is expanded with function symbols on nominals. We call this hybrid language $\mathcal{HCON}$ (**H**ybrid **L**ogic with **O**perators on **N**ominals), henceforth. The new language appears to be strong enough to express for example, any equationally definable class of algebras or the property of 2-colorability of a graph (not definable in the first order logic). The paper contains a number of worked examples, discussing the expressive power of $\mathcal{HCON}$.

Keywords: Hybrid Logics, skolem functions, Henkin Quantifiers

1 Introduction

Hybrid logic is the result of extending the basic modal language with a second sort of atomic propositions called nominals, and with satisfaction operators. Precisely, the nominals (denoted by $i, j, \dots$) behave similarly to ordinary proposition letters, except that nominals are true at a unique world. In other words, a nominal names a state by being true there and nowhere else. The language obtained by adding nominals to the basic modal language, is called the *minimal hybrid logic* $\mathcal{H}$. Satisfaction operators allow one to express that a formula holds at the world named by a nominal. A formula of the form $@_i\phi$ expresses the fact that ϕ holds at the world named by the nominal i . The extension of the basic modal language with nominals and satisfaction operators is called the *basic hybrid language* $\mathcal{H}(@)$.

1.1 Motivation

In this paper we introduce and study extensions of the basic hybrid logic adding function symbols on nominals. There are many reasons for being interested in these hybrid languages:

Algebraic-Structure: The motivation of this point comes from [5]: *you can name states but you can not give them structure*. In this paper we consider an application of hybrid logics to relational structures on algebras, thus a set with a relation and an algebraic structure. In general, modal logics which can explicitly handle both relations and functions have received little attention. We should mention that in [17] there is a complete list of papers which study Kripke frames in which the universe of possible worlds has a specific algebraic structure, besides the traditional relational component.

Frame Definability: Following the research lines of hybrid languages we accommodate the notion of pure formulas to $\mathcal{H}\mathcal{L}\mathcal{O}\mathcal{N}$. Pure formula is a formula containing only nominals (rather than propositional variables), defining frame classes that are always first-order definable (see, e.g., [16], Proposition 3.1). It is known from [11] that axiomatizations with pure formulae give rise to *natural* completeness proofs, in [8] a completeness proof for languages containing nominals and @-operator, was given. Although many first-order conditions that are modally definable are definable by pure formulae, for instance reflexivity to name one, this is not true in general. The right directedness property: $\forall vu\exists t(vRt \wedge uRt)$ is not definable by any orthodox modal formula and also is not equivalent to any pure formulae [8]. In this paper we show how frame properties, not definable by pure formulas of $\mathcal{H}(@)$, are definable by $\mathcal{H}\mathcal{L}\mathcal{O}\mathcal{N}$ – pure formulas. To be precise, we show that pure formulas of $\mathcal{H}\mathcal{L}\mathcal{O}\mathcal{N}$ can express any frame property definable by a first order formula.

Expressivity: We show that $\mathcal{H}\mathcal{L}\mathcal{O}\mathcal{N}$ is strong enough to express, under a modified definition of frame validity, the property of 2-colorability of graph, not definable in first order logic [14].

The paper is as follows. In section 2 we name a version of the basic hybrid logic $\mathcal{H}(@)$ and we call it $\mathcal{H}\mathcal{L}\mathcal{O}\mathcal{N}$ henceforth. In section 3, we name a version of the strong hybrid language $\mathcal{H}(@, \forall, \exists)$ and we call it $\mathcal{H}\mathcal{L}\mathcal{O}\mathcal{V}(@, \forall, \exists)$ (**H**ybrid **L**ogic with **O**perators only for **V**ariables). In the same chapter we show how to move from the universal fragment of $\mathcal{H}\mathcal{L}\mathcal{O}\mathcal{V}(@, \forall, \exists)$ to $\mathcal{H}\mathcal{L}\mathcal{O}\mathcal{N}$. In section 4 we present a number of examples discussing the expressive power of $\mathcal{H}\mathcal{L}\mathcal{O}\mathcal{N}$. In section 5 we show that $\mathcal{H}\mathcal{L}\mathcal{O}\mathcal{N}$ is strong enough to express any frame property definable by a first order formula.

2 The hybrid logic: $\mathcal{H}\mathcal{L}\mathcal{O}\mathcal{N}$

We are ready now to start our formal work on Hybrid Logics. First some basic definitions:

Definition 1. *An algebraic similarity type is an ordered pair $\sigma = (A, \rho)$ where A is a non empty set and ρ is a function $A \rightarrow N$. Elements of A are called functions symbols; the function ρ assigns to each operator $f \in A$ a finite arity (rank), indicating the number of arguments that f can be applied to. Function symbols of rank zero are called constants.*

Let **ANOM** be a countable set of atomic nominals: **ANOM** = $\{i, i_1, \dots\}$. Let σ be an algebraic similarity type.

Definition 2. *An algebraic language of nominals $(\sigma, \mathbf{ANOM})$ is built up using an algebraic similarity type $\sigma = (A, \rho)$ and a set of atomic nominals **ANOM**. The **NOM** = $TERM(\sigma, \mathbf{ANOM})$ of nominals formulas over σ and **ANOM** is given by the rule:*

$$\alpha := i \mid f(\alpha_1, \dots, \alpha_{\rho(f)})$$

where $\alpha \in \mathbf{NOM}$ and $f \in \sigma$.

Note that $\mathbf{CNOM} = \mathbf{NOM} \setminus \mathbf{ANOM}$, is the set of non atomic nominals; we call them complex nominals henceforth. Let $\mathbf{PROP} = \{p_1, p_2, \dots\}$ be a countable set of propositional variables. Let $\Phi = \mathbf{PROP} \cup \mathbf{NOM}$.

Definition 3. $\mathcal{HLOON}$ FRAMES and MODELS.

A model $\mathcal{M}$ for $\mathcal{HLOON}$ is a $\mathcal{M} = (W, R_\Diamond, (F^f)_{f \in \sigma}, V)$ such that, $W \neq \emptyset$, a set of possible worlds, $R_\Diamond$ is a binary relation over W , for each $n \geq 0$ and for all $f \in \sigma$, if f is an n -ary function symbol then F^f is an n -ary operator on W . Note that $F^f : W^n \rightarrow W$ and

- $V : \Phi \rightarrow \text{Pow}(W)$ and V is such that for all nominals $\alpha \in \mathbf{NOM}$, $V(\alpha)$ is singleton of W and:
 - If $\alpha \in \mathbf{NOM}$ then: $V(\alpha) = \{w\}$;
 - If $\alpha \in \mathbf{CNOM}$ then: $V(f(\alpha_1, \dots, \alpha_n)) = \{F^f(V(\alpha_1), \dots, V(\alpha_n))\}$.

We call the elements of $\mathcal{M}$ states or worlds, $R_\Diamond$ accessibility relations, F^f operators on W and V the valuation. We call the triple $(W, R_\Diamond, (F^f)_{f \in \sigma})$ the *functional-relational frame* underlying the model.

Definition 4. The Hybrid Language $\mathcal{HLOON}$ is a language that is used for describing models and frames. Its formulas are given by the following recursive definition.

$$\varphi := \alpha \mid p \mid \neg\varphi \mid \varphi \rightarrow \psi \mid \Diamond\varphi \mid @_\alpha\varphi$$

where $\alpha \in \mathbf{NOM}$ and $p \in \mathbf{PROP}$.

Note that all types of atomic symbols (propositional variables and nominals) are formulas. Furthermore a formula is **pure** if it contains no propositional variables and nominal free if it contains no nominals.

Definition 5. Semantics for $\mathcal{HLOON}$

Let a model $\mathcal{M} = (W, R_\Diamond, (F^f)_{f \in \sigma}, V)$ for $\mathcal{HLOON}$ (we give only the new case)

$$\mathcal{M}, w \models \alpha \text{ iff } V(\alpha) = \{w\}$$

$(\mathcal{M}, w)$ is called a *pointed model*. This is a model with a designated world, called its *point*, which is taken to be the *actual world*.

For simplicity we limit the language to one binary relation $R_\Diamond$ and to one function F (unary or binary depends on the example that we will use). In other words our hybrid model is: $\mathcal{M} = (W, R_\Diamond, F^f, V)$ or for brevity $\mathcal{M} = (W, R, F, V)$.

2.1 $\mathcal{H}\mathcal{L}\mathcal{O}\mathcal{N}$ Frame definability

In this section, we revisit the notion of *functional relational frame that is central to this paper*. Formally, on page 3 we defined the triple $\mathcal{F} = (W, R_\diamond, (F^f)_{f \in \sigma})$ to be a *functional relational frame*. Using this notion of a relational functional frame we can define classes of ordinary relational Kripke frames as follows: Let Φ be a set of $\mathcal{H}\mathcal{L}\mathcal{O}\mathcal{N}$ formulas. Define $\mathcal{F}r(\Phi)$ the class of Kripke frames validating Φ as the class of consisting of all $(W, R_\diamond)$ such that there is a functional relational frame $(W, R_\diamond, (F^f)_{f \in \sigma})$ extending it, and $(W, R_\diamond, (F^f)_{f \in \sigma}) \models \phi$ for all $\phi \in \Phi$. That is, we let Φ define the class of all Kripke frames obtained as **reducts** of functional relational frames validating Φ .

Frames like $\mathcal{F} = (W, R_\diamond, (F^f)_{f \in \sigma})$ are useful for at least two reasons. First, there may be application driven motivations to exclude certain valuations [6]. For example, in this paper we consider some linguistic and graph theoretical examples, showing that the use of function symbols with fixed valuation increases the expressive power of the basic hybrid logic. A second reason and perhaps the most important, is that frames like $\mathcal{F}$ support a notion of validity that definitely is weaker than the frame based one, but without losing too many information that make models so easy to work with.

However we can meet such frames in other logical formalisms. We list some of them here:

- General frames ([6], page 28) are frames with a collection of valuations A satisfying modally natural closure conditions. That is, a general frame is a structure of the form $(\mathcal{F}, A)$ where $\mathcal{F}$ is a relational frame and A stands for admissible valuations or algebra ([6], page 28). The motivation in [6] was the following: there is a universal completeness result for general frame validity, something that Kripke frame semantics lacks.
- Distance functions play an important role in the context of modal distance logics, where distance functions map worlds of a domain W to numbers. A consequence of using functions to our logical formalism is that we need to modify many Kripkean notions, like the notion of a frame. Precise definitions regarding distance frames and modal distance logics could be found in [20] (pages 18-22)

In this paper we relate the use of function symbols with the node creating rule of [7]. Actually this node creating rule amounts to using nominals as skolem functions. Roughly speaking, in [7] nominals were used to eliminate existential quantifiers of formulas not definable by any pure nominal formula. This approach might be better understood in terms of an example.

Church-Rosser is a frame condition, defined by the following first order formula $\forall x, y, z \exists t (xRy \wedge xRz \rightarrow (yRt \wedge zRt))$, but not definable by a pure formula (page 53 in [13], proposition 4.2). Therefore, one might ask whether we can obtain complete axiomatizations for frame classes definable by a set of pure formulas together with the Church-Rosser formula. In [7] the following inference rule was used:

Whenever $@_i \diamond j$ and $@_i \diamond k$ occur on a branch we are allowed to create new nominal l and add $@_j \diamond l$ and $@_k \diamond l$ to the branch

That is, l is a new nominal, eliminating the existential quantifier in $\forall x, y, z \exists t (xRy \wedge xRz \rightarrow (yRt \wedge zRt))$. As was proved in [7], the above rule preserves validity and is complete for the class of confluent frames. In section 4.3, using the notion of functional relational frames we show how the nominal creating rule can be accommodated to the level of frames. The notion of skolemization is important there.

Definition 6. A formula ϕ is valid at a state w in a functional relational frame $\mathcal{F} = (W, R_\diamond, (F^f)_{f \in \sigma})$, if ϕ is true at w in every model $(\mathcal{F}, V)$ on $\mathcal{F}$ and ϕ is valid on a functional relational frame $\mathcal{F}$ if ϕ is true at every state in every model $(\mathcal{F}, V)$ on $\mathcal{F}$.

We use the notation $\mathcal{HLOV}_F$ for the language based on the frame $\mathcal{F} = (W, R_\diamond, F)$, where $F = (F^f)_{f \in \sigma}$ indicates the set of operators in $\mathcal{F}$. It should be clear that the truth or falsity of a formula at a point depends only on propositional variables (denoted as $p_0^M, \dots$), nominals (denoted as $i_0^M, \dots$) and operators in it. Thus, given a sublanguage $\mathcal{HLOV}_{F'} \subset \mathcal{HLOV}_F$ with $F' \subset F$ and a frame $\mathcal{F} = (W, R, F)$ for $\mathcal{HLOV}_F$, we may define the **frame reduct** of $\mathcal{F}$ as $\mathcal{F} \downarrow_{F'} = (W, R, F')$.

We then have for every formula ϕ of $\mathcal{HLOV}_{L'}$ and every model $\mathcal{M}$ based on $\mathcal{F}$:

$$(\mathcal{F}, p_0^M, \dots, i_0^M, \dots), w \models \phi \text{ if and only if } (\mathcal{F} \downarrow_{F'}, p_0^M, \dots, i_0^M, \dots), w \models \phi$$

The observation does not depend on the presence of nominals.

3 $\mathcal{HLOV}(@, \forall, \exists)$

In this section, we name a version of the strong hybrid language $\mathcal{H}(@, \forall, \exists)$ and we call it $\mathcal{HLOV}(@, \forall, \exists)$ (**H**ybrid **L**ogic with **O**perators only for **V**ariables). We set up its language and semantics. Please treat this section as a basis for the next sections.

3.1 $\mathcal{HLOV}(@, \forall, \exists)$

Let **SVAR** be a countable set of state variables **SVAR** = $\{x_1, x_2, \dots\}$ and let **PROP** = $\{p_1, p_2, \dots\}$ be a countable set of propositional letters. We assume that these sets are pairwise disjoint. For simplicity, let **f** be a binary function (over state variables, and with fixed interpretation) and $R_\diamond$ be an accessibility relation symbol.

Definition 7. The terms s and formulas ϕ of the hybrid Language $\mathcal{HLOV}(@, \forall, \exists)$ are inductively defined as follows:

$$s := x \mid \mathbf{f}(s_1, s_2)$$

$$\varphi := \top \mid s \mid p \mid \neg\varphi \mid \varphi \rightarrow \psi \mid \Diamond\varphi \mid @_s\varphi \mid \forall s \phi \mid \exists s \phi$$

where $x \in \mathbf{VAR}$, $s, s_1, s_2 \in \mathbf{VAR}$ and $\mathbf{VAR} = \mathbf{TERM}(\mathbf{f}, \mathbf{SVAR})$.

Remark 1. Note that the difference between nominals and state variables is that nominals can not be bound by $\forall, \exists$, whereas state variables can.

Precise definitions regarding the semantics of strong hybrid languages can be found in [10]. Strong hybrid languages and $\mathcal{HLOV}(@, \forall, \exists)$, give us full first order expressive power over frames. This is known since the work of Arthur Prior [22] and [23].

Definition 8. (Hybrid Translation) $\mathcal{HLOV}(@, \forall, \exists)$ is strong enough to express any first order concept. We extend the hybrid translation $\mathcal{HT}(\cdot)$ (as was defined in [5]), with the following case:

$$\mathcal{HT}(\mathbf{f}(x, y)) = \mathbf{f}(x, y) \quad (1)$$

where $\mathcal{HT}(\cdot)$ maps first order sentences to hybrid formulas of $\mathcal{HLOV}(@, \forall, \exists)$.

Proposition 1. Let ϕ be a first order formula. Then for every model $\mathcal{M}$ and any assignment g , $\mathcal{M} \models \phi[g]$ iff $\mathcal{M}, g \models \mathcal{HT}(\phi)$.

Proof. Straightforward. Based on proposition 3.7, on page 45 of [1].

Is worth mentioning that adding to $\mathcal{HLOV}(@, \forall, \exists)$'s syntax, nominals and function over nominals then we can extend the $\mathcal{HT}(\cdot)$ translation as follows:

$$\mathcal{HT}(x = c) = @_x c \quad (2)$$

$$\mathcal{HT}(f(c_1, c_2)) = f(c_1, c_2) \quad (3)$$

Therefore, in terms of frame correspondence language (definition 3.4 of [6]), $\mathcal{HLOV}(\forall, \exists, @)$ is equally expressive with the first order logic with equality over the vocabulary containing constants, functions over constants, first order variables, functions over first order variables, binary relation symbols and unary predicate symbols.

3.2 $\mathcal{HLOV}$ and $\forall - \mathcal{HLOV}$

In this section we study the concept of frame definability in the context of $\mathcal{HLOV}$. Note that by $\forall - \mathcal{HLOV}$ we mean the universal fragment of $\mathcal{HLOV}(@, \forall, \exists)$. In what follows, we show that any formula of $\forall - \mathcal{HLOV}$ can be expressed by a pure formula of $\mathcal{HLOV}$.

Definition 9. Call a sentence of $\mathcal{HLOV}(@, \forall, \exists)$ of the form:

$\forall x_1 \dots x_n \phi(x_1, \dots, x_n, \mathbf{f}_1, \dots, \mathbf{f}_k)$, where ϕ does not contain any quantifiers, propositional letters or nominals a PUNF-sentence (**P**ure, **U**niversal, **N**ominal free sentence). Note that $x_1, \dots, x_n$ and $\mathbf{f}_1, \dots, \mathbf{f}_k$ are the state variables and the function symbols over variables occurring in ϕ , respectively. (PUNF-sentences are actually $\forall - \mathcal{HLOV}$ sentences)

Example 1. An $\forall - \mathcal{HLOV}$ sentence is:

$$\forall x \ \Diamond \Diamond \mathbf{f}(x) \rightarrow \Diamond \mathbf{f}(x)$$

Proposition 2. 1. For any pure formula ϕ of $\mathcal{HLOV}$ there is a PUNF-sentence $\phi^\forall$ of $\mathcal{HLOV}(@, \forall, \exists)$ such that for any frame $\mathcal{F}$, ϕ is valid on $\mathcal{F}$ iff $\phi^\forall$ is true on $\mathcal{F}$.
 2. For any PUNF-sentence ϕ of $\mathcal{HLOV}(@, \forall, \exists)$ there is a pure formula ϕ^B of $\mathcal{HLOV}$ such that for any frame $\mathcal{F}$, ϕ is true in $\mathcal{F}$ iff ϕ^B is valid on $\mathcal{F}$.
 Where $\mathcal{F}$ is a functional relational frame.

Proof. 1. Suppose we are given a pure formula $\phi(i_1, \dots, i_n, f_1, \dots, f_k)$, where $i_1, \dots, i_n$ are all the nominals and the function symbols over nominals in ϕ , then the required $\phi^\forall$ is $\forall x_1 \dots x_n \phi(x_1, \dots, x_n, \mathbf{f}_1, \dots, \mathbf{f}_k)$
 2. Any PUNF-sentence ϕ of $\mathcal{HLOV}(@, \forall, \exists)$ must have the form:
 $\forall x_1 \dots x_n \phi(x_1, \dots, x_n, \mathbf{f}_1, \dots, \mathbf{f}_k)$ where ϕ contains no quantifiers, propositional letters or nominals. Therefore, the required ϕ^B is $\phi([i_1 \leftarrow x_1, \dots, i_n \leftarrow x_n, f_1 \leftarrow \mathbf{f}_1, \dots, f_k \leftarrow \mathbf{f}_k])$.
 Note that $f_1, \dots, f_n$ are function over nominals and $\mathbf{f}_1, \dots, \mathbf{f}_k$ over state variables.

Based on this proposition we will show that $\mathcal{HLOV}$ is strong enough to express any equationally definable class of algebras.

4 Examples

A number of examples is presented discussing the expressive power of $\mathcal{HLOV}$. We start studying the connections between $\mathcal{HLOV}$ and equational systems. In section 4.2 we discuss whether we can express, using pure formulas of $\mathcal{HLOV}$, graph properties not definable in the first order logic. In fact, we show that pure formulas of $\mathcal{HLOV}$ are strong enough to express frame properties definable in the second order logic; this a new result as far as we know.

4.1 $\mathcal{HLOV}$ and Equational Systems

An example of the expressive power of $\mathcal{HLOV}$ is:

Example 2. (Semigroups) Let G be a nonempty set and $*$ be an operation on G . By the latter we mean a rule $*$ which assigns to each pair (x, y) of elements in G a unique element denoted $x * y$ in G . Now consider the following condition:

- (G1) The operation is associative: $\forall x, y, z \quad x * (y * z) = (x * y) * z$.

If (G1) holds, then G is said to be a semigroup under $*$. The condition (G1) in $\mathcal{HLO}\mathcal{N}$ can be expressed by the formula: $@_{i*(j*k)}(i * j) * k$, substituting nominals for variables.

Indeed the process is completely general. $\mathcal{HLO}\mathcal{N}$ is strong enough to express any equationally definable class of algebras. Let's be more precise summarizing some elementary definitions regarding equational systems ([6], Appendix B).

Definition 10. (*Syntax of Algebras*) Let σ an algebraic similarity type and a set $\mathcal{X}$ of variables. The language $\mathcal{E}$ of algebras (equational systems) is built up as follows: $t := x \mid f(t_1, \dots, t_{\rho(f)})$, we call t , $\mathcal{E}$ – terms.

Definition 11. (*Equation System*) An $\mathcal{E}(X, \sigma, =)$ equation system is an expression of the form: $s = t$, where s, t are $\mathcal{E}$ – terms.

Definition 12. (*Semantics of Equational systems*) A structure $\mathcal{M}$ for the equational system $\mathcal{E}$ is $\mathcal{M} = (D, (F^f)_{f \in \sigma}, V)$ such that: $D \neq \emptyset$, a non empty domain D ; for each $n \geq 0$ and for all $f \in \sigma$, if f is an n –ary function symbol then F^f is an n –ary operator on D . Note that $F : D^n \rightarrow D$; A variable x is assigned a meaning $V(x)$ which is an element D ; The $V(t)$ where t is a term, is defined as follows:

- If t is a variable x then: $V(x)$ is an element D ;
- If t is $f(t_1, \dots, t_n)$ then: $V(f(t_1, \dots, t_n)) = F(V(t_1), \dots, V(t_n))$

That is, an equational system is a set of equations. In other words an equational system is a set of universally closed equations of the form: $\forall x_1, \dots, x_n \quad s = t$, where $x_1, \dots, x_n$ is the set of variables occurring in the terms s, t . Therefore we have:

Proposition 3. For any equation of the form $\forall x_1, \dots, x_n \quad s = t$ there is a pure formula ϕ of $\mathcal{HLO}\mathcal{N}$ such that for any frame $\mathcal{F}$, ϕ is true on $\mathcal{F}$ iff ϕ is valid on $\mathcal{F}$.

Proof. As we have already mentioned, $\mathcal{HLO}\mathcal{V}(@, \forall, \exists)$ is strong enough to express any first order concept. Therefore any formula of the form $\phi = \forall x_1, \dots, x_n \quad s = t$ can be expressed by the following $\mathcal{HLO}\mathcal{V}(@, \forall, \exists)$ formula: $\phi' = \forall x_1, \dots, x_n \quad @_s t$, preserving truth satisfiability. But ϕ' preserves frame validity as well, since the $\forall$ quantifier captures the effect of trying out all possible assignments to the variables.

Then by proposition 2 we have that there is a pure formula ϕ^{nom} of $\mathcal{HLO}\mathcal{N}$, being valid on a frame $\mathcal{F}$ if and only if ϕ is valid on $\mathcal{F}$.

An immediate consequence of theorem 1 is that adding the $@$ – operator to the algebraic language (definition 2) we obtain a full equational system.

Definition 13. A propositional equational system $\mathcal{PES}$ of nominals $(\mathbf{ANOM}, \sigma, @)$ is built up using an algebraic type $\sigma = (A, \rho)$, a set of atomic nominals $\mathbf{ANOM}$ and the $@-$ operator. Its formulas are given the following recursive definition:

$$\alpha := i \mid f(\alpha_1, \dots, \alpha_{\rho(f)}) \mid @_{\alpha_1} \alpha_2$$

where $\alpha_1, \alpha_2 \in \mathbf{TERMS} = (\mathbf{ANOM}, \sigma, @)$.

To sum up, since the $@-$ operator can be regarded as the equality symbol of the first order logic, is abundantly clear that pure formulas of $\mathcal{HLON}$ can express equations, therefore algebras. To be precise, $\mathcal{HLON}$ is strong enough to express those algebras satisfying the following theorem of Birkhoff [3]:

Theorem 1. A class of algebras is equationally definable if and only if is a variety.

Another example regarding the expressive power of $\mathcal{HLON}$ is the following:

Example 3. Many important terms of universal algebra ([12]) like a discriminator term or a switching term are naturally expressed by $\mathcal{HLON}$ pure formulas. For instance:

$$(@_a b \rightarrow @_c t(a, b, c)) \vee (\neg @_a b \rightarrow @_a t(a, b, c)) \text{ discriminator term}$$

$$(@_a b \rightarrow @_c s(a, b, c, d)) \vee (\neg @_a b \rightarrow @_d s(a, b, c, d)) \text{ switching term}$$

4.2 Linguistics and Graph Theory

It is useful to mention some linguistic aspects of functions symbols in the context of hybrid logics. Actually using functions is possible to express sentences that speak about themselves. We will be clear on this point. First we should recall that a formula $@_i f(i)$ expresses the fact that $f(i)$ holds at the world named i and something stronger, that $f(i)$ depends on i .

Example 4. For instance the formula (can be found in [25]): **The (only) woman that every man loves is his mother**, illustrates a functional reading. The pronoun *his* is bound by the noun phrase *every man*, although it is not within its syntactic scope. We can get what we want by the formula, where f_{only} maps every man to his mother:

$$@_i < loves > f_{only}(i)$$

Roughly speaking the function $mother(i)$ is very similar to the notion of *Henkin Quantifiers*, expressing the notion of independence [9] and [21]. We continue using some terminology from graph theory:

Definition 14. A k -coloring of an undirected graph $G = (V, E)$ is a function $col : V \rightarrow \{1, \dots, k\}$ such that $col(u) \neq col(v)$ for all $(u, v) \in E$. G is called k -colorable if there is a k -coloring of G .

Example 5. (k-self referred functions) Assume that our hybrid frame is $\mathcal{F} = (W, R, F_1^{f_1}, \dots, F_k^{f_k})$ where $F_l^{f_l}$ are unary operators on W . $\mathcal{HLCN}$'s syntax is:

$$\varphi := i \mid f_1(i) \mid \dots \mid f_k(i) \mid p \mid \neg\varphi \mid \varphi \rightarrow \psi \mid \Diamond\varphi \mid @_i\varphi$$

where the intended meaning of $f_i(k)$ is: the color of i is k . For simplicity, we will be sloppy in our notation, assuming that the definition above is recursive, which clearly is not. Note that $F_l^{f_l}$ ($l = 1, \dots, k$) is the interpretation of f_l and:

$$F_l^{f_l}(w) = \begin{cases} w, & \text{if } w \text{ is } l \text{ colored} \\ !w, & \text{otherwise} \end{cases}$$

where $!w$, means some world other than w .

Therefore the condition $col(u) \neq col(v)$ if $(u, v) \in E$ is expressed by the formula:

$$\phi = \bigwedge_{l=1, \dots, k} (@_i \Diamond j \wedge @_i f_k(i)) \rightarrow \neg @_j f_k(j)$$

Also, the condition that every node is colored can be expressed as follows:

$$@_i f_1(i) \vee \dots \vee @_i f_k(i)$$

It is well known that the property of 2-colorability is not definable in first order logic; [14] and definable in monadic second order logic. We show that this property is definable by $\mathcal{HLCN}$ pure formulas as well.

Let ϕ be a conjunction of:

$$@_i f_1(i) \vee @_i f_2(i)$$

$$\bigwedge_{l=1, 2} (@_i \Diamond j \wedge @_i f_l(i)) \rightarrow \neg @_j f_l(j)$$

Then ϕ is valid on the frame $\mathcal{F}$ if and only if the graph defined by the relation R is 2-colorable. Note that a Kripke frame is the reduct of a functional relational frame validating ϕ if and only if the graph defined by the relation R is 2-colorable.

That is, we have shown that using pure nominal formulas we can express properties not definable in first order logic and definable in monadic second order logic.

4.3 Frame Definability and $\mathcal{H}\mathcal{L}\mathcal{ON}$

In this section we show that any frame property definable by a formula of the strong hybrid language can be expressed using pure formulas of $\mathcal{H}\mathcal{L}\mathcal{ON}$. We start with an example motivating our formalism.

Example 6. (Right directed). Right directness can be defined by the following formula of the strong hybrid language $\mathcal{H}\mathcal{L}\mathcal{ON}(@, \forall, \exists)$:

$$\phi = \forall v, u \exists t (@_v \diamond t \wedge @_u \diamond t)$$

Let us assume that ϕ is valid on a frame $F = (W, R)$. Then we can expand the given frame by adding a new binary function symbol F^f , obtaining the sentence $\phi' : \forall v, u @_v \diamond \mathbf{f}(u, v) \wedge @_u \diamond \mathbf{f}(u, v)$ such that:

$$\phi \text{ is valid on } \mathcal{F} \text{ iff } \phi' \text{ is valid on } \mathcal{F}'$$

where $\mathcal{F}' = (W, R, F^f)$. By proposition 2 we get:

$$\phi \text{ is valid on } \mathcal{F} \text{ iff } \phi^{nom} \text{ is valid on } \mathcal{F}'$$

where $\phi^{nom} = @_i \diamond f(i, j) \wedge @_j \diamond f(i, j)$.

We assume that i and j are particular names, but their interpretation in the model is arbitrary, so the universal quantification is implicit. Note that no orthodox modal formula defines the condition of *right – directed*.

Indeed the process can be shown to be completely general. Essentially, to express an $\mathcal{H}\mathcal{LOV}(@, \forall, \exists)$ formula, we introduce functions symbols to act as Skolem functions and obtain a pure $\mathcal{H}\mathcal{L}\mathcal{ON}$ formula whose validity is equivalent to that of the given $\mathcal{H}\mathcal{LOV}$. In [24] we proved a completeness result for functional relational frames and $\mathcal{H}\mathcal{L}\mathcal{ON}$ pure nominal formulas.

It is a routine exercise to show that every formula of $\mathcal{H}\mathcal{LOV}(@, \forall, \exists)$ can be rewritten in to a PNF equivalent. Recall that, a formula is on Prenex normal formula (PNF) if it has the form $Q_1 x_1 Q_2 x_2 \cdots Q_n x_n \phi$ where $Q_i \in \{\forall, \exists\}$ are quantifiers and ϕ is quantifier free.

Lemma 1. *Given the sentence $\exists y G(y)$ ($G(y)$ is a quantifier free formula) of $\mathcal{H}(@, \forall, \exists)$, augment the language with a new constant c and form the sentence $G(c)$ such that:*

$$G(y) \text{ is valid on } \mathcal{F} \text{ iff } G(c) \text{ is valid on } \mathcal{F}'$$

Given the sentence $\forall x_1, \dots, x_n \exists y G(\bar{x}, y)$ of $\mathcal{H}(@, \forall, \exists)$, augment the language with a new n -ary function symbol f and form the sentence $\forall x_1, \dots, x_n G'(\bar{x}, \mathbf{f}(\bar{x}))$. Then,

$$\forall x_1, \dots, x_n \exists y G(\bar{x}, y) \text{ is valid on } \mathcal{F} \text{ iff } \forall x_1, \dots, x_n G'(\bar{x}, \mathbf{f}(\bar{x})) \text{ is valid on } \mathcal{F}'$$

where $\mathcal{F}$ and $\mathcal{F}'$ are frames of $\mathcal{H}(@, \forall, \exists)$.

Proof. In each step we can expand a frame of the sentence on the left hand side to one on the right hand side by adding the appropriately designated element for the constant, or by adding an appropriate function $F^{\mathbf{f}}$, where F , is the interpretation of $\mathbf{f}$ in $\mathcal{F}$. Conversely, we can take a frame of the sentence on the right hand side and remove the constant symbol, respectively function symbol and we get a frame for the left hand side.

We call the procedure of removing one existential quantification and replacing the variable it bound by a functional term, as in the Lemma 1, a *Skolemization step*. We use the term *Skolemization* for the removal of all existential quantifications in this manner:

Theorem 2. *Given a formula ϕ of the strong hybrid language $\mathcal{H}(@, \forall, \exists)$ there is an effective procedure for finding a universal sentence $\phi^\forall$ of $\forall - \mathcal{HLOV}$ such that:*

$$\phi \text{ is valid on } \mathcal{F} \text{ iff } \phi^\forall \text{ is valid on } \mathcal{F}'$$

Furthermore, we can choose $\phi^\forall$ such that every frame of ϕ can be expanded to a frame of $\phi^\forall$ and every frame $\phi^\forall$ can be reduced to frame of ϕ .

Proof. First we put ϕ in prenex form (this is straightforward). Then we just apply the lemma 1 repeatedly until there are no existential quantifiers.

Then by proposition 2,

Theorem 3. *For any frame $\mathcal{F}$ definable by a formula ϕ of the strong hybrid language $\mathcal{HLOV}(@, \forall, \exists)$ we can associate a pure formula of $\mathcal{HLOV}$ such that:*

$$\phi \text{ is valid on } \mathcal{F} \text{ iff } \phi^{nom} \text{ is valid on } \mathcal{F}'$$

To sum up, we have shown that the traditional fragment of the hybrid logic becomes in a rather unorthodox interpretation equivalent to first order logic by extending the basic hybrid language with functions working on nominals denoting worlds in the frame.

5 Conclusions

We hope that we have convinced the reader that is useful to add function symbols in the context of hybrid logics. Roughly speaking, we have shown that $\mathcal{HLOV}$ is strong enough to express, any equationally definable class of algebras or the property of 2-colorability of a graph (not definable in the first order logic).

I am indebted to Robin Hirsch for his assistance and guidance throughout. I would like to thank the three anonymous referees for their useful comments and suggestions.

References

1. Carlos Eduardo Areces. Logic Engineering, The case of Description and Hybrid Logics, PhD thesis, ILLC, University of Amsterdam, 2000.
2. C. Areces, P. Blackburn, and M. Marx, Hybrid logics, Characterization, interpolation and complexity. *Journal of Symbolic Logic*, 66(3), 977-1010, 2001.
3. G. Birkhoff, On the structure of abstract algebras, *Proc. Cambr. Philos. Soc.* 31 (1935), 433-454.
4. P. Blackburn, Internalizing labelled deduction. *Journal of Logic and Computation*, 10(1): 137 - 168, 2000.
5. Patric Blackburn, Representation, Reasoning, and Relational Structures: a Hybrid Logic Manifesto. *Proc. of the 1st. Method for Modalities Workshop*. Amsterdam. Areces, C., Franconi, E., Gor, R., de Rijke, M., Schlingloff, H., editors. Special Issue of the *Logic Journal of the IGPL*. Vol 8:3, 339-625, 2000.
6. Patrick Blackburn, Maarten de Rijke and Yde Venema, *Modal Logic*, Cambridge University Press, 2001.
7. Patrick Blackburn and Balder ten Cate, Beyond Pure Axioms: Node Creating Rules in Hybrid Tableaux. In Carlos Areces, Patrick Blackburn, Maarten Marx and Ulrike Sattler, editors, *Proceedings of the 4th Workshop on Hybrid Logics (HyLo 2002)*.
8. P. Blackburn and M. Tzakova. Hybrid languages and temporal logic. *Logic Journal of the IGPL*, 7(1):2754, 1999.
9. A. Blass and Y. Gurevich. Henkin quantifiers and complete problems. *Ann. Pure Appl. Logic* 32:1, 1-16 (1986).
10. P. Blackburn and J. Seligman. Hybrid Languages. *Journal of Logic, Language and Information*, 4(3): 251-272, 1995.
11. R. Bull. An approach to tense logic. *Theoria*, 36(3):282300, 1970.
12. Stanley Burris, H P Sankappanavar, *A course in universal algebra*, Springer-Verlag, New York, 1981.
13. Balder ten Cate, *Model theory for extended modal languages*. PhD thesis, University of Amsterdam, 2005.
14. B. Courcelle. Graph rewriting: An algebraic and logic approach. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume Volume B: Formal Models and Semantics, pages 193-242. Elsevier and MIT Press, 1990.
15. Bruno Courcelle. The Expression of Graph Properties and Graph Transformations in Monadic Second-Order Logic. *Handbook of Graph Grammars 1997*: 313-400
16. George Gargov and Valentin Goranko. Modal Logic with Names. *Journal of Philosophical Logic*, 22(6):607636, 1993.
17. Valentin Goranko and Dimitar Vakarelov. Universal Algebra and Modal Logic, in: *Advances in Modal Logic*, vol. II, CSLI publications, Stanford, 2000, 265-292.
18. Melvin Fitting, *First Order logic and automated theorem proving*. Graduate Texts in Computer Science. Springer Verlag, Berlin, second edition, 1996.
19. Hylo page. <http://www.hylo.net/>, Papers, Bibliography, Course and Slides....
20. O.Kutz, *E-Connections and Logics of Distance*. Ph.D. thesis, University of Liverpool, 2004.
21. Ahti-Veikko Pietarinen, Tero Tulenheimo. An introduction to IF logic, Course material, 16th European Summer School in Logic, Language and Information, Université Henri Poincaré, Nancy, France.
22. A. Prior, *Past, Present and Future*. Oxford: Clarendon Press, 1967.
23. A. Prior, *Papers on Time and Tense*. Oxford: Clarendon Press, 1968.

24. Evangelos Tzanis, Algebraizing Hybrid Logic. Msc Thesis, University of Amsterdam, 2005. (<http://www.illc.uva.nl/Publications/ResearchReports/MoL-2005-03.text.pdf>)
25. Yoad Winter, Functional Quantification. Research on Language and Computation 2:331-363,2004.