Lietuvos matematikos rinkinys
Proc. of the Lithuanian Mathematical Society, Ser. A
Vol. **53**, 2012, 90–95

ISSN 0132-2818
www.mii.lt/LMR/

On decidability of pure hybrid logic

Stanislovas Norgėla

Faculty of Mathematics and Informatics, Vilnius University
Naugarduko 24, LT-03225 Vilnius
E-mail: stasys.norgela@mif.vu.lt

Abstract. In this paper we study a decidable class of pure Hybrid Logic. To prove decidability we use terminating sequent for formulae of $\mathcal{H}(@)$ logic.

Keywords: hybrid logic, decidability.

Introduction

Hybrid Logic $\mathcal{H}(@,\downarrow)$ extends Multimodal Logic K by introducing new operators $@,\downarrow$ and new kind of atomic formulae – nominals. Nominals have an important semantical property: each of them is satisfied only in one node of any model. A statement *formula F is true in node i* is denoted by formula $@_i F$. The $\downarrow$ binder binds variables to the current point. For example, the formula $\downarrow x.\Diamond x$ is true at the point i iff i is related to itself.

Let PROP be a set of propositional symbols $\{p,q,\ldots\}$, NOM a set of nominals $\{i,j,\ldots\}$, REL a set of relational symbols $\{R_1,R_2,\ldots\}$, VNOM a set of nominal variables $\{x,y,\ldots\}$. We define the hybrid multimodal language to be the following collection of formulae:

$$WFF := p|i|x|\neg F|F\vee G|F\&G|\Diamond_R F|\Box_R F|@_i F|\downarrow x.F,$$

where $p\in$ PROP, $i\in$ NOM, $R\in$ REL, $x\in$ VNOM, $F,G\in$ WFF. If REL contains only one element, instead of writing $\Box_R,\Diamond_R$ we write $\Box,\Diamond$.

One of the applications of Hybrid Logic is semistuctured databases, where data is organized in a finite graph [4]. Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents. Data stored in XML format could be viewed as a finite graph (usually as a tree). Hybrid Logic formulae could be used as a query language for such data. Why Hybrid Logic? First of all, nodes and edges in graph constructed from XML can have names. Modal Logics are too week to express such queries (e.g. we cannot express irreflexivity). One needs to deal with nodes explicitly. Secondly, a query written as Hybrid Logic formula is simpler, more natural and has an ability to reach a node without specifying any additional structures (which cannot be avoided in Xpath or XML-query).

A formula is *pure* if it contains no propositional variables. A Hybrid Logic is pure if it contains only pure formulas. A formula is in *negation normal form* if the negation symbol appears only in front of atomic subformulas. Each hybrid formula is equivalent to a hybrid formula in negation normal form. Transformation to negation normal form uses the fact that operators $\Box_R,\Diamond_R$ are duals, i.e. $\neg\Box_R F\equiv\Diamond_R\neg F$, $\neg\Diamond_R F\equiv\Diamond_R\neg F$, and operators @, $\downarrow$ are self-duals (see, for example, [1]), i.e. $\neg @_i F \equiv @_i \neg F$, $\neg \downarrow x.F \equiv \downarrow x.\neg F$.

A set of formulae without $\downarrow$, is denoted by $\mathcal{H}(@)$. Undecidability of $\mathcal{H}(@, \downarrow)$ logic is proved in [1, 2, 5, 6, 8]. Normally the proof of undecidability is based on an encoding of the $\aleph \times \aleph$ tiling problem. If we bounded degree of nodes in $\mathcal{H}(@, \downarrow)$ models, we would get a decidable problem [3, 6].

Two decidable classes are proved in [9]. They also include formulae of the form $\ldots \Box(\ldots \downarrow x.(\ldots \Box \ldots) \ldots) \ldots$, with additional constraints.

First class. It studies formulae of pure Hybrid Logic in negation normal form. The class of formulae, where $\Diamond$ stands in front of any positive literal, is decidable. Let us remind that literals are nominals and negated nominals. Literal occurrence in a formula is positive if there is no negation in front of it. Otherwise literal occurrence is called negative.

Second class. We say that a subformula G of a formula F of the form $G = \Box H$, $G = \Diamond H$ is x – minimal if there are no H subformulae of the form $\Box N$, $\Diamond N$ that contain all H subformulae x occurrences.

Suppose a formula F which contains $\downarrow x$ occurrences and contains no nominals (i.e. only nominal variables) is given. Suppose F satisfies a condition: every x – minimal subformula of the formula F does not contain a subformula of the form $\Diamond G$. Such class of formulae is decidable.

Formulae, which do not have any $\downarrow$-binder occurrences in the scope of another $\downarrow$-binder, are called $\downarrow$-*linear formulae*.

Definition 1. Let the rank of the $\downarrow$ occurrence be a number of modals operators whose scope contains that $\downarrow$ occurrence.

For example, $F = \Box(\Diamond\Box\Diamond G \& \Box \downarrow x.H)$. The rank of $\downarrow x$ is 2.

Suppose formula F contains only constants (nominals) $r, v_1, v_2, \ldots, v_n$ that satisfies a condition: in every F model node r does not have any incoming edges and nodes v_i are leaves, i.e. they do not have any outgoing edges. We will call such formulae – *formulae with XML constants.* Usually only leaves have names in graphs corresponding to XML documents. Therefore, constants described above will be called XML constants.

In B. ten Cate and M. Franceschet work [6] it is proved that a class of $\downarrow$-linear Hybrid Logic formulae, which do not have $\downarrow$ occurrences with rank higher than two, is undecidable.

In this work we will prove the decidability of a such class (for simplicity let us denote it by A): the class of *$\downarrow$-linear pure Hybrid Logic $\mathcal{H}(\downarrow)$ formulae in negation normal form with XML constants is decidable if all $\downarrow$ occurrences have the same rank.*

We say that a sequent for formulae of $\mathcal{H}(@)$ logic is terminating if its all proof-search trees are finite.

1 Decidable class

First we recall sequent calculus for pure Hybrid Logic $\mathcal{H}$ presented in [9]. We will refer to this calculus as H.

Calculus axioms:

$$\Gamma, @_s \neg s$$

Rules:

$$\frac{\Gamma}{\Gamma, F} \qquad (\&)\ \frac{\Gamma, @_s F, @_s G}{\Gamma, @_s(F\&G)} \qquad (\vee)\ \frac{\Gamma, @_s F \quad \Gamma, @_s G}{\Gamma, @_s(F \vee G)}$$

$$(\Diamond)\ \frac{\Gamma, @_s\Diamond F, @_s\Diamond t, @_t F}{\Gamma, @_s\Diamond F} \quad t \text{ is new}$$

$$(\Box)\ \frac{\Gamma, @_t F, @_s\Box F, @_s\Diamond t}{\Gamma, @_s\Box F, @_s\Diamond t}$$

$$(Simp)\ \frac{\Gamma, @_t F}{\Gamma, @_s @_t F} \qquad (Sub)\ \frac{[\Gamma]_t^s}{\Gamma, @_s t} \qquad (\downarrow)\ \frac{\Gamma, @_s F[s/w]}{\Gamma, @_s \downarrow w.F}.$$

Each formula is in negation normal form. $[\Gamma]_t^s$ means that s is replaced with t. $F[s/w]$ means that every occurrence of w is replaced by s in formula F.

We construct a new calculus, which first of all differs from $\mathcal{H}$, since rules ($\Box$), ($\Diamond$) are replaced by a new rule ($\Diamond\Box$):

$$\frac{\Gamma, @_s\Diamond t, @_t F, @_t G_1, \ldots, @_t G_l, \Diamond^{N_0} F, @_s\Box^{N_1\cup\{t\}} G_1, \ldots, @_s\Box^{N_l\cup\{t\}} G_l}{\Gamma, @_s\Diamond^{N_0} F, @_s\Box^{N_1} G_1, \ldots, @_s\Box^{N_l} G_l} \quad (\Diamond\Box).$$

We start derivation by a sequent in which all $\Diamond$ and $\Box$ occurrences are given superscript $\varnothing$. A sequence $@_s\Box G_1, \ldots, @_s\Box G_l$ is a full list of formulae in conclusion. Let us call this calculus $\mathcal{H}'$. Calculi $\mathcal{H}_{\Diamond\Box}$ and $\mathcal{H}'$ for non-indexed formulae are equivalent (see [9]), i.e. sets of derivable sequents are equal. This also implies the completeness of calculus $\mathcal{H}'$. Indices are only used to describe history of term assignments (derivability is not affected). It is possible not to use indices. In that case before using rule ($\Diamond\Box$) we would have to look through the derivation tree from current sequent to the end-sequent. Below we will prove a theorem about a property of indexed formulae derivation tree and use corollary of the theorem to show that A is decidable.

Let C_T denote a set of all nominals in the considered sequent. We will search for a proof in $\mathcal{H}'$ using such tactics (we will call it *the main tactics*):

- We apply rules (&), ($\vee$), (Sub) as long as we can;
- We apply rule ($\Diamond\Box$), if the following conditions are satisfied:
  a) $s \notin C_T$. Conclusion of the rule application does not contain the same formulae as $\Diamond F, \Box G_1, \ldots, \Box G_l$ just indexed, i.e. formulae of the form $\Diamond^{M_0} F, @_i\Box^{M_1} G_1, \ldots, @_i\Box^{M_l} G_l \in \Gamma$, such that $M_0 \cap \cdots \cap M_l \neq \varnothing$ (i may not be equal to s);
  b) $s \in C_T$. Conclusion of the rule application does not contain formulae $\Diamond^{M_0} F, @_i\Box^{M_1} G_1, \ldots, @_i\Box^{M_l} G_l \in \Gamma$, such that $M_0 \cap \cdots \cap M_l = B$, $B \neq \varnothing$ and $s \in B$.

Again, we repeat the application of rules in the given order.

Theorem 1. *For every pure Hybrid Logic formula of the form* $@_s F$ ($\downarrow\notin F$, $@ \notin F$) *proof search following the main tactics in calculus* $\mathcal{H}'$ *will terminate after a finite number of steps iff* $@_s F$ *is derivable in calculus* $\mathcal{H}$.

Proof. Notice that the number of subformulae in the derivation tree starting with a modal operator, i.e. of the form $\Diamond F$, $\Box F$, is finite. All of them are subformulae of formulae from the end-sequent. Calculus $\mathcal{H}$ allows an infinite number of formulae of the form $@_i\Diamond F$, $@_i\Box F$ with the same $\Diamond F$, $\Box F$ in the derivation tree. Only @ indices are different. S. Cerrito and M. Cialdea Mayer give an example of a formula with an infinite derivation tree in [7]. This work introduces partial order of nominals found in derivation tree and adds additional constraint that nominals do not have descendants. We use a different idea. We show that a number of different subformulae of specific form is finite in derivation tree.

If we search for a proof in calculus $\mathcal{H}$, we will get the same derivation tree for the same combination $@_s\Diamond F$, $@_s\Box G_1, \ldots, @_s\Box G_n$, just newly introduced nominal will have a different name. In calculus $\mathcal{H}'$ with the main tactics we cannot use rule $(\Diamond\Box)$ for the same combination twice. The number of combinations of formulae under consideration is finite. Consequently after finite number of steps we will either get axioms in all branches or make sure that sequent is not derivable. Theorem is proved.

Corollary 1. *For every pure Hybrid Logic formula $@_sF$ ($\downarrow\notin F$, $@\notin F$) it is possible to assign a natural number l (l is a function of F), such that if $@_sF$ is derivable in calculus $\mathcal{H}$, the height of its derivation tree is not greater than l.*

Theorem 2. *The class A of pure Hybrid Logic formulae is decidable.*

Proof. Consider non-indexed variant of calculus $\mathcal{H}'$. The rule $(\Diamond\Box)$ will be

$$\frac{\Gamma, @_s\Diamond t, @_tF, @_tG_1, \ldots, @_tG_l, @_s\Box G_1, \ldots, @_s\Box G_l}{\Gamma, @_s\Diamond F, @_s\Box G_1, \ldots, @_s\Box G_l}\quad(\Diamond\Box).$$

Let us add another rule to the calculus.

$$\frac{\Gamma, @_s\Diamond t, @_tF, @_tG_1, \ldots, @_tG_l}{\Gamma, @_s\Diamond F, @_s\Box G_1, \ldots, @_s\Box G_l}\quad(\Diamond\Box)^*.$$

As we can see, the premise of the rule $(\Diamond\Box)^*$ does not contain the main formulae.

Suppose formula F is derivable. We can also find its derivation tree using the following tactics:

1. First of all we apply rules $(\wedge)$, $(\vee)$, (Sub) as long as we can.
2. We apply rule $(\Diamond\Box)^*$ as long as derivation tree does not contain any formula of the form $@_i \downarrow x.G$.

During every application of rule $(\Diamond\Box)^*$ we decrease the rank of formulae by one. We will prove that if formula is derivable, we will find its derivation tree following the tactics described above.

Suppose we have a sequent which satisfies conditions for applying rule $(\Diamond\Box)^*$ and every formula in a sequent has one of the forms: a) $@_i\Diamond j$, b) $@_i\Diamond G$ (G is not a nominal), c) $@_i\Box G$. i can be:

- name of a leaf node, i.e. $i = v_j$,
- name of the root node, i.e. $i = r$,
- newly introduced nominal in derivation tree.

Let us explain why we do not need to repeat main formulae in all of these three cases, i.e. why we are allowed to use rule $(\Diamond\Box)^*$.

1. If sequent has a formula of the form $@_{v_j}\Diamond G$, we take it as an axiom and terminate the branch. If sequent has formulae of the form $@_{v_j}\Box G$, we delete them. Let us remind that leaf nodes v_j do not have outgoing edges in the models which we consider.

2. If sequent has formulae of the form $@_r\Diamond G, @_r\Box H_1, \dots, @_r\Box H_l$, it is not necessary to repeat formulae $@_r\Box H_1, \dots, @_r\Box H_l$, because we will not be able to get new formulae of the form $@_r\Diamond u$ above in the derivation tree (except those, which we will get in the sequent just above the current sequent). To get $@_r\Diamond u$, we need $r = j$, but this would be considered as an axiom, since r cannot be reached from any other node (there are no loops).

3. If i is a newly introduced nominal, we cannot get formulae of the form $@_i\Diamond j$, where j is new, above in the derivation tree. To be precise, we cannot get other formulae of the form $@_i\Diamond j$ except those, which we have in the sequent just above the current sequent. This is because i is not met above in the derivation tree.

For the same reason, if there are formulae of the form $@_i\Box G$ and there are no formulae of the form $@_i\Diamond H$, we delete all $@_i\Box G$. As we can see, some of the formulae are deleted. Consequently, if we get an empty sequent, the end-sequent is not derivable.

When all ranks of formulae in a sequent become zero, we apply rule $(\downarrow)$ until the are no more occurrences of $(\downarrow)$. We get a sequent which satisfies conditions for the Theorem 1. Hence, we can indicate a natural number such that if a sequent is derivable, the height of the derivation tree is not greater than this number. Theorem is proved.

References

[1] C. Areces. *Logic engineering. The case of description and hybrid logics.* Phd thesis, University of Amsterdam, October 2000. LLC Dissertation Series DS-200-05.

[2] C. Areces, P. Blackburn and M. Marx. A road-map on complexity for hybrid logics. In J. Flum and M.R. Guez-Antalejo(Eds.), *Proceedings of the 8th Annual Conference of the EACSL,* Madrid, LNCS, 1683, pp. 307–321, 1999.

[3] C. Areces and B. ten Cate. *Hybrid Logics. Handbook of Modal Logic.* 2006.

[4] N. Bidoit, S. Cerrito and V. Thion. A first step towards modelling semistructured data in hybrid multi-modal logic. *J. Appl. Non-Class. Log.*, **14**(4):447–475, 2004.

[5] P. Blackburn and J. Seligman. Hybrid languages. *J. Log. Lang. Inf.*, **4**(3):251–272, 1995.

[6] B. ten Cate and M. Franceschet. On the complexity of hybrid logics with binders. *Lect. Not. Comp. Sci.*, **3634**:329–354, 2005.

[7] S. Cerrito and M. Cialdea Mayer. Terminating tableaux for $\mathcal{HL}(@)$ without loop-checking. *Technical report* IBISC-RR2007-07, Universit'e d'Evry Val d'Essonne, 2007.

[8] M. Marx. Narcissitis, stepmothers and spies. In *Proceedings of the International Workshop on Description Logics.* Toulouse, France, 2002.

[9] S. Norgėla and A. Šalaviejienė. Some decidable classes of formulas of pure hybrid logic. *Lith. Math. J.*, **47**(4):462–469, 2007.

REZIUMĖ

Apie grynosios hibridinės logikos išsprendžiamumą

S. Norgėla

Darbe nagrinėjama viena grynosios hibridinės logikos formulių klasė. Įrodomas jos išsprendžiamumas. Išsprendžiamumui įrodyti pasinaudojama sekvenciniu be ciklų skaičiavimu logikos $\mathcal{H}(@)$ formulėms.

Raktiniai žodžiai: hibridinė logika, išsprendžiamumas.