Polynomial Hierarchy Graph Properties in Hybrid Logic

F. M. Ferreira[a,1,*], C. M. Freire[b,c,2], M. R. F. Benevides[d,3], L. Menasché Schechter[e], A. T. Martins[a,4]

[a]*Departamento de Computação, Universidade Federal do Ceará, Caixa Postal 12166, Fortaleza – CE 60455-900, Brasil*

[b]*Department of Computer Science, University of Massachusetts Amherst, 140 Governors Drive, Amherst, MA 01003-9264, EUA*

[c]*Capes Foundation, Ministry of Education of Brazil, Caixa Postal 250, Brasília – DF 70040-020, Brazil*

[d]*Programa de Engenharia de Sistemas e Computação, Universidade Federal do Rio de Janeiro, Av. Horácio Macedo, 2030, Cidade Universitária, Caixa Postal 68511, Rio de Janeiro – RJ, Brasil*

[e]*Departamento de Ciência da Computação, Universidade Federal do Rio de Janeiro, Rua Athos da Silveira Ramos, 274, Cidade Universitária, Caixa Postal 68530, Rio de Janeiro – RJ, Brasil*

Abstract

In this article, we show that for each property of graphs $\mathcal{G}$ in the Polynomial Hierarchy (PH) there is a sequence $\phi_1, \phi_2, \ldots$ of formulas of the full hybrid logic which are satisfied exactly by the frames in $\mathcal{G}$. Moreover, the size of ϕ_n is bounded by a polynomial in n. These results lead to the definition of syntactically defined fragments of hybrid logic whose model checking problem is complete for each degree in the polynomial hierarchy.

Keywords: hybrid logics, polynomial hierarchy, graph properties

*Corresponding author.

Email addresses: `fran@lia.ufc.br` (F. M. Ferreira), `cibelemf@cs.umass.edu` (C. M. Freire), `mario@cos.ufrj.br` (M. R. F. Benevides), `luisms@dcc.ufrj.br` (L. Menasché Schechter), `ana@lia.ufc.br` (A. T. Martins)

[1]This research was partially supported by CAPES(DS, PROCAD 2010).

[2]Bolsista da Capes - Processo nº 5459-10-9.

[3]This research was partially supported by FAPERJ, CNPq and CAPES.

[4]This research was partially supported by CNPq(PQ, Casadinho 2008, Universal 2012 and 2010), CAPES(PROCAD 2009), CNPq/CAPES(Casadinho/PROCAD 2011).

1. Introduction

The use of graphs as a mathematical abstraction of objects and structures makes it one of the most used concepts in computer science. Plenty of problems people want to solve using computers have their inputs modelled by graphs, and such problems commonly involve evaluating some graph property. To mention a well-known example, deciding whether a map can be coloured with a certain number of colors equals to a similar problem on planar graphs [11, 15]. The applications of graphs in computer science are not restrict to modelling the input of problems. Graphs can be used in the theoretical framework in which some branches of computer science are formalized. This is the case, for example, in distributed systems, in which the model of computation is based on a graph [3, 13]. Again, properties of graphs can be exploited in order to obtain results about such models of computation.

In the last few decades, modal logics have attracted the attention of computer scientists working with logic and computation [5]. Among the reasons is the fact that modal logics often have interesting computability properties, like decidability [16, 9]. This is due to a lack of expressive power in comparison with other logics such as first-order logic and its extensions. Many modal logics present also good logical properties, like interpolation, definability, etc. Research in modal logic include augmenting the expressive power of logics using resources as fixed-point operators [6] or hybrid languages [2, 1]. Modal logics are particularly suitable to deal with graphs because standard semantics of most modal logics are based on structures called *frames*, which are essentially directed graphs.

In [4], hybrid logics are used to express graph properties, like being connected, Hamiltonian or Eulerian. Several hybrid logics and fragments were studied to define graph properties through the concept of *validity in a frame* (see Definition 9 below). Some graph properties, like being Hamiltonian, require a high expressive power and cannot be expressed by a single sentence in traditional hybrid logics. There are, however, sentences ϕ_n which can express such properties for frames of size n.

We are interested in expressing graph properties in NP, and more generally in the polynomial hierarchy (PH) [14], using hybrid logics (HL). The hybrid modal logics that we studied have low expressive power in comparison, for example, to second-order logic, hence we do not aim to associate to each graph property a single formula. Instead, we present, to each graph property a sequence of hybrid sentences ϕ_1, ϕ_2, ..., such that a graph of size n has the desired property iff ϕ_n is valid in the graph, regarded as a frame. In Section 2.4, we define the hybrid logic which we will study and define a prenex form for such logic. In Section 3, we show that, for any graph property, there is a sequence of sentences ϕ_1, ϕ_2, ... of the fragment of hybrid logic with nominals and the @ operator such that a graph of size n has the property iff ϕ_n is valid in the corresponding frame. However, the size of ϕ_n obtained is exponential on n. In Section 4, we show that, for graph properties in NP, and more generally in the polynomial hierarchy PH, there is such a sequence such that the size of the sentences is bounded by a polynomial in n. In Section 5, we show that, in general, the global modality cannot be disregarded. We also show how to obtain the results of the previous section for the fragment of HL without the global modality E and without nominals, provided that graphs are connected and with loops. In Section 6, we show fragments of HL whose model-checking problem is complete for each degree of the polynomial hierarchy based on the results of the other sections. This gives an alternative proof for the NP-hardness of the model-checking problem for the fragment FHL$\backslash\downarrow\Box\downarrow$ of full hybrid logic given in [7].

2. Preliminaries

In this section we introduce the basic definitions concerning classical and hybrid logics and computational complexity used in this article. In this article, graphs are finite, directed and possibly with loops.

2.1. First-Order Logic

The basic notation used here for classical logic follows the one in [10]. We work with first- and second-order languages without function symbols since they play no special role in the complexity results presented here. For instance, a symbol set S is a set of predicate and constant symbols. An S-structure is a pair $\mathfrak{A} = (A, \sigma)$, where A is a set, the domain of $\mathfrak{A}$ and σ is a map which associates an n-ary relation $\sigma(P) = P^{\mathfrak{A}} \subset A^n$ to each n-ary relation symbol $P \in S$ and an element $\sigma(c) = c^{\mathfrak{A}} \in A$ to each constant symbol $c \in S$.

An *S-term* is either a variable or a constant symbol in S. An *atomic first-order S-formula* is a first-order relation symbol in S applied to S-terms. For example, if R is a binary relation symbol in S and x_1 and c are S-terms, then Rx_1c is an atomic first-order S-formula. The set of *first-order S-formulas* is the least set which contains the atomic formulas and such that, if x is a first-order variable, ϕ and ψ are S-formulas, then $\neg\phi$, $(\phi \wedge \psi)$, $\exists x\phi$ and $\forall x\phi$ are S-formulas. The other propositional connectives $\vee$, $\rightarrow$, $\leftrightarrow$ are defined as usual. Given a symbol set $S = \{R_1, \ldots, R_l, c_1, \ldots, c_k\}$, we usually denote an S-structure $\mathfrak{A} = (A, \sigma)$ as $\mathfrak{A} = (A, R_1^{\mathfrak{A}}, \ldots, R_l^{\mathfrak{A}}, c_1^{\mathfrak{A}}, \ldots, c_k^{\mathfrak{A}})$, where $\sigma(R_i) = R_i^{\mathfrak{A}}$, $1 \leq i \leq l$ and $\sigma(c_j) = c_j^{\mathfrak{A}}$, $1 \leq j \leq k$. A *first-order S-interpretation* (or simply an *S-interpretation*) is a $\mathfrak{I} = (\mathfrak{A}, \beta)$ where $\mathfrak{A}$ is an S-structure and β is an assignment of first-order variables that maps first-order variables to elements in the domain A of $\mathfrak{A}$. Given an element $a \in A$ and a variable x, we define the assignment $\beta\frac{a}{x}$ such that $\beta\frac{a}{x}(x') = \beta(x')$ if $x' \neq x$ and $\beta\frac{a}{x}(x') = a$ if $x = x'$. We define the interpretation $\mathfrak{I}\frac{a}{x} = (\mathfrak{A}, \beta\frac{a}{x})$.

The semantics of first-order logic is defined as follows:

Definition 1 (Semantics of First-Order Logic). Let $\mathfrak{I} = (\mathfrak{A}, \beta)$ be an S-interpretation. We define $\mathfrak{I}(x) \in A$ for each variable x as $\mathfrak{I}(x) = \beta(x)$.

The satisfiability relation between S-interpretations and S-formulas is defined as:

- $\mathfrak{I} \models Rt_1 \ldots t_r$ iff $(\mathfrak{I}(t_1), \ldots, \mathfrak{I}(t_r)) \in R^{\mathfrak{A}}$, for each r-ary relation symbol R and S-terms $t_1, \ldots, t_r$;
- $\mathfrak{I} \models \neg\phi$ iff $\mathfrak{I} \not\models \phi$;
- $\mathfrak{I} \models (\phi \wedge \psi)$ iff $\mathfrak{I} \models \phi$ and $\models \psi$;
- $\mathfrak{I} \models \exists x\phi$ iff there is an $a \in A$ such that $\mathfrak{I}\frac{a}{x} \models \phi$;
- $\mathfrak{I} \models \forall x\phi$ iff for all $a \in A$ we have $\mathfrak{I}\frac{a}{x} \models \phi$.

2.2. Second-Order Logic

Second-order logic (SO) is defined as follows. Besides first-order variables, the alphabet of second-order logic has relation variables of all possible arities. An *atomic second-order S-formula* is either a first-order atomic S-formula or a relational variable applied to first-order variables and constants. For example, if X is a ternary relation variable and x_1, x_2 and x_3 are variables, then $Xx_1x_2x_3$ is a second-order atomic S-formula.[5]

[5]We often regard a second-order atomic formula as a first-order atomic formula too, in an extended symbol set where X is not regarded as a relation variable but a relation symbol. That is because, since in this formula there is no second-order quantification involved, it has essentially a first-order character.

The set of *second-order S-formulas* is the least set which contains the atomic formulas and such that, if x is a first-order variable, X is a second-order variable and ϕ and ψ are S-formulas, then $\neg\phi$, $(\phi \wedge \psi)$, $(\phi \vee \psi)$, $\exists x\phi$, $\forall x\phi$, $\exists X\phi$, $\forall X\phi$ are S-formulas. An existential SO formula is a formula of the form $\exists X_1 \ldots \exists X_k \phi$ where ϕ is an FO formula. Existential second-order logic ($\exists$SO) is the fragment of SO obtained by considering existential formulas only. The second-order (respectively first-order) language of graphs is the set of second-order (respectively first-order) S-formulas for $S = \{E\}$, where E is a binary relation symbol (intended to represent the edge relation).

A *second-order S-interpretation* is a pair $\mathfrak{I} = (\mathfrak{A}, \beta)$ where $\mathfrak{A}$ is an S-structure and β is an assignment of first- and second-order variables that maps first-order variables to elements in the domain A of $\mathfrak{A}$ and r-ary relation variables to r-ary relations on A. Given an r-ary relation variable X and an r-ary relation $\mathbf{X}$ on A, we define the assignment $\beta\frac{\mathbf{X}}{X}$ as $\beta\frac{\mathbf{X}}{X}(X') = \beta(X')$ if $X' \neq X$ and $\beta\frac{\mathbf{X}}{X}(X') = \mathbf{X}$ if $X = X'$. We define the interpretation $\mathfrak{I}\frac{\mathbf{X}}{X} = (\mathfrak{A}, \beta\frac{\mathbf{X}}{X})$.

The semantics of second-order logic is defined as follows:

Definition 2 (Semantics of Second-Order Logic). Let $\mathfrak{I} = (\mathfrak{A}, \beta)$ be an S-interpretation, X be a r-ary relation variable and $x_1, \ldots, x_r$ be first-order variables. We extend the satisfiability relation of first-order logic to second-order logic as follows:

- $\mathfrak{I} \models Xt_1 \ldots t_r$ iff $(\mathfrak{I}(t_1), \ldots, \mathfrak{I}(t_r)) \in \beta(X)$;
- $\mathfrak{I} \models \exists X\phi$ iff there is an $\mathbf{X} \in A^r$ such that $\mathfrak{I}\frac{\mathbf{X}}{X} \models \phi$;
- $\mathfrak{I} \models \forall X\phi$ iff for all $\mathbf{X} \in A^r$ we have $\mathfrak{I}\frac{\mathbf{X}}{X} \models \phi$.

Now we define the alternation hierarchy Π and Σ for second-order formulas.

Definition 3 (Π^1_n and Σ^1_n). We define the sets Π^1_n and Σ^1_n of second-order formulas as:

- $\Pi^1_0 = \Sigma^1_0$ the set of first-order formulas;
- Π^1_{n+1} = the set of formulas of the form $\forall X_1 \ldots \forall X_m \phi$ where $\phi \in \Sigma^1_n$;
- Σ^1_{n+1} = the set of formulas of the form $\exists X_1 \ldots \exists X_m \phi$ where $\phi \in \Pi^1_n$;

A second-order formula without free variables is called a *sentence*. We use the terms *second-order sentence* or *first-order sentence* to stress the fact that the formula is a second- or first-order formula. If α is a sentence and $\mathfrak{A}$ is a structure, then $\mathfrak{A} \models \alpha$ iff there is an interpretation $\mathfrak{I} = (\mathfrak{A}, \beta)$ such that $\mathfrak{I} \models \alpha$.

2.3. Polynomial Hierarchy

The basics of computational complexity and descriptive complexity can be found in [14] and [12] respectively. A well known result of descriptive complexity [12] is the correspondence between the Polynomial Hierarchy (PH) and the alternation hierarchy of second-order logic (with respect to finite models) [12], which follows from Fagin's Theorem [8]. Fagin's Theorem states that the class of NP problems coincides with the class of problems expressed in existential second-order logic. Extending Fagin's Theorem to the entire second-order Logic, we have that each problem in PH can be expressed by a second-order sentence.

There are several ways to define PH, for example using alternating Turing machines [12]. In this article we assume the definition presented in [14], which uses Turing machines with oracles to define PH.

A Turing machine with an oracle is a machine that has the special ability of guessing some specific questions. When a Turing machine has an oracle for a decision problem B, during its execution it can ask for the oracle if some instance of problem B is positive or negative. This is made in constant time, regardless of the size of the instance. We use the notation M^B to define a Turing machine with an oracle for a problem B. In a similar way, we define $\mathcal{C}^{\mathcal{B}}$, where $\mathcal{C}$ and $\mathcal{B}$ are complexity classes, as the class of problems solved by a Turing machine in $\mathcal{C}$ with an oracle in $\mathcal{B}$. See [14] for details.

Definition 4. Consider the following sequence of complexity classes. First, $\Sigma_0^p = \Pi_0^p = \mathrm{P}$, the class of problems solvable in deterministic polynomial time, and, for all $i \geq 0$,

1. $\Sigma_{i+1}^p = \mathrm{NP}^{\Sigma_i^p}$.
2. $\Pi_{i+1}^p = \mathrm{coNP}^{\Sigma_i^p}$.

We define the Polynomial Time Hierarchy as the class $\mathrm{PH} = \bigcup_{i \geq 0} \Sigma_i^p$.

For graphs, Fagin's Theorem can be stated as follows:

Theorem 1 (Fagin's Theorem [8]). *Let $\mathcal{G}$ be a graph property in NP. Then there is an existential second-order sentence ϕ in the language of graphs such that $G \in \mathcal{G}$ iff $G \models \phi$.*

Fagin's Theorem can be extended to PH as follows:

Theorem 2 ([12]). *Let $\mathcal{G}$ be a graph property in the polynomial hierarchy. Then there is a second-order sentence ϕ in the language of graphs such that $G \in \mathcal{G}$ iff $G \models \phi$. Specifically, a graph problem is in Σ_i^p (respectively Π_i^p) iff it can be expressed by a sentence in Σ_i^1 (respectively Π_i^1).*

In the next section, we present the basics of hybrid logics we will use in this article.

2.4. Hybrid Logic

In this section, we present the hybrid logic and the fragments which we will use. Hybrid modal logics extends classic modal logics by adding nominals and state variables to the language. Nominals and state variables behave like propositional atomic formulas which are true in exactly one state. Other extensions include the operators $\downarrow$ (binder) and @. The $\downarrow$ allows one to assign the current state to a variable state. This can be used to keep a record of the visited states. The @ operator allows one to evaluate a formula in the state assigned to a certain nominal or state variable.

Definition 5 (Hybrid Logic). The alphabet of the *Hybrid Logic* (HL) *with the $\downarrow$ binder* consists of a set PROP of countably many proposition symbols $p_1, p_2, \ldots$, a set NOM of countably many nominals $i_1, i_2, \ldots$, a set SVAR of countably many state variables $x_1, x_2, \ldots$, such that PROP, NOM and SVAR are disjoint, the boolean connectives $\neg$ and $\wedge$ and the modal operators $@_i$, for each nominal i, $@_x$, for each state variable x, $\Diamond$, $\Diamond^{-1}$, E and $\downarrow$. The language of the Hybrid Logic with the $\downarrow$ binder can be defined by the following BNF rule:

$$\alpha := \top \mid p \mid t \mid \neg\alpha \mid \alpha \wedge \alpha \mid \Diamond\alpha \mid \Diamond^{-1}\alpha \mid E\alpha \mid @_t\alpha \mid {\downarrow}x.\alpha.$$

An *atomic formula* is a proposition symbol, a nominal or a state variable. For each $C \subseteq \{@, \downarrow, \Diamond^{-1}, E\}$, we define HL($C$) to be the corresponding fragment. Here, we denote by FHL, standing for *Full Hybrid Logic*, for the entire Hybrid Logic with the $\downarrow$ binder, that is, FHL = HL$(@, \downarrow, \Diamond^{-1}, E)$. We also use HL($C$)\NOM and HL($C$)\PROP to refer to the fragments of HL(C) without nominals and propositional symbols respectively.

The standard boolean abbreviations $\to$, $\leftrightarrow$, $\vee$ and $\bot$ can be used with the standard meaning as well as the abbreviations of the dual modal operators: $A\phi = \neg E \neg \phi$, $\Box\phi = \neg\Diamond\neg\phi$ and $\Box^{-1}\phi = \neg\Diamond^{-1}\neg\phi$.

Formulas of hybrid modal logics are evaluated in *hybrid Kripke structures* (or *hybrid models*). These structures are built using *frames*.

Definition 6. A *frame* is a directed graph possibly with loops $\mathcal{F} = (W, R)$, where W is a non-empty set (finite or not) of states and R is a binary relation over W, i.e., $R \subseteq W \times W$.

Definition 7. A *(hybrid) model* for the hybrid logic is a pair $\mathcal{M} = (\mathcal{F}, \mathbf{V})$, where $\mathcal{F}$ is a frame and $\mathbf{V} : \mathrm{PROP} \cup \mathrm{NOM} \mapsto \mathcal{P}(W)$ is a valuation function mapping proposition symbols to subsets of W and nominals to singleton subsets of W, i.e, if i is a nominal then $\mathbf{V}(i) = \{v\}$ for some $v \in W$.

In order to deal with the state variables, we need to introduce the notion of *assignments*.

Definition 8. An *assignment* is a function g that maps state variables to states of the model, i.e., $g : \mathcal{S} \mapsto W$. We use the notation

$$g' = g\frac{x_1 \dots x_n}{v_1 \dots v_n}$$

to denote an assignment g' such that $g'(x) = g(x)$ if $x \notin \{x_1, \dots, x_n\}$ and $g'(x_i) = v_i$, otherwise.

The semantical notion of satisfaction is defined as follows:

Definition 9. Let $\mathcal{M} = (\mathcal{F}, \mathbf{V})$ be a model. The notion of *satisfaction* of a formula φ in a model $\mathcal{M}$ at a state v under an assignment g, notation $\mathcal{M}, g, v \Vdash \varphi$, can be inductively defined as follows:

$\mathcal{M}, g, v \Vdash p$ iff $v \in \mathbf{V}(p)$;

$\mathcal{M}, g, v \Vdash \top$ always;

$\mathcal{M}, g, v \Vdash \neg\varphi$ iff $\mathcal{M}, g, v \nVdash \varphi$;

$\mathcal{M}, g, v \Vdash \varphi_1 \wedge \varphi_2$ iff $\mathcal{M}, g, v \Vdash \varphi_1$ and $\mathcal{M}, g, v \Vdash \varphi_2$;

$\mathcal{M}, g, v \Vdash \Diamond\varphi$ iff there is a $w \in W$ such that vRw and $\mathcal{M}, g, w \Vdash \varphi$;

$\mathcal{M}, g, v \Vdash \Diamond^{-1}\varphi$ iff there is a $w \in W$ such that wRv and $\mathcal{M}, g, w \Vdash \varphi$;

$\mathcal{M}, g, v \Vdash E\varphi$ iff there is a $w \in W$ such that $\mathcal{M}, g, w \Vdash \varphi$;

$\mathcal{M}, g, v \Vdash i$ iff $v \in \mathbf{V}(i)$;

$\mathcal{M}, g, v \Vdash @_i\varphi$ iff $\mathcal{M}, g, d_i \Vdash \varphi$, where $\{d_i\} = \mathbf{V}(i)$;

$\mathcal{M}, g, v \Vdash x$ iff $g(x) = v$;

$\mathcal{M}, g, v \Vdash @_x\varphi$ iff $\mathcal{M}, g, d \Vdash \varphi$, where $d = g(x)$;

$\mathcal{M}, g, v \Vdash \downarrow x.\varphi$ iff $\mathcal{M}, g^x_v, v \Vdash \varphi$.

For each nominal i, the formula $@_i\varphi$ means that if $\mathbf{V}(i) = \{v\}$ then φ is satisfied at v. If $\mathcal{M}, g, v \Vdash \varphi$ for every state v, we say that φ is *globally true in the model* $\mathcal{M}$ *under the assignment* g ($\mathcal{M}, g \Vdash \varphi$). If $\mathcal{M}, g \Vdash \varphi$ for every assignment g, we say that φ is *globally true in the model* $\mathcal{M}$ ($\mathcal{M} \Vdash \varphi$) and if φ is globally true in all models $\mathcal{M}$ of a frame $\mathcal{F}$, we say that φ is *valid* in $\mathcal{F}$ ($\mathcal{F} \Vdash \varphi$). We say that two formulas ϕ and ψ are *equivalent* ($\phi \equiv \psi$) iff, for each model $\mathcal{M}$, assignment g and state v, $\mathcal{M}, g, v \Vdash \phi$ iff $\mathcal{M}, g, v \Vdash \psi$.

We will use the following definitions for model- and frame-checking problems for hybrid logic.

Definition 10. The *model-checking problem* has as input a model $\mathcal{M}$ and a formula ϕ and consists of deciding, for each state v of $\mathcal{M}$ and each assignment g on $\mathcal{M}$, whether $\mathcal{M}, g, v \Vdash \phi$.

The *frame-checking problem* has as input a frame $\mathcal{F}$ and a formula ϕ and consists of deciding, for each model $\mathcal{M} = (\mathcal{M}, \mathbf{V})$, assignment g on $\mathcal{M}$ and state v in $\mathcal{M}$ whether $\mathcal{M}, g, v \Vdash \phi$.

In the following, we define a prenex form for formulas in FHL and show that any formula in FHL has an equivalent in prenex form. We use this form to define classes of formulas whose model-checking problem is complete for the degrees of the polynomial hierarchy (see Section 6).

Definition 11 (Prenex Form). A formula ϕ in FHL is in *prenex form* iff $\phi = q_1 \dots q_n \psi$ where each q_i is A, E, $\Box$, $\Diamond$, $\Box^{-1}$, $\Diamond^{-1}$ or $\downarrow x$. for some x and ψ has no occurrence of $\downarrow$ and modalities in ψ occur only in front of atomic formulas.

The first-order language of graphs (the first-order language on the symbol set $\{E\}$, E a binary relation[6]) can be translated into the full hybrid logic and the full hybrid logic can be translated into the first-order language of graphs as well, and both translations preserve truth [2]. That is, full hybrid logic has the same expressive power as first-order logic. Using the *hybrid translation* from first-order logic to full hybrid logic and the *standard translation* from full hybrid logic back to first-order logic, we can prove a prenex normal form theorem for hybrid logic. We present below the hybrid translation HT and the standard translation ST. Let E be a binary relation symbol and for each propositional symbol p in the modal hybrid language let P be a monadic relation symbol. Let $\Psi = \{P | p \in \text{PROP}\} \cup \{c_i | i \in \text{NOM}\}$. Let $L^{\{E\} \cup \Psi}$ be the first-order language on the symbol set $\{E\} \cup \Psi$ and x and y fresh first-order variables. The functions $ST_x : \text{FHL} \to L^{\{E\} \cup \Psi}$ and $HT : L^{\{E\} \cup \Psi} \to \text{FHL}$ are defined as follows (see [2]):

$$
\begin{aligned}
HT(E(x,y)) &= @_x \Diamond y \\
HT(P(x)) &= @_x p \\
HT(x = y) &= @_x y \\
HT(\neg \phi) &= \neg HT(\phi) \\
HT(\phi \wedge \psi) &= HT(\phi) \wedge HT(\psi) \\
HT(\exists x \phi) &= E{\downarrow} x. HT(\phi)
\end{aligned}
$$

$$
\begin{aligned}
ST_x(s) &= x = s \\
ST_x(p) &= P(x) \\
ST_x(\neg \phi) &= \neg ST_x(\phi) \\
ST_x(\phi \wedge \psi) &= ST_x(\phi) \wedge ST_x(\psi)
\end{aligned}
$$

[6] We use the symbol E to represent both the global existential modality and the binary relation symbol of the graph language. It is clear from the context which one is the case.

$$\begin{aligned}
ST_x(\Diamond\phi) &= \exists y(R(x,y) \wedge ST_y(\phi)) \\
ST_x(\Diamond^{-1}\phi) &= \exists y(R(y,x) \wedge ST_y(\phi)) \\
ST_x(E\phi) &= \exists y(ST_y(\phi)) \\
ST_x(@_s\phi) &= \exists y(y = s \wedge ST_y(\phi)) \\
ST_x(\downarrow z.\phi) &= \exists z(z = x \wedge ST_x(\phi))
\end{aligned}$$

The function $ST_y : \text{FHL} \to L^{\{E\}\cup\Psi}$ is defined exactly like ST_x by changing x for y, and vice versa, everywhere in the definition of ST_x above. Given a hybrid model $\mathcal{M} = (\mathcal{F}, \mathbf{V})$ we can associate a first-order structure $\mathcal{M}' = (\mathcal{F}, \{\mathbf{P}|p \in \text{PROP}\}, \{\mathbf{i}|i \in \text{NOM}\})$ where $\mathbf{P} = \mathbf{V}(p)$ and $\mathbf{i} = g(i)$. In the following, we will use the same symbol to refer both to the hybrid model and the corresponding first-order structure.

Lemma 1 ([7, 1]). *Let α be a formula in* FHL *where the variable x does not occur and ϕ a formula of first-order logic. Let $\mathcal{M}$ be a model, g an assignment of variables, w a state in $\mathcal{M}$ and $g\frac{x}{w}$ the assignment of variables that maps x to w and $y \neq x$ to $g(y)$. Then we have*

1. $\mathcal{M}, g, w \Vdash \alpha$ *iff* $(\mathcal{M}, g\frac{x}{w}) \models ST_x(\alpha)$.
2. $(\mathcal{M}, g) \models \phi$ *iff* $\mathcal{M}, g \Vdash HT(\phi)$.

It follows from Lemma 1 that the modality $\Diamond^{-1}$ can be eliminated by translating a hybrid formula into FO using the standard translation and back to FHL using the hybrid translation. Hence, we do not need to consider it in the proofs below.

In the following, we will show a series of lemmas about hybrid logic that will be used in the following sections.

Using the standard and hybrid translation, the definition of the satisfaction relation and Lemma 1, we have the following equivalence in hybrid modal logic.

Lemma 2. *Let α and β be formulas in* FHL. *The following equivalences hold in* FHL*:*

1. $(A\alpha \wedge \beta) \equiv\ \downarrow x.A(\alpha \wedge @_x\beta)$, *$x$ not occurring free in α or β;*
2. $(E\alpha \wedge \beta) \equiv\ \downarrow x.E(\alpha \wedge @_x\beta)$, *$x$ not occurring free in α or β;*
3. $(A\alpha \vee \beta) \equiv\ \downarrow x.A(\alpha \vee @_x\beta)$, *$x$ not occurring free in α or β;*
4. $(E\alpha \vee \beta) \equiv\ \downarrow x.E(\alpha \vee @_x\beta)$, *$x$ not occurring free in α or β;*

5. $\neg{\downarrow}x.\alpha \equiv {\downarrow}x.\neg\alpha$;
6. $@_x E\alpha \equiv E\alpha$;
7. $@_x A\alpha \equiv A\alpha$;
8. $(({\downarrow}x.\alpha) \wedge \beta) \equiv\ {\downarrow}x.(\alpha \wedge \beta)$, *$x$ not occurring free in β;*
9. $(({\downarrow}x.\alpha) \vee \beta) \equiv\ {\downarrow}x.(\alpha \vee \beta)$, *$x$ not occurring free in β;*
10. $(\Diamond\alpha \wedge \beta) \equiv {\downarrow}x.E{\downarrow}y.[(@_x\Diamond y \wedge @_y\alpha) \wedge @_x\beta]$, *$x$ and y not occurring free in α or β;*
11. $(\Box\alpha \wedge \beta) \equiv {\downarrow}x.A{\downarrow}y.[(@_x\Diamond y \to @_y\alpha) \wedge @_x\beta]$, *$x$ and y not occurring free in α or β;*
12. $(\Diamond\alpha \vee \beta) \equiv {\downarrow}x.E{\downarrow}y.[(@_x\Diamond y \wedge @_y\alpha) \vee @_x\beta]$, *$x$ and y not occurring free in α or β;*
13. $(\Box\alpha \vee \beta) \equiv {\downarrow}x.A{\downarrow}y.[(@_x\Diamond y \to @_y\alpha) \vee @_x\beta]$, *$x$ and y not occurring free in α or β.*

By performing iterated applications of these equivalences, we can see that each formula in FHL can be put into the prenex form.

Lemma 3. *If $\phi \in$ FHL, then there is $\psi \in$ FHL in prenex form which is equivalent to ϕ.*

PROOF. By induction on the number of modality occurrences in ϕ and applying the equivalences in Lemma 2.

This prenex form can be strengthened with the following lemma:

Lemma 4. *${\downarrow}x.{\downarrow}y.\phi(x, y) \equiv {\downarrow}x.\phi(x, x)$ if x does not occur bound in ϕ, where $\phi(x, x)$ is obtained from $\phi(x, y)$ by substituting all free occurrences of y with x.*

PROOF. It follows straightforward by induction on ϕ.

Lemma 5. *If $\phi \in$ FHL, then ϕ is equivalent to a formula in FHL in prenex form whose prefix has no consecutive applications of binders.*

We end this section with the definition of the hierarchies of hybrid formulas induced by the prefix in the prenex form. Based on Lemma 3, we define below the classes of formulas σ^i and π^i. We will see in Theorem 9 that they are closely related with the degrees of the Polynomial Hierarchy.

Definition 12. We recursively define the classes of formulas σ^i and π^i in prenex form as:

- $\sigma^0 = \pi^0 = \{\phi \in \text{FHL} \mid \text{modalities occur in } \phi \text{ only in front of atomic formulas}\}$;
- $\sigma^{i+1} = \{\phi \in \text{FHL} \mid \phi = q_1 \dots q_n\psi, \psi \in \pi^i, q_j = \Diamond, E \text{ or } {\downarrow}x., \text{ for some } x\}$;
- $\pi^{i+1} = \{\phi \in \text{FHL} \mid \phi = q_1 \dots q_n\psi, \psi \in \sigma^i, q_j = \Box, A \text{ or } {\downarrow}x., \text{ for some } x\}$.

We say that a formula is σ^i (resp. π^i) if it is equivalent to a formula in σ^i (resp. π^i).

From Lemma 3 it follows that each formula in FHL is π^i or σ^i for some i. In the following section, we talk about the expressibility of graph properties in hybrid logic with respect to frame definability.

3. Properties of Graphs in HL

In [4], it was shown that there is a formula ϕ_n of FHL for each natural number n such that a graph G of size n is Hamiltonian iff ϕ_n is valid in G. The main question which underlies this investigation is whether there is a sequence of formulas $(\phi_n)_{n\in N}$ for each graph property $\mathcal{G}$ in NP such that a graph G of size n is in $\mathcal{G}$ iff ϕ_n is globally true in G, regarded as a frame. Actually, we can show that such sequence exists for each graph property. This follows directly from the equivalence between FHL and FO. With respect to frame definability, we can show the existence of such formulas in a very restrict fragment of FHL. Recall that a graph property is any set of graphs closed under isomorphisms.

Let $G = (V, E)$ be a graph of size n. Let us consider that the set V of states coincides with the set $\{1, \dots, n\}$ of nominals. Consider the formula:

$$\psi_G = \bigwedge_{(i,j)\in E} @_i\Diamond j \wedge \bigwedge_{(i,j)\notin E} @_i\neg\Diamond j.$$

Let $\mathcal{G}$ be any property of graphs. We define the formulas

$$\psi^n_{\mathcal{G}} = \bigvee_{G\in\mathcal{G},|G|=n} \psi_G \quad , \quad \theta^n = \bigwedge_{i,j\in\{1,\dots,n\},i\neq j} @_i\neg j \quad \text{and} \quad \phi^n_{\mathcal{G}} = \theta^n \to \psi^n_{\mathcal{G}}.$$

Lemma 6. *Let G be a graph of size n and $\mathcal{G}$ a property of graphs. Then $G \in \mathcal{G}$ iff $G \Vdash \phi^n_{\mathcal{G}}$.*

Proof. Let $G = (V, E)$ be a graph. Then $G \Vdash \phi^n_{\mathcal{G}}$ iff, for each valuation function $\mathbf{V}$ of the nominals we have, $(G, \mathbf{V}) \Vdash \phi^n_{\mathcal{G}}$. Let $\mathbf{V}$ be a valuation function. Suppose $(G, \mathbf{V}) \Vdash \phi^n_{\mathcal{G}}$. If $(G, \mathbf{V}) \Vdash \theta^n$, then $\mathbf{V}$ assigns to each nominal a different element (that is, the restriction of $\mathbf{V}$ to the set $\{1, \ldots, n\}$ of nominals is injective). In this case, $(G, \mathbf{V}) \Vdash \psi_{G'}$ for some $G' \in \mathcal{G}$. It follows that $(G, \mathbf{V}) \Vdash @_i \Diamond j$ iff $(\mathbf{V}(i), \mathbf{V}(j)) \in E$ iff $(i, j) \in E'$. Hence, the restriction of $\mathbf{V}$ to $\{1, \ldots, n\}$ is an isomorphism between G and G'. Then $G \in \mathcal{G}$.

Now, suppose $G \in \mathcal{G}$. Let $\mathbf{V}$ be a valuation function which is injective in the set $\{1, \ldots, n\}$ of nominals. It follows that $(G, \mathbf{V}) \Vdash \theta^n$ and $(G, \mathbf{V}) \Vdash @_i \Diamond j$ iff $(\mathbf{V}^{-1}(i), \mathbf{V}^{-1}(j)) \in E$. $\mathbf{V}$ induces a graph $G' = (V', E')$ isomorphic to G where $V' = V$ and $(i, j) \in E'$ iff $(\mathbf{V}^{-1}(i), \mathbf{V}^{-1}(j)) \in E'$. Hence, $(i, j) \in E'$ iff $(G, \mathbf{V}) \Vdash @_i \Diamond j$. It follows that $(G, \mathbf{V}) \Vdash \psi_{G'}$. Then $(G, \mathbf{V}) \Vdash \phi^n_{\mathcal{G}}$, for each valuation function $\mathbf{V}$. It follows that $G \Vdash \phi^n_{\mathcal{G}}$.

Since there are 2^{n^2} graphs with states in $\{1, \ldots, n\}$, we have that the size of $\phi^n_{\mathcal{G}}$ is $O(2^{n^2})$ for any graph property $\mathcal{G}$. Obviously, there is no hope that such sequence of formulas will always be recursive. We can show, however, that for properties in the polynomial hierarchy such sequence is recursive and, moreover, there is a polynomial bound in the size of formulas.

4. Translation

In this section, we show that for each graph property $\mathcal{G}$ in the polynomial hierarchy there is a sequence $(\phi_n)_{n \in N}$ of formulas such that a graph G is in $\mathcal{G}$ iff $G \Vdash \phi_{|G|}$ and such that ϕ_n is bounded from above by a polynomial in n. We will use the well-known characterization of problems in PH and classes of finite models definable in second-order logic from descriptive complexity theory. To this end, we define a translation from formulas in SO to formulas in FHL which are equivalent with respect to frames of size n, for some $n \in N$. Such translation will give us formulas whose size is bounded by a polynomial in n. Moreover, the formulas obtained by the translation have no occurrence of propositional symbols, nominals or free state variables, which means that, for these formulas, the complexity of model-checking and frame-checking coincides. We use the well known definitions and concepts related to first- and second-order logic which can be found in most textbooks (see, for instance, [10]).

Definition 13 (Translation from FO to FHL). Let ϕ be a first-order formula in the symbol set $S = \{E, R_1, \ldots, R_m\}$, E binary, n a natural number and f a function from the set of first-order variables into $\{1, \ldots, n\}$. Let $t, z_1, \ldots, z_n$ be state variables and for each $R \in \{R_1, \ldots, R_m\}$ of arity h, let $y^R_{j_1,\ldots,j_h}$ be a state variable, with $j_i \in \{1, \ldots, n\}$, $1 \leq i \leq h$. We define the function $tr^f_n : L^S_{FO} \to$ FHL as:

- $tr^f_n(x_1 \equiv x_2) = @_{z_{f(x_1)}} z_{f(x_2)}$;
- $tr^f_n(E(x_1, x_2)) = @_{z_{f(x_1)}} \Diamond z_{f(x_2)}$;
- $tr^f_n(R(x_1, \ldots, x_k)) = @_t y^R_{f(x_1),\ldots,f(x_k)}$, for each $R \in \{R_1, \ldots, R_m\}$;
- $tr^f_n(\gamma \wedge \theta) = tr^f_n(\gamma) \wedge tr^f_n(\theta)$;
- $tr^f_n(\neg\gamma) = \neg tr^f_n(\gamma)$;
- $tr^f_n(\exists x\gamma) = \bigvee^n_{i=1} tr^{f\frac{x}{i}}_n(\gamma)$;
- $tr^f_n(\forall x\gamma) = \bigwedge^n_{i=1} tr^{f\frac{x}{i}}_n(\gamma)$.

In the translation above, t is intended to represent a state v such that, if $z^R_{j_1,\ldots,j_h}$ is assigned to t and $z_{j_1}, \ldots, z_{j_h}$ are assigned to $v_1, \ldots, v_h$, then $(v_1, \ldots, v_h)$ belongs to the interpretation of R. The function $f\frac{x}{i}$ maps x to i and y to $f(y)$ for $y \neq x$. The translation above only works for frames with more than one state but, since there are only two frames of size 1, we can state for each graph property which frames of size 1 belong to the property.

Note that if ϕ is a sentence, then $tr^f_n(\phi) = tr^{f'}_n(\phi)$. Hence we write $tr_n(\phi)$ instead of $tr^f_n(\phi)$ for a sentence ϕ.

Example 1. We give an example of application of the translation above. Let $\veebar^3(\psi_1, \psi_2, \psi_3)$ be the ternary connective "exactly one", which is true iff exactly one among ψ_1, ψ_2 and ψ_3 is true. Consider the following first-order sentence:

$$\begin{aligned} \phi \quad = \quad & \forall x(\veebar^3 (R(x), G(x), B(x))) \wedge \forall x \forall y((E(x, y) \wedge x \neq y) \to \\ & \neg((R(x) \wedge R(y)) \vee (G(x) \wedge G(y)) \vee (B(x) \wedge B(y)))). \end{aligned}$$

The sentence above says that each element belongs to one of the sets R, G and B, each adjacent pair does not belong to the same set, and no element belongs to more than one set. This sentence is true iff the sets R, G and B

forms a 3-colouring of a graph with edges in E. Below we translate ϕ into a formula of hybrid logic using the translation given above and setting $n = 3$:

$$tr_3(\phi) \;=\; \bigwedge_{i=1}^{3} \underline{\vee}^3(@_t y_i^R, @_t y_i^G, @_t y_i^B) \wedge \bigwedge_{i=1}^{3} \Bigg[\bigwedge_{j=1}^{3} ((@_{z_i} \Diamond z_j \wedge \neg @_{z_i} z_j) \rightarrow$$
$$\neg((@_t y_i^R \wedge @_t y_j^R) \vee (@_t y_i^G \wedge @_t y_j^G) \vee (@_t y_i^B \wedge @_t y_j^B))) \Bigg].$$

Lemma 7. *$tr_n(\phi)$ has polynomial size in n for each fixed formula ϕ, that is, $tr_n(\phi) \in O(n^k)$ for some $0 \leq k$.*

Proof. By induction on ϕ one can see that $tr_n(\phi)$ is $O(n^k)$, where k is the quantifier rank of ϕ.

Lemma 8. *Let $G = (V, E^G)$ be a graph of size n, $\mathbf{R}_1, \ldots, \mathbf{R}_m$ relations on V with arities $r_1, \ldots, r_m$, g an assignment of state variables, β an assignment of first-order variables and f a function from the set of first-order variables to $\{1, \ldots, n\}$ such that:*

(i) g assigns to each variable z_i, $1 \leq i \leq n$, a different element in V;
(ii) $g(y^R_{i_1,\ldots,i_k}) = g(t)$ iff $(g(z_{i_1}), \ldots, g(z_{i_k})) \in \mathbf{R}$ for each $R \in \{R_1, \ldots, R_m\}$;
(iii) $\beta(x) = g(z_{f(x)})$ for each first-order variable x.

If ϕ is a first-order formula in the symbol set $\{E, R_1, \ldots, R_m\}$, then

$$(G, \mathbf{R}_1, \ldots, \mathbf{R}_m, \beta) \models \phi \text{ iff for all } w \in V,\ (G, g, w) \Vdash tr_n^f(\phi).$$

Proof. We proceed by induction on ϕ.

— ϕ is atomic: In this case $\phi = x \equiv y$, $\phi = E(x, y)$ or $\phi = R_i(x_1, \ldots, x_n)$. If $\phi = x \equiv y$, then $(G, \mathbf{R}_1, \ldots, \mathbf{R}_m, \beta) \models \phi$ iff $\beta(x) = \beta(y)$ iff, by (iii), $g(z_{f(x)}) = g(z_{f(y)})$ iff $(G, g, w) \Vdash @_{z_{f(x)}} z_{f(y)} = tr_n^f(\phi)$. If $\phi = E(x, y)$, then $(G, \mathbf{R}_1, \ldots, \mathbf{R}_m, \beta) \models \phi$ iff $(\beta(x), \beta(y)) \in E^G$ iff, by (iii), $(g(z_{f(x)}), g(z_{f(y)}) \in E^G$ iff $(G, g, w) \Vdash @_{z_{f(x_1)}} \Diamond z_{f(x_2)} = tr_n^f(\phi)$. If $\phi = R_i(x_1, \ldots, x_h)$, then $(G, \mathbf{R}_1, \ldots, \mathbf{R}_m, \beta) \models \phi$ iff $(\beta(x_1), \ldots, \beta(x_n)) \in \mathbf{R}_i$ iff, by (iii), $(g(z_{f(x_1)}), \ldots, g(z_{f(x_n)})) \in \mathbf{R}_i$ iff, by (ii), $g(y^R_{f(x_1),\ldots,f(x_k)}) = g(t)$ iff $(G, g, w) \Vdash @_t y^R_{f(x_1),\ldots,f(x_k)} = tr_n^f(\phi)$.

— $\phi = \gamma \wedge \theta$ or $\phi = \neg\gamma$: These cases follow directly from the definition of tr_n^f and the inductive hypothesis.

— $\phi = \exists x\gamma$: In this case, $(G, \mathbf{R}_1, \ldots, \mathbf{R}_m, \beta) \models \phi$ iff there is a $v \in V$ such that $(G, \mathbf{R}_1, \ldots, \mathbf{R}_m, \beta\frac{x}{v}) \models \gamma$. By (i), there is a j be such that $v = z_j$. Hence we have $\beta\frac{x}{v}(y) = g(z_{f\frac{x}{j}(y)})$ for each first-order variable y. By inductive hypothesis we have, $(G, \mathbf{R}_1, \ldots, \mathbf{R}_m, \beta\frac{x}{v}) \models \gamma$ iff $(G, g, w) \Vdash tr_n^{f\frac{x}{j}}(\gamma)$ iff $(G, g, w) \Vdash \bigvee_{i=1}^n tr_n^{f\frac{x}{i}}(\gamma) = tr_n^f(\phi)$.

— $\phi = \forall x\gamma$: In this case, $(G, \mathbf{R}_1, \ldots, \mathbf{R}_m, \beta) \models \phi$ iff, for each $v \in V$, $(G, \mathbf{R}_1, \ldots, \mathbf{R}_m, \beta\frac{x}{v}) \models \gamma$. By (i), for each $v \in V$ there is a j such that $v = z_j$. Hence we have $\beta\frac{x}{v}(y) = g(z_{f\frac{x}{j}(y)})$ for each first-order variable y. By inductive hypothesis we have, for each $v \in V$, $(G, \mathbf{R}_1, \ldots, \mathbf{R}_m, \beta\frac{x}{v}) \models \gamma$ iff, for each $j \in \{1, \ldots, n\}$, $(G, g, w) \Vdash tr_n^{f\frac{x}{j}}(\gamma)$ iff $(G, g, w) \Vdash \bigwedge_{i=1}^n tr_n^{f\frac{x}{i}}(\gamma) = tr_n^f(\phi)$.

We introduce below the translation from SO to FHL we will use to construct sequences of FHL formulas that express graph problems in PH. Without loss of generality, we can suppose that SO formulas are in prenex normal form, since every SO formula is equivalent to one in prenex normal form.

Definition 14 (Translation from SO to FHL). Let X_i be a relation variable of arity r_i. Let $y_{\overline{a}}^{X_i}$, where $\overline{a} \in \{1, \ldots, n\}^{r_i}$ be a state variable (in particular we set $\overline{i} = i, \ldots, i$ for each $i \in \{1, \ldots, n\}$). Let $\phi = Q_1 X_1 \ldots Q_l X_l \psi$ be a second-order formula, $Q_i \in \{\forall, \exists\}$, where ψ is a first-order sentence. We define

$$T^n(\phi) = \spadesuit_1{\downarrow}y_{\overline{1}}^{X_1}.\ldots.\spadesuit_1{\downarrow}y_{\overline{n}}^{X_1}.\ldots.\spadesuit_l{\downarrow}y_{\overline{1}}^{X_l}.\ldots.\spadesuit_l{\downarrow}y_{\overline{n}}^{X_l} tr_n(\psi),$$

where $\spadesuit_i = E$ if $Q_i = \exists$ and A otherwise, $1 \leq i \leq l$.

Example 2. Consider the sentence ϕ of Example 1. Let ψ be the following second-order sentence:

$$\psi = \exists R \exists G \exists B(\phi).$$

The sentence ψ above states that there are three sets R, G and B which form a 3-colouring of elements in the domain of a structure. Hence, ψ is satisfied in a graph with edges in E iff this graph is 3-colourable. Deciding whether a graph is 3-colourable is an NP-complete problem [14]. We apply the translation T^n for $n = 3$ below. Let

$$\hat{\mathbf{Q}} = E{\downarrow}y_1^R.E{\downarrow}y_2^R.E{\downarrow}y_3^R.E{\downarrow}y_1^G.E{\downarrow}y_2^G.E{\downarrow}y_3^G.E{\downarrow}y_1^B.E{\downarrow}y_2^B.E{\downarrow}y_3^B..$$

We have $T^3(\psi) = \hat{\mathbf{Q}} tr_3(\phi)$. That is,

$$T^3(\psi) \;=\; \hat{\mathbf{Q}}\Big(\bigwedge_{i=1}^{3} \underline{\vee}^3(@_t y_i^R, @_t y_i^G, @_t y_i^B) \wedge \bigwedge_{i=1}^{3} \Big[\bigwedge_{j=1}^{3} ((@_{z_i} \Diamond z_j \wedge \neg @_{z_i} z_j) \rightarrow$$
$$\neg((@_t y_i^R \wedge @_t y_j^R) \vee (@_t y_i^G \wedge @_t y_j^G) \vee (@_t y_i^B \wedge @_t y_j^B))) \Big] \Big).$$

Lemma 9. *Let $G = (V, E^G)$ be a graph of size $n \geq 2$, $\mathbf{R}_1, \ldots, \mathbf{R}_m$ relations on V with arities $r_1, \ldots, r_m$, g an assignment of state variables, β an assignment of first-order variables and f a function from the set of first-order variables to $\{1, \ldots, n\}$ such that:*

(i) g assigns to each variable z_i a different element in V;
(ii) $g(y^R_{i_1,\ldots,i_k}) = g(t)$ iff $(g(z_{i_1}), \ldots, g(z_{i_k})) \in \mathbf{R}$ for each $R \in \{R_1, \ldots, R_m\}$;
(iii) $\beta(x) = g(z_{f(x)})$ for each first-order variable x.

If $\phi = Q_1 X_1 \ldots Q_l X_l \psi$ is a second-order formula in the symbol set

$$\{E, R_1, \ldots, R_m\},$$

then $(G, \mathbf{R}_1, \ldots, \mathbf{R}_m, \beta) \models \phi$ iff for all $w \in V$, $(G, g, w) \models T^n(\phi)$.

Proof. Let $v \in V$ such that $v \neq g(t)$. For each $\mathbf{X}_1$ on V, let $v^{\mathbf{X}_1}_{i_1,\ldots,i_h} = g(t)$ if $(g(z_{i_1}), \ldots, g(z_{i_h})) \in \mathbf{X}_1$ and v otherwise. Then, for each $\mathbf{X}_1$ on V there is an assignment $g^{\mathbf{X}_1}$ defined as

$$g^{\mathbf{X}_1} = g\frac{y^{X_1}_{1,\ldots,1} \ldots y^{X_1}_{i_1,\ldots,i_h} \ldots y^{X_1}_{n,\ldots,n}}{v^{\mathbf{X}_1}_{1,\ldots,1} \ldots v^{\mathbf{X}_1}_{i_1,\ldots,i_h} \ldots v^{\mathbf{X}_1}_{n,\ldots,n}}.$$

Conversely, given an assignment g' we can find $\mathbf{X}_1$ such that $g' = g^{\mathbf{X}_1}$.

The assignment $g^{\mathbf{X}_1}$ can be described as:

$$g^{\mathbf{X}_1}(s) = \begin{cases} g(t) & \text{, if } s = y^{X_1}_{i_1,\ldots,i_h} \text{ and } (i_1, \ldots, i_h) \in \mathbf{X}_1; \\ v & \text{, for some } v \neq g(t) \text{, if } s = y^{X_1}_{i_1,\ldots,i_h} \text{ and } (i_1, \ldots, i_h) \notin \mathbf{X}_1; \\ g(s) & \text{, otherwise;} \end{cases}$$

It follows that $g^{\mathbf{X}_1}$ and $\mathbf{X}_1$ satisfies (i) and (iii) and

(ii') $g^{\mathbf{X}_1}(y^R_{i_1,\ldots,i_k}) = g^{\mathbf{X}_1}(t)$ iff $(g^{\mathbf{X}_1}(z_{i_1}), \ldots, g^{\mathbf{X}_1}(z_{i_k})) \in \mathbf{R}$ for each

$$R \in \{R_1, \ldots, R_m, X_1\}.$$

Now, we proceed by induction on the size l of the prefix $Q_1X_1 \dots Q_lX_l$. If $l = 0$, then ϕ is first-order and the result follows immediately from Lemma 8. Suppose that $l > 0$. If $\phi = \exists X_1 \dots Q_lX_l\psi$, then

$$(G, \mathbf{R}_1, \dots, \mathbf{R}_m, \beta) \models \phi$$

iff there is $\mathbf{X_1} \subseteq V^{r_1}$ such that

$$(G, \mathbf{R}_1, \dots, \mathbf{R}_m, \mathbf{X}_1, \beta) \models Q_2X_2 \dots Q_lX_l\psi$$

iff, by the inductive hypothesis, there is $g^{\mathbf{X}_1}$ such that

$$(G, g^{\mathbf{X}_1}, w) \models T^n(Q_2X_2 \dots Q_lX_l\psi)$$

iff

$$(G, g\frac{y^{X_1}_{1,\dots,1} \dots y^{X_1}_{i_1,\dots,i_h} \dots y^{X_1}_{n,\dots,n}}{v^{\mathbf{X}_1}_{1,\dots,1} \dots v^{\mathbf{X}_1}_{i_1,\dots,i_h} \dots v^{\mathbf{X}_1}_{n,\dots,n}}, w) \models T^n(Q_2X_2 \dots Q_lX_l\psi)$$

iff

$$(G, g, w) \models E{\downarrow}y^{X_1}_{\overline{1}} \dots . E{\downarrow}y^{X_1}_{\overline{n}}.T^n(Q_2X_2 \dots Q_lX_l\psi) = T^n(\phi).$$

If $\phi = \forall X_1 \dots Q_lX_l\psi$, then

$$(G, \mathbf{R}_1, \dots, \mathbf{R}_m, \beta) \models \phi$$

iff, for all $\mathbf{X_1} \subseteq V^{r_1}$,

$$(G, \mathbf{R}_1, \dots, \mathbf{R}_m, \mathbf{X}_1, \beta) \models Q_2X_2 \dots Q_lX_l\psi$$

iff, by the inductive hypothesis, for all $g^{\mathbf{X}_1}$,

$$(G, g^{\mathbf{X}_1}, w) \models T^n(Q_2X_2 \dots Q_lX_l\psi)$$

iff, for all $v^{\mathbf{X}_1}_{1,\dots,1} \dots v^{\mathbf{X}_1}_{i_1,\dots,i_h} \dots v^{\mathbf{X}_1}_{n,\dots,n}$,

$$(G, g\frac{y^{X_1}_{1,\dots,1} \dots y^{X_1}_{i_1,\dots,i_h} \dots y^{X_1}_{n,\dots,n}}{v^{\mathbf{X}_1}_{1,\dots,1} \dots v^{\mathbf{X}_1}_{i_1,\dots,i_h} \dots v^{\mathbf{X}_1}_{n,\dots,n}}, w) \models T^n(Q_2X_2 \dots Q_lX_l\psi)$$

iff

$$(G, g, w) \models A{\downarrow}y^{X_1}_{\overline{1}} \dots . A{\downarrow}y^{X_1}_{\overline{n}}.T^n(Q_2X_2 \dots Q_lX_l\psi) = T^n(\phi).$$

We have the following:

Theorem 3. *Let ϕ be a second-order sentence on the symbol set $S = \{E\}$, and G a graph of size $n \geq 2$. Let*

$$\psi_n = \downarrow t.E\downarrow z_1.\ldots E\downarrow z_n.\left(\bigwedge_{1\leq i<j\leq n} @_{z_i}\neg z_j \wedge T^n(\phi)\right).$$

Then $G \models \phi$ iff $G \models \psi_n$.

Proof.

$$(G, g, w) \models \downarrow t.E\downarrow z_1.\ldots E\downarrow z_n.\left[\left(\bigwedge_{1\leq i<j\leq n} @_{z_i}\neg z_j\right) \wedge T^n(\phi)\right]$$

iff

$$(G, g\frac{t}{w}, w) \models E\downarrow z_1.\ldots E\downarrow z_n.\left[\left(\bigwedge_{1\leq i<j\leq n} @_{z_i}\neg z_j\right) \wedge T^n(\phi)\right]$$

iff there are $v_1, \ldots, v_n \in V$ such that

$$(G, g\frac{tz_1 \ldots z_n}{wv_1 \ldots v_n}, w) \models \left[\left(\bigwedge_{1\leq i<j\leq n} @_{z_i}\neg z_j\right) \wedge T^n(\phi)\right]$$

iff there are $v_1 \neq \ldots \neq v_n \in V$ such that

$$(G, g\frac{tz_1 \ldots z_n}{wv_1 \ldots v_n}, w) \models T^n(\phi)$$

iff, by Lemma 9,

$$(G, \beta, w) \models \phi$$

for β as in Lemma 9 and all w in V. Since ϕ has no free variables, we have that

$$(G, \beta, w) \models \phi \text{ iff } G \models \phi.$$

The following is the main theorem of this section:

Theorem 4. *Let $\mathcal{G}$ be a graph property in PH. Then there is a sequence of sentences $\Phi = \{\phi_1, \phi_2, \ldots\}$ of* FHL$\backslash\{$NOM, PROP$\}$*, such that:*
(1) $G \in \mathcal{G}$ iff $G \models \Phi$ iff $G \models \phi_{|G|}$, and
(2) $|\phi_n|$ is $O(n^k)$ for some constant k depending only on $\mathcal{G}$.

PROOF. Without loss of generality, we may suppose that all graphs in $\mathcal{G}$ have at least two vertices, since there are finitely many (actually two) graphs with only one vertex and they can be defined up to isomorphism by an FHL sentence. Let ψ be a second-order formula expressing $\mathcal{G}$. This formula exists by Theorem 2. Let

$$\begin{aligned}\theta_n \quad = \quad & E{\downarrow}z_1.\dots E{\downarrow}z_n.\left[\left(\bigwedge_{1\leq i<j\leq n} @_{z_i}\neg z_j\right)\right] \wedge \\ & A{\downarrow}z_1.\dots A{\downarrow}z_n.\left[\bigwedge_{1\leq i<j\leq n} @_{z_i}\neg z_j \rightarrow A{\downarrow}z.\left(\bigvee_{1\leq i\leq n} @_{z_i} z\right)\right].\end{aligned}$$

The sentence θ_n says that there are exactly n states in the frame. Let ψ_n be the formula from Theorem 3. We define ϕ_n as:

$$\phi_n = \theta_n \rightarrow \psi_n.$$

Let $G \in \mathcal{G}$. Let g be any assignment of state variables and w be any point in G. If $G \not\models \theta_n$, then $G \models \phi_n$. It follows that $G \models \phi_n$ for each $n \neq |G|$. Hence, $G \models \Phi$ iff $G \models \phi_{|G|}$. Let $|G| = n$. Then $(G, g, w) \models \phi_n$ iff

$$(G, g, w) \models {\downarrow}t.E{\downarrow}z_1.\dots E{\downarrow}z_n.\left[\left(\bigwedge_{1\leq i<j\leq n} @_{z_i}\neg z_j\right) \wedge T^n(\psi)\right]$$

iff, by Theorem 3, $G \in \mathcal{G}$.

In the following section, we will show that we can discharge the global modalities if we consider connected frames with loops.

5. Connected Frames with Loops

In this section, we will show that some results presented above still hold if we consider certain fragments of FHL provided that we restrict ourselves to connected frames with loops, that is, frames such that, for every two states v and w, there is a path from v to w or a path from w to v and for every state v there is a loop, that is, an edge from v to itself.

Let $\text{FHL}\backslash\{E, \text{NOM}, \text{PROP}\}$ be the hybrid logic without the modality E and without nominals. One can see that an analogous to Theorem 4 does not hold for $\text{FHL}\backslash\{E, \text{NOM}, \text{PROP}\}$. Let us define the disjoint union of frames $G = (V, E)$ and $G' = (V', E')$ such that $V \cap V' = \emptyset$ as the frame $G'' = (V \cup V', E \cup E')$. Similarly, for models $\mathcal{M} = (G, \mathbf{V})$ and $\mathcal{M}' = (G', \mathbf{V}')$ the disjoint union is defined as $\mathcal{M}'' = (G'', \mathbf{V} \cup \mathbf{V}')$, where $(\mathbf{V} \cup \mathbf{V}')(p) = \mathbf{V}(p) \cup \mathbf{V}'(p)$.

Its is not difficult to show that:

Lemma 10. *Frame validity and global truth for sentences in the fragment* FHL$\backslash\{E, \mathrm{NOM}, \mathrm{PROP}\}$ *are invariant under disjoint union.*

Proof. Let $G = (V, E)$ and $G' = (V', E')$ be two frames such that $V \cap V'$, and let G'' be the disjoint union of G and G'. Let $\mathcal{M} = (G, \mathbf{V})$ and $\mathcal{M}' = (G', \mathbf{V}')$ and $\mathcal{M}''$ the corresponding disjoint union. Let ϕ be a formula of FHL$\backslash\{E, \mathrm{NOM}, \mathrm{PROP}\}$. It follows easily by induction on ϕ that,

- $\mathcal{M} \Vdash \phi$ iff $\mathcal{M}', g, v \Vdash \phi$ for all $v \in V$ and all assignment g which maps free state variables occurring in ϕ in elements of V, and
- $\mathcal{M}' \Vdash \phi$ iff $\mathcal{M}', g, v' \Vdash \phi$ for all $v' \in V'$ and all assignment g which maps free state variables in elements of V'.

Hence, if ϕ is a sentence, it follows that $\mathcal{M} \Vdash \phi$ and $\mathcal{M}' \Vdash \phi$ iff $\mathcal{M}'' \Vdash \phi$.

It follows from Lemma 10 above that an analogue of Theorem 4 does not hold for FHL$\backslash\{E, \mathrm{NOM}, \mathrm{PROP}\}$.

Corollary 1. *There are graph properties in P for which there is no set $\Phi = \{\phi_1, \phi_2, \ldots\}$ of sentences in* FHL$\backslash\{E, \mathrm{NOM}, \mathrm{PROP}\}$ *which satisfies condition (1) from Theorem 4 above.*

Proof. Connectivity is one such a property. Suppose there is such set. Let G and G' be connected frames of size n, then G" be the disjoint union of G and G'. Then $G'' \nVdash \phi_{2n}$, hence $G'' \Vdash \neg\phi_{2n}$ since ϕ_{2n} has no propositional symbol, and by Lemma 10 we have that $G \Vdash \neg\phi_{2n}$ or $G' \Vdash \neg\phi_{2n}$, which contradicts condition (1).

However, Theorem 4 still holds if we restrict ourselves to connected frames with loops. Consider the following translation from formulas in SO into formulas in FHL$\backslash\{E, \mathrm{NOM}, \mathrm{PROP}\}$:

Definition 15. Let $\phi = Q_1X_1 \ldots Q_lX_l\psi$ be a second-order formula where ψ is a first-order sentence. We define

$$\hat{T}^n(\phi) = \spadesuit_1{\downarrow}y_{\overline{1}}^{X_1}.\ldots.\spadesuit_1{\downarrow}y_{\overline{n}}^{X_1}.\ldots.\spadesuit_l{\downarrow}y_{\overline{1}}^{X_l}.\ldots.\spadesuit_l{\downarrow}y_{\overline{n}}^{X_l}tr_n(\psi),$$

where $\spadesuit_i = (\Diamond\Diamond^{-1})^n = \underbrace{\Diamond\Diamond^{-1}\ldots\Diamond\Diamond^{-1}}_{n}$ if $Q_i = E$ and $(\Box\Box^{-1})^n$ otherwise.

Lemma 11. *Let $G = (V, E^G)$ be a connected graph of size $n \geq 2$ with loops on each state, $\mathbf{R}_1, \ldots, \mathbf{R}_m$ relations on V with arities $r_1, \ldots, r_m$, g an assignment of state variables, β an assignment of first-order variables and f a function from the set of first-order variables to $\{1, \ldots, n\}$ such that:*

(i) g assigns to each variable z_i a different element in V;
(ii) $g(y^R_{i_1,\ldots,i_k}) = g(t)$ iff $(g(z_{i_1}), \ldots, g(z_{i_k})) \in \mathbf{R}$ for each $R \in \{R_1, \ldots, R_m\}$;
(iii) $\beta(x) = g(z_{f(x)})$ for each first-order variable x;

If $\phi = Q_1X_1 \ldots Q_lX_l\psi$ is a second-order formula in the symbol set

$$\{E, R_1, \ldots, R_m\},$$

then

$$(G, \mathbf{R}_1, \ldots, \mathbf{R}_m, \beta) \models \phi \text{ iff for all } w \in V,\ (G, g, w) \models \hat{T}^n(\phi).$$

Proof. Analogous to the proof of Lemma 9

Theorem 5. *Let ϕ be a second-order sentence and G a connected graph of size $n \geq 2$ with loops. Let*

$$\psi_n = {\downarrow}t.(\Diamond\Diamond^{-1})^n{\downarrow}z_1.\ldots.(\Diamond\Diamond^{-1})^n{\downarrow}z_n.\left(\bigwedge_{1\leq i<j\leq n} @_{z_i}\neg z_j \wedge \hat{T}^n(\phi)\right).$$

Then $G \models \phi$ iff $G \models \psi_n$.

Proof. Analogous to the proof of Theorem 3, using Lemma 11 instead of Lemma 9.

Theorem 6. *Let $\mathcal{G}$ be a property of connected graphs with loops in the polynomial hierarchy. Then there is a set of sentences*

$$\Phi = \{\phi_1, \phi_2, \ldots\}$$

of $\mathrm{FHL}\backslash\{E, \mathrm{NOM}, \mathrm{PROP}\}$*, such that:*

(1)for all connected G with loops, $G \in \mathcal{G}$ iff $G \models \Phi$ iff $G \models \phi_{|G|}$, and

(2) ϕ_m is $O(n^k)$ for some constant k depending only on $\mathcal{G}$.

Proof. Similar to Theorem 4, we may suppose that each graph in $\mathcal{G}$ has at least two vertices. Let ψ be a second-order formula expressing $\mathcal{G}$. Let

$$\begin{aligned}\theta_n \quad = \quad & (\Box\Box^{-1})^n{\downarrow}z_1.\ldots(\Box\Box^{-1})^n{\downarrow}z_n.\,[\\ & \bigwedge_{1\le i<j\le n} @_{z_i}\neg z_j \to (\Box\Box^{-1})^n{\downarrow}z.\left(\bigvee_{1\le i\le n} @_{z_i}z\right)\Bigg].\end{aligned}$$

Let ψ_n be the formula from Lemma 11. We define ϕ_n as:

$$\phi_n \quad = \quad \theta_n \to \psi_n.$$

The remainder of the proof is similar to the proof of Theorem 4.

In the following section, we will show that the prefix pattern of the prenex form is closely related with the degrees of the polynomial hierarchy.

6. Polynomial Hierarchy and Fragments of HL

In [7], it is proved that the model-checking problem for the $\mathrm{FHL}\backslash{\downarrow}\Box{\downarrow}$ fragment is NP-complete. The translation given in Section 4 above can be used to produce hybrid formulas of polynomial size using formulas of second-order logic. This leads to a alternative proof that the model-checking problem for the fragment $\mathrm{FHL}\backslash{\downarrow}\Box{\downarrow}$ is hard for NP, since there is a polynomial reduction for any instance of an NP problem to the model-checking problem of $\mathrm{FHL}\backslash{\downarrow}\Box{\downarrow}$. To show this, we need the following lemma:

Lemma 12. *If $\phi \in \exists SO$, the existential fragment of second-order logic, then T^n is in* $\mathrm{HL}(@, \downarrow, E)\backslash\{{\downarrow}\Box{\downarrow}, \mathrm{NOM}, \mathrm{PROP}\}$*, that is, the fragment of* $\mathrm{HL}(@, \downarrow, E)$ *without, nominals, propositional symbols and the patterns ${\downarrow}\Box{\downarrow}$ and ${\downarrow}A{\downarrow}$, that is, an $\downarrow$ inside the scope of a universal modality A or $\Box$, which in turn is inside the scope of other $\downarrow$.*

Theorem 7 ([7]). *The model-checking problem for $\sigma^1 \subseteq$ FHL$\backslash\downarrow\Box\downarrow$ is NP-hard.*

Proof. Let $\mathcal{G}$ be an NP-complete graph property. Let ϕ be an $\exists$SO sentence which express $\mathcal{G}$. By Fagin's Theorem [8], such sentence exists. Let G be a graph. Let $T^{|G|}(\phi)$ be as defined in Definition 14. It is easy to see that $T^{|G|}(\phi)$ can be constructed from ϕ in time polynomial in $|G|$. Now, $(G, T^{|G|}(\phi))$ is an instance of the model-checking problem for FHL$\backslash\downarrow\Box\downarrow$. By Theorem 4, the model-checker returns *true* for $(G, T^{|G|}(\phi))$ iff $G \in \mathcal{G}$. Hence, the model-checking problem for FHL$\backslash\downarrow\Box\downarrow$ is hard for NP.

Actually, for each degree of the polynomial hierarchy, there is a syntactically defined fragment of HL whose model-checking problem is hard.

Theorem 8. *The model-checking problem for σ^i (resp. π^i) is Σ_i^p-hard (resp. Π_i^p-hard).*

Proof. Analogous to the proof of Theorem 7 above, since each graph property in Σ_i^p (resp. Π_i^p) can be expressed by a SO sentence in Σ_i^1 (resp. Π_i^1), and $T^n(\phi)$ can always be constructed in time polynomial in n, for a fixed $\phi \in$ SO.

Also, the model-checking problem for σ^i and π^i are in Σ_i^p and Π_i^p, respectively.

Theorem 9. *The model-checking problem for σ^i (resp. π^i) is in Σ_i^p (resp. Π_i^p).*

Proof. We proceed by induction on i. For the sake of simplicity, we consider only the modalities $\Diamond$ and $\Box$ in the prefix, but the proof is analogous for E and A. In [7], it is shown that the model-checking problem for FHL$\backslash\downarrow\Box\downarrow$, which contains σ^1, is in NP. It follows that π^1 is in co-NP. Now, let $\overline{q}\phi$ be a sentence in σ^{i+1}, where ϕ is in π^i. It follows that $\overline{q}$ has the form

$$\overline{q} = \Diamond^{k_1}\downarrow x_1.\Diamond^{k_2}\downarrow x_2.\ldots\Diamond^{k_m}\downarrow x_m.\Diamond^{k_{m+1}},$$

Where $\Diamond^{k_i} = \underbrace{\Diamond\ldots\Diamond}_{k_i}$, for a natural number k_i. Let M be a finite model, g be an assignment of state variables and w a point in W. By inductive hypothesis, suppose that the model-checking problem for π^i is in Π_i^p. We can use nondeterministic Turing machine to existentially guess values v_j for x_j among the points in W which are reachable in $\sum_{i=1}^{j} k_i$ steps from w, with respect to the accessibility relation R, in polynomial nondeterministic time, and we can existentially guess states w' reachable in $\sum_{i=1}^{m+1} k_i$ steps from w in polynomial nondeterministic time also. Finally, we can use an oracle for the model-checking problem for π^i with the input

$$(M, g\frac{v_1 \dots v_m}{x_1 \dots x_m}, w'), \phi).$$

By inductive hypothesis, such an oracle is in Π_i^p. As the existential guesses initially performed can be made in (existential) nondeterministic polynomial time, the model-checking problem for σ^{i+1} is in Σ_{i+1}^p.

The proof is analogous for the model-checking problem of π^{i+1}.

Corollary 2. *Let $\Phi = \{\phi_1, \phi_2, \dots\}$ be such that each ϕ_i can be constructed in time polynomial in i and each ϕ_i is in π^j (resp. σ^j). Then the graph property $\mathcal{G}$ defined as:*

$$G \in \mathcal{G} \textit{ iff } G \models \phi_{|G|}$$

is in Π_j^p (resp. Σ_j^p).

From Theorems 8 and 9 we have:

Corollary 3. *The model-checking problem for σ^i (resp. π^i) is Σ_i^p-complete (resp. Π_i^p-complete).*

Corollary 4. *The frame-checking problem for $\sigma^i \backslash \mathrm{PROP}$ (resp. $\pi^i \backslash \mathrm{PROP}$) is Σ_i^p-complete (resp. Π_i^p-complete).*

7. Conclusions

In this article we studied the expressibility of properties in the polynomial hierarchy by sequences of formulas of hybrid logic. In [4], Benevides and Schechter show a sequence ϕ_1, ϕ_2, ... of formulas of FHL that expresses the property that a graph is Hamiltonian, in the sense that a graph of size n satisfies ϕ_n iff it is Hamiltonian. We show that this can be done for every property in NP, actually any property in the polynomial hierarchy PH (Theorem 4). Beside this, the size of formulas is bounded by a polynomial in the size of the graph. This leads to an alternative proof of the NP-hardness

(Theorem 7), different from the one presented in [7]. If we do not use the global modalities E and A, we can no more express properties in PH, by sequences of formulas, actually, relatively simple properties like connectivity cannot be expressed by sets of sentences in $\mathrm{FHL}\backslash\{E, \mathrm{NOM}, \mathrm{PROP}\}$. However, if we consider only connected frames with loops, the result still holds (Theorem 6). We also defined the fragments π_i and σ_i of HL and showed that the model-checking problem for those fragments are complete problems for the corresponding degree of the polynomial hierarchy.

Acknowledgements

We would like to thank the anonymous referee for its detailed comments and suggestions.

References

[1] C. Areces, P. Blackburn, M. Marx, Hybrid logics: Characterization, interpolation and complexity, Journal of Symbolic Logic 66 (2001) 977–1010.

[2] C. Areces, B. ten Cate, Hybrid logics, in: P. Blackburn, J. van Benthem, F. Wolter (Eds.), Handbook of Modal Logic, Elsevier, 2007, pp. 821–868.

[3] V.C. Barbosa, An introduction to distributed algorithms, The MIT Press, 1996.

[4] M.R.F. Benevides, L.M. Menasché Schechter, Using modal logics to express and check global graph properties, Logic Journal of IGPL 17 (2009) 559–587.

[5] P. Blackburn, M. De Rijke, Y. Venema, Modal Logic, volume 53, Cambridge University Press, 2002.

[6] J. Bradfield, C. Stirling, Modal μ-calculi, in: P. Blackburn, J. van Benthem, F. Wolter (Eds.), Handbook of Modal Logic, volume 3, Elsevier, 2007, pp. 721–756.

[7] B. ten Cate, M. Franceschet, On the complexity of hybrid logics with binders, in: L. Ong (Ed.), Proceedings of Computer Science Logic 2005, volume 3634 of *Lecture Notes in Computer Science*, Springer, 2005, pp. 339–354.

[8] R. Fagin, Generalized first-order spectra and polynomial-time recognizable sets, in: R. Karp (Ed.), Complexity and Computation, volume 7, SIAM-AMS Proceedings, 1974, pp. 43–73.

[9] E. Grädel, Why are modal logics so robustly decidable?, in: G. Paun, G. Rozenberg, A. Salomaa (Eds.), Current Trends in Theoretical Computer Science. Entering the 21st Century, World Scientific, 2001, pp. 393–408.

[10] H.–D. Ebbinghaus and J. Flum and W. Thomas, Mathematical Logic, Springer, 1994.

[11] W. Haken, K. Appel, J. Koch, Every planar map is four colorable, volume 98 of *Contemporary Mathematics*, American Mathematical Society, 1989.

[12] N. Immerman, Descriptive Complexity, Graduate Texts in Computer Science, Springer, 1999.

[13] N.A. Lynch, Distributed algorithms, Morgan Kaufmann, 1996.

[14] C. Papadimitriou, Computational Complexity, John Wiley and Sons Ltd., 2003.

[15] N. Robertson, D. Sanders, P. Seymour, R. Thomas, The four-colour theorem, Journal of Combinatorial Theory, Series B 70 (1997) 2 – 44.

[16] M.Y. Vardi, Why is modal logic so robustly decidable, in: N. Immerman, P.G. Kolaitis (Eds.), Descriptive complexity and finite models, volume 31 of *Series in Discrete Mathematics and Theoretical Computer Science*, American Mathematical Society, 1996, pp. 149–184.