Hybrid Logic with Operations on Nominals

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

Abstract. In this paper we introduce and study an extension of hybrid logic in which the set of nominals is endowed with an algebraic structure. Among other things we present an adaption of the notion of a bisimulation to this language, and a sound and complete tableau calculus.

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 $i, j, ...$) behave similar to ordinary proposition letters, expect that nominals are true uniquely at a world. In other words, a nominal names a state by being true there and nowhere else. An example of a formula involving nominals is $\Box\Box i \rightarrow \neg\Box i$. 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 $p$ 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 $\mathcal{H}(\@)$.

In this paper we introduce and study an extension of hybrid logic in which the set of nominals is endowed with an algebraic structure. In other words we add operators on nominals. The main motivation of the paper comes from [3]: 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. Roughly speaking we try to give to nominals a structure, we study the case where this structure is an algebraic structure. As far as we know, the possible algebraic structure of nominals has not been studied in the context of hybrid logic before. We should note that in [7] 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.

The paper is as follows. In section 2 we name a version of the basic hybrid logic $\mathcal{H}(\@)$ and we call it $\mathcal{HLO}$ (Hybrid Logic with Operators on Nominals) henceforth. We set up the syntax and we give the notions of Kripke model and validity to it. We give a bisimulation of models for this language providing also that validity is invariant under bisimulation. While in section 3 we deal with tableau proof methods applied to $\mathcal{HLO}$.
2 A brand new hybrid logic $\mathcal{HLON}$

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 $\rho$ is a function $A \rightarrow N$. Elements of $A$ are called functions symbols; the function $\rho$ 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: $ANOM = \{i, i_1, \cdots\}$. Let $\sigma$ be an algebraic similarity type.

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

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

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

Definition 3. $\mathcal{HLON}$ MODELS.

A model $M$ for $\mathcal{HLON}$ is a $M = (W, R_\circ, (F^f)_{f \in \sigma}, V)$ such that, $W \neq 0$, a set of possible worlds, $R_\circ$ 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 operation symbol of $W$. Note that $F^f : W^n \rightarrow W$ and $V : \Phi \rightarrow \text{Pow}(W)$ and $V$ is such that for all nominals $\alpha \in NOM$, $V(\alpha)$ is singleton of $W$. In more detail:

- If $\alpha \in NOM$ then: $V(\alpha) = \{w\}$;
- If $\alpha \in CNOM$ then: $V(f(\alpha_1, \cdots, \alpha_n)) = \{F^f(V(\alpha_1), \cdots, V(\alpha_n))\}$.

We call the elements of $M$ states or worlds, $R_\circ$ accessibility relations, $F^f$ operation symbols of $W$ and $V$ the valuation. We call the triple $(W, R_\circ, (F^f)_{f \in \sigma})$ the algebraic frame underlying the model.

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

$$\varphi := \alpha \mid p \mid \neg \varphi \mid \varphi ightarrow \psi \mid \Diamond \varphi \mid \Box_\alpha \varphi$$

where $\alpha \in NOM$. 
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 $\mathcal{HLON}$

Let a model $M = (W, R, (F^f)_{f \in \sigma}, V)$ for $\mathcal{HLON}$ and a valuation $V$ which assigns an element of the domain to every nominal.

\[
\begin{align*}
M, w \models i & \iff w \in V(i), \text{ where } i \in \text{ANOM} \\
M, w \models f(\alpha_1, \ldots, \alpha_n) & \iff \text{ for some } v_1, \ldots, v_n \in W \text{ with } F^f(v_1, \ldots, v_n) = w \\
& \quad \text{we have for each } l, M, v_l \models \alpha_l. \\
M, w \models p & \iff w \in V(p), \text{ where } p \in \text{PROP} \\
M, w \models @_i \varphi & \iff M, v \models \varphi \text{ where } V(\alpha) = \{v\} \\
M, w \models \varphi \rightarrow \psi & \iff M, w \not\models \varphi \text{ or } M, w \models \psi \\
M, w \models \neg \varphi & \iff M, w \not\models \varphi \\
M, w \models \diamond \psi & \iff \text{ if there is a } v \in W \text{ such that } R_0(v, w) \text{ and } M, v \models \varphi.
\end{align*}
\]

$(M, w)$ 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 $R_0$ and to one binary function $F$. In other words our hybrid model is: $M = (W, R_0, F^f, V)$ or better $M = (W, R, F, V)$.

3 Bisimulation and Elementary Equivalence

Bisimulation is a useful notion in modal logic. Bisimulation is a relation between the universes of two models in which related states have identical atomic information and matching transition possibilities. Roughly speaking the notion of bisimulation defines an equivalence relation between models. In the case of $\mathcal{HLON}$ models the notion of identical atomic information means satisfying the same sentences. A well known result in modal logic is that if two pointed models are bisimilar, then they satisfy the same sentences (see in [4]). In this section we show that such a result holds for $\mathcal{HLON}$ as well.

Definition 6. Bisimulations for $\mathcal{HLON}$

Let two $\mathcal{HLON}$ models $M = (W, R, F, V)$ and $M' = (W', R', F', V')$ be given. A non empty binary relation $\sim \subseteq W \times W'$ is a bisimulation when the following holds:

- (prop) if $w \sim w'$ then $w \in V(p)$ iff $w' \in V'(p)$, for all $p \in \text{ANOM} \cup \text{PROP}$.
- (fun-forth) if $w \sim w'$ and there are $u_1, u_2 \in W$ such that $F(u_1, u_2) = w$, then there are $v_1, v_2 \in W'$ such that for all $l$, $u_l \sim v_l$ and $F'(v_1, v_2) = w'$.
- (fun-back) A similar condition from $M'$ to $M$. 

$M, w \models = i$ iff $w \in V(i)$, where $i \in \text{ANOM}$

$M, w \models f(\alpha_1, \ldots, \alpha_n)$ iff for some $v_1, \ldots, v_n \in W$ with $F^f(v_1, \ldots, v_n) = w$
we have for each $l, M, v_l \models \alpha_l$.

$M, w \models p$ iff $w \in V(p)$, where $p \in \text{PROP}$

$M, w \models @_i \varphi$ iff $M, v \models \varphi$ where $V(\alpha) = \{v\}$

$M, w \models \varphi \rightarrow \psi$ iff $M, w \not\models \varphi$ or $M, w \models \psi$

$M, w \models \neg \varphi$ iff $M, w \not\models \varphi$

$M, w \models \diamond \psi$ iff if there is a $v \in W$ such that $R_0(v, w)$ and $M, v \models \varphi$. 

$(M, w)$ is called a pointed model. This is a model with a designated world, called its point, which is taken to be the actual world.
– (forth) if \( w \sim w' \) and \( Rwv \), then there is \( v' \in W' \) such that \( R'w'v' \) and \( v \sim v' \).

– (back) A similar condition from \( M' \) to \( M \).

– (\( \oplus \)): if \( w \in V(\alpha) \) and \( w' \in V(\alpha) \) for some \( \alpha \in \text{NOM} \), then \( w \sim w' \).

Since \( @ \) is self-dual we can collapse the back and forth clauses for this operator into one. Prop. forth and back are the usual conditions for bisimulation. We added fun-forth and fun-back in order to deal with functional sentences.

If there is a bisimulation between \( M \) and \( M' \) linking \( w \) and \( w' \) then we write \( M, w \leftrightarrow M', w' \).

**Theorem 1.** For all models \((M, w)\) and \((M', w')\) and for all sentences \( \phi \), if \((M, w) \leftrightarrow (M', w')\) then \( M, w \models \phi \) iff \( M', w' \models \phi \).

**Proof.** By induction \( \phi \). Suppose \( M, w \models \phi \leftrightarrow M', w' \models \phi \).

– The case where \( \varphi \in \text{PROP} \cup \text{ANOM} \) follows from clause (prop).

– For formulas \( \varphi \in \text{CNOM} \) we want to show that \( M, w \models f(\alpha_1, \alpha_2) \) iff \( M', w' \models f(\alpha_1, \alpha_2) \). Suppose \( M, w \models f(\alpha_1, \alpha_2) \). This means that there are worlds \( x, y \in W \) with \( M, x \models \alpha_1 \) and \( M, y \models \alpha_2 \). Note that \( F(x, y) = w \). Using the definition of bisimulation we know that there are worlds \( x', y' \in W' \) such that \( M, x' \models \alpha_1, M, y' \models \alpha_2, x \sim x' \) and \( y \sim y' \). We also know that \( F'(x', y') = w' \), thus \( M', w' \models f(\alpha_1, \alpha_2) \). For the converse direction use the clause (fun-back).

– As for formulas \( \diamond \varphi \), we have \( M, w \models \diamond \varphi \) iff there exists a \( u \) in \( M \) such that \( Rwu \) and \( M, u \models \varphi \). As \( w, w' \) are bisimilar by clause (forth) we know that there is a world \( u' \in M' \) such that \( Rw'u' \) and \( u \sim u' \). By the induction hypothesis, \( M', u' \models \varphi \), hence \( M', w' \models \diamond \varphi \).

– For the converse direction use the clause (back).

### 4 Tableau Methods

In this section we will give a completeness proof by constructing Hintikka sets. That is, extending the basic tableau methods of \( H(\oplus) \) (more details in [2]) we suggest a reasoning machinery for systems which combine relational and algebraic formalisms.

**Definition 7.** A substitution is a function from atomic nominals to nominals, which is identity on all but finitely many atomic nominals.

We typically use Greek letters such as \( \theta, \rho, \sigma \) for substitutions. If \( \theta \) is a substitution and \( i \) is an atomic nominal, then we indicate the application of \( \theta \) to \( i \) by \( \theta(i) \) as usual. We also extend substitutions homomorphically to nominals. If \( t \) is a nominal and \( \theta \) is a substitution, we indicate by \( t\theta \) the application of \( \theta \) to \( t \). This is defined inductively as follows: If \( t \) is an atomic nominal then \( t\theta = \theta(t) \). Otherwise, \( t \) is of the form \( f(t_1, t_2, ..., t_n) \) (a complex nominal) for some \( n \) (possibly zero). Then \( t\theta \) is \( f(t_1\theta, t_2\theta, ..., t_n\theta) \). Thus \( t\theta \) is \( t \) with all atomic nominals
replaced by nominals as specified by \( \theta \). We often write a substitution as a set \( \{ i_1 \mapsto s_1, \ldots, i_n \mapsto s_n \} \), indicating that the nominal \( i_k \) is mapped to \( s_k \) by the substitution. Given any \( \theta \), we call \( t\theta \) an instance of \( t \). Also, if \( @t\theta \) where \( t \) and \( u \) are atomic nominals and \( \theta \) is a substitution, we call \( @t\theta \) an instance of \( @t\theta \).

**Example 1.** (Substitution) If \( @x.1x \) then \( @((x \cdot 1) \cdot x) \).

Since in \( HLON \) we deal with function symbols also the following rules are needed:

\[
\frac{t \rightarrow s}{f(...,t,...) \rightarrow f(...,s,...)}\quad \text{Repl}
\]

\[
\frac{t \rightarrow s \rightarrow t}{t \theta \rightarrow s \theta \rightarrow t \theta}\quad \text{Sub}
\]

Where \( \theta \) is a substitution.

We continue by giving a completeness proof by constructing Hintikka sets. But we should make clear, before we proceed any further, that our tableau calculus are the usual hybrid tableau rules given in [2], [3] plus these two new rules: Repl and Sub. Note that the old hybrid tableau rules and the new ones apply to nominals, that is, both to atomic nominals and to complex ones.

**Definition 8. Hintikka Set**: A set of satisfaction statements \( \Sigma \) is a Hintikka set if it satisfies the following conditions.

1. For all atoms \( \pi \in \Phi \) and all nominals \( s \), if \( @s \pi \in \Sigma \) then \( \neg @s \pi \notin \Sigma \).
2. For all nominals \( s \) and \( t \), if \( @s \rightarrow t \) then \( \neg @s \rightarrow t \notin \Sigma \).
3. If a nominal \( s \) occurs in any formula in \( \Sigma \), then \( @s s \in \Sigma \).
4. If \( \Sigma \) contains a formula that one of the branching rules can be applied to, then it contains at least one of the formulas obtainable by making this application.
5. If \( \Sigma \) contains a pair of formulas that one of the binary rules can be applied to, then it contains all the formulas obtainable by making this application.
6. If \( \Sigma \) contains a formula that one of the existential rules can be applied to, then for some nominal \( i \) it also contains the formulas that would be obtained by applying that rule to that formula using \( i \) as the new nominal \( \tau \).
7. For any other rule, if \( \Sigma \) contains a formula that one of the rules applies to, then it contains all the formulas obtainable by making this application.

**Definition 9.** Let \( \Sigma \) be a Hintikka set, we define \( \text{NOM}(\Sigma) \) to be:

\[
\{ i | i \text{ is a nominal that occurs in some formula in } \Sigma \}
\]

and define a binary \( \sim \) on \( \text{TERM}(\Sigma) \) by \( i \sim j \iff @i,j \in \Sigma \). Clearly \( \sim \) is an equivalence relation. If \( k \in \text{NOM}(\Sigma) \), then \( |k| \) is the equivalence class of \( k \) under \( \sim \).

**Remark 1** Item 3 in the definition of Hintikka sets is an analog for first order reflexivity rule for \( = \), while closure under \( \text{SYM} \) and \( \text{NOM} \) ensures symmetry and transitivity.
Lemma 1. Let $\Sigma$ be a Hintikka set and assume that $i, j \in \text{NOM}(\Sigma)$. Then the following assertions are equivalent:

1. $i \sim_{\Sigma} j$
2. For every formula $\phi$, $@_i, \phi \in \Sigma$ iff $@_j, \phi \in \Sigma$

Proof. We start assuming $i \sim_{\Sigma} j$. Then $@_i, j \in \Sigma$ and hence $@_i, j \in \Sigma$. Since Hintikka sets are closed under $\text{NOM}$ then we have: $@_i, \phi \in \Sigma$ iff $@_k, \phi \in \sigma$. For the converse, assume that $@_i, \phi \in \Sigma$ iff $@_j, \phi \in \sigma$. $j$ occurs in $\Sigma$ we have that $@_j, j \in \Sigma$. Thus, if $\phi$ is $j$ we have that $@_i, j \in \Sigma$ then $i \sim_{\Sigma} j$.

Definition 10. Model induced by a Hintikka set: Given any Hintikka set $\Sigma$, there is a model $\mathcal{M}_{\Sigma} = (W, R, V, F)$ — the model induced by $\Sigma$ — such that the following hold.

1. $W = \{ |k|_{\Sigma} | k \in \text{NOM}(\Sigma) \}$.
2. $|\alpha|_{\Sigma} R |\alpha|_{\Sigma}$ iff $@_{\alpha, \phi} \in \Sigma$.
3. $F(|k_1|_{\Sigma}, |k_2|_{\Sigma}) = |\alpha|_{\Sigma}$ iff $@_{\alpha, f(k_1, k_2)} \in \Sigma$
4. $|\alpha|_{\Sigma} \in V(p)$ iff $\alpha, p \in \Sigma$
5. $|\alpha|_{\Sigma} \in V(j)$ iff $@_{\alpha, j} \in \Sigma$. Note $j$ is a nominal.

Anticipating the next lemma, we call such $\mathcal{M}_{\Sigma}$ the standard model induced by $\Sigma$.

Lemma 2. Let $\Sigma$ be a Hintikka set. Then any model $\mathcal{M}_{\Sigma} = (W, R, V, F)$ of the kind just above defined is an model induced by $\Sigma$.

Proof. In this lemma we should prove that $R, V$ and $F$ are well defined.

To show that $R$ is well defined we need to show that for all $i, j, k, l \in \text{NOM}(\Sigma)$, if $i \sim_{\Sigma} k$ and $j \sim_{\Sigma} l$ then $F(|i|, |j|) = |\alpha| \implies F(|k|, |l|) = |\alpha|$. That is, we need to show that $F$ is function. Suppose that $i \sim_{\Sigma} k$, $j \sim_{\Sigma} l$ and $F(|i|, |j|) = \alpha$. That is, we want to show that $F(|k|, |l|) = |\alpha|$. By definition this means that $@_{\alpha, f(i, j)} \in \Sigma$, hence as $i \sim_{\Sigma} k$ and $j \sim_{\Sigma} l$ then by the replacement rule of $\mathcal{R}$: $@_{\alpha, f(k, l)} \in \Sigma$. But since $@_{\alpha, f(k, l)} \in \Sigma$, we have by item 3 that $F(|k|, |l|) = \alpha$ as required.

By the same way, $R$ is well defined when for all $i, j, k, l \in \text{NOM}(\Sigma)$ if $i \sim_{\Sigma} k$ and $j \sim_{\Sigma} l$ then $|i| R |j|$ implies $|k| R |l|$. Let us assume $i \sim_{\Sigma} k$, $j \sim_{\Sigma} l$ and $|i| R |j|$. Since $|i| R |j|$ we have $@_{\alpha, j} \in \Sigma$, hence as $i \sim_{\Sigma} k$ it follows that $@_{k, \alpha}$. Also, since $|j| R |l|$ we also have that $@_{j, \alpha} \in \Sigma$. Then by the bridge rule we have:

Since $@_{k, \alpha} \in \Sigma$, $@_{j, \alpha} \in \Sigma$ then $@_{k, \alpha}$. Therefore, $|k| R |l|$ as required.

It is straightforward to see that $V$ is well defined. But we need to show that for all nominals $\alpha$, $V(\alpha)$ is a singleton. As we will see there is no need to study two cases, for atomic nominals and complex ones. Let us assume that $\alpha$ occurs in $\Sigma$. Then $V(\alpha)$ contains $|\alpha|$, and as consequence $@_{\alpha, \alpha} \in \Sigma$. Now, let us assume that $|\beta| \in V(\alpha)$. Then $@_{\alpha, \beta} \in \Sigma$, that is $\beta \sim_{\Sigma} \alpha$ which means that $|\beta| = |\alpha|$. So, for all nominals $\alpha$, $V(\alpha)$ is a singleton subset of $W$. 

Hybrid Logic with Operations on Nominals 179
Lemma 3. Truth lemma Let $\Sigma$ be a Hintikka set and let $M_\Sigma$ be the model induced by $\Sigma$. Then for all terms $\alpha$ and formulas $\phi$, the following hold.

1. If $@\alpha\phi \in \Sigma$ then $M_\Sigma, |\alpha| \models \phi$
2. If $\neg@\alpha\phi \in \Sigma$ then $M_\Sigma, |\alpha| \not\models \phi$

That is, every formula in $\Sigma$ is satisfied in $M_\Sigma$.

Proof. The proof follows the lines of [2], so we just give two clauses leaving the remainder to the reader:

- $\phi$ has the form $f(s_1, s_2)$. Suppose $@\alpha f(s_1, s_2)$. Using the condition 3 of Lemma 1 we know that $F(|s_1|, |s_2|) = |\alpha|$. By induction hypothesis we get $M, |\alpha| \models f(s_1, s_2)$.

- $\phi$ is of the form $@l\psi$, $l$ is a nominal. Suppose $@s_\phi \in \Sigma$. Then $@s_\alpha @l\psi \in \Sigma$. Note that $s$ is a nominal. By closure under the tableau rules for $\@$, we can infer that $@l\psi \in \Sigma$. By induction hypothesis, $M, |l| \models \psi$. Using the fact that the denotation of $l$ in $M$ is $|l|$, concluding that $M, |s| \models @l\psi$

Suppose $\neg@s_\phi \in \Sigma$. Then $\neg@s_\alpha @l\psi \in \Sigma$. By closure under the tableau rules for $\@$, we can infer that $\neg@l\psi \in \Sigma$. By induction hypothesis, $M, |l| \not\models \psi$. Using the fact that the denotation of $l$ in $M$ is $|l|$, we conclude that $M, |s| \not\models @l\psi$

5 Acknowledgments

I am indebted to Prof. dr. Peter van Emde Boas and Dr. Balder ten Cate for their assistance and guidance throughout. I am grateful to Prof. Stathis Zachos for his advice and encouragement. I would like to thank an anonymous referee for his comments. I am grateful for the financial support from the Association of Symbolic Logic and the Department of Computer Science of the University College of London, which made my participation in this symposium possible.

References