DECIDABLE FRAGMENTS OF FIRST-ORDER AND FIXED-POINT LOGIC

From prefix vocabulary classes to guarded logics*

Erich Grädel

October 2003

Abstract

We survey decidable and undecidable satisfiability problems for fragments of first-order logic and beyond. Classical studies, related to Hilbert's programme and to which László Kalmár has made important contributions, focussed on prefix-vocabulary fragments in first-order logic; for these a complete classification of the decidable and undecidable cases has been obtained.

Modern studies focus on different classes, not necessarily restricted to first-order logic, but including, for instance, fixed-point operators. In particular we discuss decision problems for modal logics, two-variable logics, and guarded logics. Powerful modal logics, such as the modal μ-calculus, are of fundamental importance in practical applications. The more recent concept of guarded logics provides a bridge between modal logic and first-order order, and seems to deliver an explanation for the good algorithmic properties of modal logics.

1 The classical decision problem

At the beginning of the last century, Hilbert formulated the *classical decision problem* for first-order logic: Find an algorithm which, given any first-order sentence, determines whether it is satisfiable. This was an essential part of his formalist programme for the foundations of mathematics and he considered it to be the central problem of mathematical logic. Early results, obtained in the 1920s and the early 1930s, showed that decision algorithms do indeed exist for certain syntactic fragments of first-order logic, such as the monadic class or formula classes that satisfy certain restrictions on the occurrences of quantifiers. Other fragments were shown to be *reduction classes*, which means that the satisfiability problem for arbitrary first-order formulae can be effectively reduced to the satisfiability problem for the fragments. A number of these results were contributed by Lásló Kalmár [21, 22, 23, 24, 25, 26, 27]. However, in 1936 Church and Turing proved that the classical decision problem is algorithmically unsolvable.

*This is a summary of an invited talk presented at the Kalmár Workshop on Logic and Computer Science, October 1-2, 2003 in Szeged (Hungary). The objective of this workshop was to commemorate the work of László Kalmár (1905 - 1976), a pioneer in mathematical logic and computer science, and to present recent research results in the broad area of logic and computer science.

Although the undecidability of first-order logic, together with Gödel's Incompleteness Theorems, showed the impossibility of Hilbert's programme, the classical decision problem did not disappear, but was transformed into a *classification problem*: which formula classes are decidable for satisfiability and which are not? In this new form, the classical decision problem remained of foremost importance in mathematical logic during many decades. Traditionally most attention was given to fragments of first-order logic that are determined by the quantifier prefix and the vocabulary of relation and function symbols. With respect to such fragments the classical decision problem is completely solved, i.e., we now have a complete list of all decidable prefix-vocabulary fragments of first-order logic. Again, Kalmár has made substantial contributions to this classification [28, 29, 30, 31, 32, 33][1]. The final step that completed the classification was obtained by Goldfarb in 1982, when he proved that satisfiability of relational first-order formulae of form $\forall x \forall y \exists z \varphi$ where φ quantifier-free is undecidable, provided that the formulae may contain equality. By the way, this fragment is also historically rather interesting. In the early 1930s, Kalmár [24] and, independently, Gödel and Schütte, had established, that satisfiability of relational formulae with quantifier prefix $\exists x_1 \cdots \exists x_m \forall y_1 \forall y_2 \exists z_1 \cdots \exists z_n$, *without equality*, is decidable. Moreover, Gödel [9] proved that this class has the finite model property (which means that every satisfiable formula in the class has a finite model). In the last paragraph of this paper he claimed, without substantiation, that his proof persists in the presence of equality. Only in the 1960s it became apparent that there are serious problems with the extension to the case with equality, and eventually, almost fifty years after Gödel's paper, Goldfarb disproved Gödels claim showing that even the $\forall\forall\exists$-prefix class is undecidable for satisfiability, once equality is allowed.

2 The complete classification for prefix vocabulary classes

We summarise the solution of the classical decision problem with respect to prefix vocabulary classes. These classes are denoted in the form $[\Pi, (p_1, p_2, \dots), (f_1, f_2, \dots)]$ or $[\Pi, (p_1, p_2, \dots), (f_1, f_2, \dots)]_=$ where Π is a word over $\{\exists, \forall, \exists^*, \forall^*\}$ denoting a set of quantifier prefixes, and where $p_i, f_i \leq \omega$ are bounds on the number of relation symbols and functions symbols of arity i that may appear in the formulae. The presence or absence of the index $_=$ indicates whether or not the formulae may contain equality. Some obvious abbreviations are used: in $(p_1, p_2, \dots)$ and $(f_1, f_2, \dots)$ infinite sequences of zeroes are omitted, i.e., we write $(0,1)$ instead of $(0,1,0,0,\dots)$ and (ω) instead of $(\omega,0,0,\dots)$; further we write *"all"* if there are no restrictions. For instance $[\forall\exists^*, (0,1), (\omega)]$ is the class of all first-order sentences, without equality, of form $\forall x \exists y_1 \cdots \exists y_n \varphi$, with arbitrary n, where φ is quantifier-free and whose vocabulary consists of a binary relation and an arbitrary number of unary functions. Similarly, $[\exists^*, \mathit{all}, \mathit{all}]_=$ is the the set of existential first-order sentences, without restriction on the vocabulary. The following tables describe the complete classification.

[1]For more detailed information, consult the annotated bibliography in [5]

2.1 Undecidable Cases

A: Pure predicate logic (without functions, without =)

(1)	$[\forall\exists\forall, (\omega, 1), (0)]$	(Kahr 1962)
(2)	$[\forall^3\exists, (\omega, 1), (0)]$	(Surányi 1959)
(3)	$[\forall^*\exists, (0, 1), (0)]$	(Kalmár-Surányi 1950)
(4)	$[\forall\exists\forall^*, (0, 1), (0)]$	(Denton 1963)
(5)	$[\forall\exists\forall\exists^*, (0, 1), (0)]$	(Gurevich 1966)
(6)	$[\forall^3\exists^*, (0, 1), (0)]$	(Kalmár-Surányi 1947)
(7)	$[\forall\exists^*\forall, (0, 1), (0)]$	(Kostyrko-Genenz 1964)
(8)	$[\exists^*\forall\exists\forall, (0, 1), (0)]$	(Surányi 1959)
(9)	$[\exists^*\forall^3\exists, (0, 1), (0)]$	(Surányi 1959)

B: Classes with functions or equality

(10)	$[\forall, (0), (2)]_=$	(Gurevich 1976)
(11)	$[\forall, (0), (0, 1)]_=$	(Gurevich 1976)
(12)	$[\forall^2, (0, 1), (1)]$	(Gurevich 1969)
(13)	$[\forall^2, (1), (0, 1)]$	(Gurevich 1969)
(14)	$[\forall^2\exists, (\omega, 1), (0)]_=$	(Goldfarb 1984)
(15)	$[\exists^*\forall^2\exists, (0, 1), (0)]_=$	(Goldfarb 1984)
(16)	$[\forall^2\exists^*, (0, 1), (0)]_=$	(Goldfarb 1984)

2.2 Decidable Cases

Note that prefix-vocabulary classes with finite quantifier prefix and finite relational vocabulary are decidable for trivial reasons. Indeed, any such class contains, up to trivial syntactic reformulations, only a finite number of sentences, so the satisfiability problem can be solved by looking up the answer in a finite table. The remaining decidable cases are the following.

A: Classes with the finite model property

(1)	$[\exists^*\forall^*, all, (0)]_=$	(Bernays, Schönfinkel 1928)
(2)	$[\exists^*\forall^2\exists^*, all, (0)]$	(Gödel 1932, Kalmár 1933, Schütte 1934)
(3)	$[all, (\omega), (\omega)]$	(Löb 1967, Gurevich 1969)
(4)	$[\exists^*\forall\exists^*, all, all]$	(Gurevich 1973)
(5)	$[\exists^*, all, all]_=$	(Gurevich 1976)

B: Classes with infinity axioms

$$(6) \qquad [all, (\omega), (1)]_= \qquad \text{(Rabin 1969)}$$

$$(7) \qquad [\exists^*\forall\exists^*, all, (1)]_= \quad \text{(Shelah 1977)}$$

It is a simple, but a bit lengthy and boring exercise, to prove that this classification is indeed complete, i.e. every prefix vocabulary class is either contained in one of the decidable classes, or contains one of the undecidable classes in this list. There exist numerous variations and ramifications of this classification. For almost all of the decidable classes, also the complexity has been precisely determined. Further there also is an almost complete classification of the classes that have the finite model property as opposed to those that contain infinity axioms. A comprehensive account of all these results is given in [5].

Why prefix-vocabulary classes? It may be asked, why the traditional studies on Hilbert's decision problem focussed so strongly on prefix-vocabulary classes. A partial answer is of course that restrictions on quantifier prefixes and vocabulary provide natural and straightforward syntactic fragments. Another partial answer is purely historical: the early partial positive and negative results that had been obtained in the 1920s and 1930s were mostly in terms of these classes. After Church and Turing had solved the decision problem, it therefore seemed natural to extend the early results to a complete classification along the same lines, and variations of them.

But of course, there are other, related traditions. One of them is the classification of *mathematical theories*, such the theories of groups, rings, fields, of various variants of arithmetic, and many others, according to decidability and undecidability. We will not discuss this issue here. A related theme is the study of *monadic second-order theories*, which started in the 1960s with the work of Büchi, Elgot, McNaughton, and Rabin, and which is closely related to automata theory, A result of fundamental importance is Rabin's Tree Theorem, saying that the monadic theory of the infinite binary tree is decidable. This has numerous applications, some of them will be mentioned below. Further, there are appeared other logical formalisms, inside and outside of the classical first-order framework, and it has turned out that some of these are very relevant for practical applications. In particular, fixed-point logics have turned out to be very important for computer science applications. We will discuss here the class of modal logics and a recent extension, the class of guarded logics (with and without fixed points).

3 Modal logics

Prefix-vocabulary classes in first-order logic are not the only, and with respect to applications probably not even the most relevant formula classes, whose decidability deserves to be investigated. In the last decades, attention has shifted to logical formalisms that are somewhat orthogonal to the classical fragments of first-order logic. From a practical point of view, *modal and temporal logics* provide a very import class of logical systems with decidable satisfiablity problems. The simplest case is propositional modal logic ML (also called K_n) which extends propositional logic by the possibility to construct formulae $\langle a \rangle \psi$

and $[a]\psi$ (for any a from a given set A of 'actions' or 'modalities') with the meaning that ψ holds at some, respectively each, a-successor of the current state. (We refer to [4] or [37] for background on modal logic).

Although ML is formally a propositional logic we view it as a fragment of first-order logic. Kripke structures, which provide the semantics for modal logics, are really just labelled graphs, i.e., relational structures $\mathcal{K} = (V, (E_a)_{a\in A}, (P_i)_{i\in I})$, whose elements $v \in V$ are called states (or worlds), with binary relations E_a describing transitions between states, and unary relations P_i defining the set of states at which atomic propositions hold. There is a standard translation taking every formula $\psi \in$ ML to a first-order formula $\psi^*(x)$ with one free variable, such that for every Kripke structure $\mathcal{K}$ with a distinguished node w we have that $\mathcal{K}, w \models \psi$ if and only if $\mathcal{K} \models \psi^*(w)$. This translation takes an atomic proposition P_i to the atom $P_i x$, it commutes with the Boolean connectives, and it translates the modal operators by quantifiers as follows:

$$\begin{array}{lcl} \langle a\rangle\psi & \rightsquigarrow & (\langle a\rangle\psi)^*(x) := \exists y(E_a xy \wedge \psi^*(y)) \\ {[a]}\psi & \rightsquigarrow & ([a]\psi)^*(x) := \forall y(E_a xy \rightarrow \psi^*(y)) \end{array}$$

where E_a is the transition relation associated with the modality a. The *modal fragment* of first-order logic is the image of propositional modal logic under this translation.

Propositional modal logic has very convenient model-theoretic and algorithmic properties. In particular, in can be decided in time $O(\|\mathcal{K}\| \cdot |\psi|)$ whether a given formula $\psi \in$ ML holds at given state of a Kripke structure $\mathcal{K}$, and the satisfiability problem for ML is decidable and PSPACE-complete [34]. On the other side, ML has very limited expressive power. Some of its limitations are:

- ML does not have equality, so we cannot say that an a-transition from the current state brings us to the same state as a b-transition.
- ML does not have unbounded quantification, so we cannot say that something holds at all states.
- In ML we cannot define new transition relations, not even on the quantifier-free level. For instance, we cannot say that the current state is reachable from some other state w by both an a-transition and a b-transition (because there is no way to define the new transition $Exy \equiv E_a yx \wedge E_b yx$).
- ML has no counting mechanism whatsoever. For instance we cannot say that there is at most one a-transition from the current state, hence we cannot distinguish nondeterministic from deterministic systems.
- Perhaps the most important limitation is that there is no recursion mechanism. With an ML-formula we can only look from the current state along a bounded number of transition steps, but for reasoning about computation, reachability problems (in various forms) are essential. Note however that, contrary to the other limitations, this is also a limitation of first-order logic.

These limitations (in particular the last one) have motivated the study of powerful extensions of modal logics that are expressive enough for computationally interesting properties and still algorithmically manageable. Today, many of these logics are of enormous practical importance such as computation tree logic CTL and the modal μ-calculus. Another interesting class of such languages are the so-called *description logics* [6] used in knowledge representation (to mention a different application area); these are also basically modal logic, although this is not immediately apparent from their 'official' definition. Rather than giving precise definitions, let us just look at some examples for CTL and the μ-calculus. CTL extends ML by path quantifiers, and by temporal operators like **next**, **eventually** and **until** on paths. So you can express statements about infinite computation paths in a transition system, such as "on all paths, eventually φ" or "never, a deadlock state can be reached" and so on. The μ-calculus is an even more powerful logic than CTL, that encompasses also many other logics such as PDL, LTL and CTL*. It extends ML by least and greatest fixed points. For instance, the formula $\mu X.(\varphi \vee [a]X)$ says that the current state is in the least fixed point of the operator $X \mapsto \{v : (\mathcal{K}, X), v \models \varphi \vee [a]X\}$. To understand the meaning of this formula, observe that the operator maps X to the set of states where either φ is true or all a-transitions lead to a state that belongs to X. The least fixed point is the set of states where all a-paths eventually terminate or hit a state where φ holds. Another example is $\nu X.(\bigvee_{a\in A}\langle a\rangle \mathit{true} \wedge \bigwedge_{a\in A}[a]X)$ where ν stands for the greatest fixed point. This formula says that the current state is in the greatest fixed point of the operator that assigns to X the set of states that are not deadlock states (i.e. have at least one outgoing transition) and where all outgoing transitions lead to a state in X. Hence the formula says that from the current state no deadlock will ever be reached.

Of course, it is not difficult to define very expressive logics (for instance, we could just take second-order logic) and expressive power by itself does not mean that a logic is useful. What is important for many application areas is that there is good balance between expressive power and algorithmic properties. This is indeed the case for modal logics, because the above-mentioned extensions of ML remain algorithmically manageable. Model checking is very fast for CTL and, more generally, for such formulae of the μ-calculus that do not contain too many alternations of least and greatest fixed points (in practice we rarely need formulae with more than two or three alternations). For satisfiability problems, we cannot expect very fast algorithms. As we have seen, satisfiability is undecidable already for rather small fragments of first-order logic, and even for plain propositional logic, satisfiability is NP-complete. For more powerful logics these problems are at least PSPACE- or EXPTIME-hard. Actually, one has to add very little to propositional modal logic to get an EXPTIME-hard formalism. Keeping this in mind, the following results [7, 8] on CTL and the μ-calculus are good news.

Theorem 3.1. *The satisfiability problems for CTL and the μ-calculus are* EXPTIME*-complete.*

Both logics also have the *finite model property.* In [39] Vardi adressed the question to identify the main reasons for these "robust decidability properties of modal logics".

4 Two-variable logics

Looking more closely at the standard translation from ML into FO, one notices that only a very small part of first-order logic is needed. Suppose that we translate $\langle a\rangle\psi$ into $(\langle a\rangle\psi)^*(x) := \exists y(E_a xy \wedge \psi^*(y))$. Then, inside the construction of $\psi^*(y)$, when we come to the translation of the next modal operator, we can re-use x and do not need a new variable. So the entire translation can be done with only two variables. Indeed it has been noticed already in the 1970s that ML can be embedded into FO^2, the *two-variable fragment* of first-order logic. This is interesting, because Mortimer has proved in 1975 [36] that FO^2 has the finite model property and that therefore, the satisfiability problem for FO^2 is decidable. For a simpler proof of these results (with optimal complexity bounds) see [15].

Traditionally the embedding of ML into FO^2 has been proposed as an explanation for the good properties of modal logic. But it has turned out that this explanation is not really convincing.

First, FO^2 is more complex. While the satisfiability of ML-formulae is in PSPACE, the same problem for FO^2 is NEXPTIME-complete. ML has a number of nice model-theoretic properties that fail for FO^2 [1, 18]. But most importantly, FO^2 is not nearly as robustly decidable as modal logic, which becomes apparent when we study the logics that extend FO^2 in a similar way as CTL or the μ-calculus extend basic modal logic. For instance one can add transitive closure operators, a variant of path quantification, or least and greatest fixed point operators to FO^2. But even if one does this in the most restrictive conceivable way, the resulting logics become undecidable, in fact undecidable of very high degree (Σ^1_1-hard) [16, 17].

What are the reasons why modal logic remains decidable when we add operators like path quantification or fixed points, and FO^2 does not? In this context Vardi [39] has insisted on the importance of the *tree model property* of modal logics: every satisfiable formula has a model that is a tree. In fact, there even is a tree model with bounded branching. Surprisingly, the tree model property often seems to be even more useful than the finite model property for obtaining good complexity results and for the design of practical algorithms (which is not the same thing). If we do not insist on bounded branching of the underlying tree, the tree model property is an immediate consequence of the bisimulation-invariance of modal logics. Take a Kripke structure $\mathcal{K}$ that satisfies the given formula ψ at some state v. Then build a tree $\mathcal{K}'$ by 'unraveling' $\mathcal{K}$ from v. That is, consider the tree of all finite paths through $\mathcal{K}$ that start at v, and consider the end-point of each finite path as a separate node (so when two paths in $\mathcal{K}$ from v lead to the same point w, then there will be two copies of w in $\mathcal{K}'$ and two different paths never meet. The unraveled structure at its root is bisimilar to the original structure at state v. Modal logics (including CTL and the μ-calculus) are invariant under bisimulation, hence ψ is satisfied at the root of the tree structure.

On the other hand, FO^2 does not have the tree model property and the extensions of FO^2 mentioned above are powerful enough to axiomatize grids and to encode undecidable tiling problems or halting problems [16].

The tree model property is algorithmically important because it paves the way for the use of powerful decidability results like Rabin's Theorem and for the use of automata. One can reformulate μ-calculus formulae as monadic second-order formulae and then use the decidability of SωS, the monadic second-order theory of the ω-branching tree [38]. This gives a reasonably simple proof of the decidability of the μ-calculus. However, it does not yield good complexity bounds, since the decision complexity of SωS is non-elementary, i.e., exceeds any bounded number of iterations of the exponential function. But the proof of Rabin's Theorem relies on tree automata, and by using automata directly for the μ-calculus, we can indeed get good algorithms, both for model checking problems and for satisfiability problems (see [19] for more information).

So the tree model property gives us a partial answer to the question why modal logics are robustly decidable [39]. We would like to generalize it, and we would like to have more powerful logics than ML, CTL and even the μ-calculus that retain the nice properties of modal logics.

5 Guarded logics

Let us look again at the standard translation from modal logic into first-order logic. Apart from the fact that it never needs more than two variables there is another crucial observation: the quantifiers are always relativized by atoms.

This motivated Andréka, van Benthem and Németi [1] to study more powerful fragments of first-order logic based on the notion of *guarded quantification.* This has then be extended, for instance in [2, 13, 20, 14] to more general concepts of guarded logics, defined by restricting quantification in first-order logic, second-order logic, fixed point logics or infinitary logics in such a way that, semantically speaking, each subformula can 'speak' only about elements that are 'very close together' or 'guarded'.

Syntactically this means that all first-order quantifiers are relativised by certain 'guard formulae' that tie together all the free variables in the scope of the quantifier. Quantification is of the form

$$\exists \boldsymbol{y}(\alpha(\boldsymbol{x},\boldsymbol{y}) \wedge \psi(\boldsymbol{x},\boldsymbol{y})) \qquad \text{or} \qquad \forall \boldsymbol{y}(\alpha(\boldsymbol{x},\boldsymbol{y}) \rightarrow \psi(\boldsymbol{x},\boldsymbol{y}))$$

where quantifiers may range over a tuple $\boldsymbol{y}$ of variables, but are 'guarded' by a formula α that must contain all the free variables of the formula ψ that is quantified over. The guard formulae are of a simple syntactic form (in the basic version, they are just atoms). In this survey we only discuss the logic GF, the *guarded fragment* of first-order logic, as it was introduced by Andréka, van Benthem and Nemeti [1], and its extension to guarded fixed point logic μGF, introduced in [20]. We will not discuss *loosely guarded* [2], *clique-guarded* [13], *action-guarded* [3, 11] and *packed* [35] logics, which are based on more liberal conditions for guard formulae; most of the techniques and results presented here extend to these more general variants of guarded logics.

Definition 5.1. The *guarded fragment* of first-order logic, GF is defined inductively as follows:

(1) Every relational atomic formula $Rx_{i_1}\cdots x_{i_m}$ or $x_i = x_j$ belongs to GF.

(2) GF is closed under boolean operations.

(3) If $\boldsymbol{x},\boldsymbol{y}$ are tuples of variables, $\alpha(\boldsymbol{x},\boldsymbol{y})$ is a positive atomic formula and $\psi(\boldsymbol{x},\boldsymbol{y})$ is a formula in GF such that $\text{free}(\psi) \subseteq \text{free}(\alpha) = \boldsymbol{x} \cup \boldsymbol{y}$, then also the formulae

$$\exists \boldsymbol{y}(\alpha(\boldsymbol{x},\boldsymbol{y}) \wedge \psi(\boldsymbol{x},\boldsymbol{y})) \quad \text{and} \quad \forall \boldsymbol{y}(\alpha(\boldsymbol{x},\boldsymbol{y}) \rightarrow \psi(\boldsymbol{x},\boldsymbol{y}))$$

belong to GF.

Here free(ψ) means the set of free variables of ψ. An atom $\alpha(\boldsymbol{x}, \boldsymbol{y})$ that relativizes a quantifier as in rule (3) is the *guard* of the quantifier. Hence in GF, guards must be atoms. But the really crucial property of guards (also for more powerful guarded logics) is that they must contain *all* free variables of the formula that is quantified over.

The main motivation for introducing the guarded fragment was to explain and generalize the good algorithmic and model-theoretic properties of modal logics. Clearly the translation of modal logic into first-order logic uses only guarded quantification, so we see immediately that the modal fragment is contained in GF. The guarded fragment generalizes the modal fragment by dropping the restrictions to use only two variables and only monadic and binary predicates, and retains only the restriction that quantifiers must be guarded.

The following properties of GF have been demonstrated [1, 12]:

(1) The satisfiability problem for GF is decidable.

(2) GF has the finite model property, i.e., every satisfiable formula in the guarded fragment has a finite model.

(3) GF has (a generalized variant of) the tree model property.

(4) Many important model theoretic properties which hold for first-order logic and modal logic, but not, say, for the bounded-variable fragments FO^k, do hold also for the guarded fragment.

(5) The notion of equivalence under guarded formulae can be characterized by a straightforward generalization of bisimulation.

Based on this kind of results Andréka, van Benthem and Németi put forward the 'thesis' that it is the guarded nature of quantification that is the main responsible for the good model-theoretic and algorithmic properties of modal logics.

Let us discuss to what extent this explanation is adequate. One way to address this question is to look at the complexity of GF. We have shown in [12] that the the satisfiability problem for GF is complete for 2EXPTIME, the class of problems solvable by a deterministic algorithm in time $2^{2^{p(n)}}$, for some polynomial $p(n)$. This seems very bad, in particular if we compare it to the well-known fact that the satisfiability problem for propositional modal logic is in PSPACE [34]. But dismissing the explanation of Andréka, van Benthem and Németi on these grounds would be too superficial. Indeed, the reason for the double exponential time complexity of GF is just the fact that predicates may have unbounded arity (wheras ML only expresses properties of graphs). Given that even a single predicate of arity n over a domain of just two element leads to 2^{2^n} possible types already on the atomic level, the double exponential lower complexity bound is hardly a surprise. Further, in most of the potential applications of guarded logics the arity of the relation symbols is bounded. But for GF-sentences of bounded arity, the satisfiability problem can be decided in EXPTIME [12], which is a complexity level that is reached already for rather weak extensions of ML (e.g. by a universal modality). Thus, the complexity analysis does not really provide a decisive answer to our question.

To approach the question from a different angle, let us look at extensions of ML. As pointed out above, ML is a very weak logic, and the really interesting modal logics extend ML by features like path quantification, temporal operators, least and greatest fixed points etc. The most important of these extension is the modal μ-calculus L_μ, which extends ML by least and greatest fixed points. Therefore, a good test for the explanation put forward by Andréka, van Benthem and Németi is the following problem:

> If we extend GF by least and greatest fixed points, do we still get a decidable logic? If yes, what is its complexity? To put it differently, what is the penalty, in terms of complexity, that we pay for adding fixed points to the guarded fragment?

So let us look at guarded fixed point logic, denoted μGF. It is defined by adding to GF the following rule for defining fixed point formulae:

Let W be a k-ary relation variable and let $\boldsymbol{x} = x_1, \ldots, x_k$ be a k-tuple of distinct variables. Further, let $\psi(W, \boldsymbol{x})$ be a guarded formula where W appears only positively and not in guards and that contains no free variables other than $\boldsymbol{x}$. Then we can build the formulae

$$[\mathbf{lfp}\, W\boldsymbol{x} \,.\, \psi](\boldsymbol{x}) \quad \text{and} \quad [\mathbf{gfp}\, W\boldsymbol{x} \,.\, \psi](\boldsymbol{x}).$$

The semantics is the usual one: The formula $[\mathbf{lfp}\, W\boldsymbol{x} \,.\, \psi(W, \boldsymbol{x})](\boldsymbol{a})$ is true in a structure $\mathfrak{A}$ (with universe A) if and only if the tuple $\boldsymbol{a}$ is in the least fixed point of the operator

$$\psi^{\mathfrak{A}} : W \mapsto \psi^{\mathfrak{A}}(W) := \{\boldsymbol{a} \in A^k : \mathfrak{A} \models \psi(W, \boldsymbol{a})\}.$$

Similarly for the greatest fixed points.

Here is an instructive example of a guarded fixed point sentence.

$$\exists xyFxy \wedge \forall xy(Fxy \rightarrow \exists xFyx) \wedge \forall xy(Fxy \rightarrow [\mathbf{lfp}\, Wx \,.\, \forall y(Fyx \rightarrow Wy)](x)).$$

The first two conjuncts say that there exists an F-edge and that every F-edge can be extended to an infinite path. The third conjunct says that each point x on an F-edge is in the least fixed point of the operator $W \mapsto \{w : \text{ all } F\text{-predecessors of } w \text{ are in } W\}$. Hence the least fixed point is the set of points that have only finitely many F-predecessors. Thus, the sentence says that there is an infinite forward F-chain, but no infinite backwards F-chain. This means in particular that F does not cycle, so this sentence has only infinite models! As a consequence, guarded fixed point logic does not have the finite model property.

One might conjecture then that the satisfiability problem for μGF is undecidable. But this is not the case. In [20] we were able to show that the model-theoretic and algorithmic methods that are available for the μ-calculus on one side, and the guarded fragments of first-order logic on the other side, can indeed be combined and generalized to provide positive results for guarded fixed point logics.

Theorem 5.2 (Grädel, Walukiewicz). *The satisfiability problems for guarded fixed point logic is decidable and* 2EXPTIME*-complete. For guarded fixed point sentences of bounded width the satisfiability problem is* EXPTIME*-complete.*

By the width of a formula ψ, we mean the maximal number of free variables in the subformulae of ψ. For sentences that are guarded in the sense of GF, the width is bounded by the maximal arity of the relation symbols, but there are other variants of guarded logics where the width may be larger. Note that for guarded fixed point sentences of bounded width the complexity level is the same as for μ-calculus and for GF without fixed points.

The proof is based on alternating tree automata. It is shown that for every sentence $\psi \in \mu$GF one can build an automaton that accepts a tableau-representation of a structure if, and only if, that structure is a model of ψ. In this way the satisfiability problem for μGF is reduced to the emptiness problem for a particular variant of tree automata, a problem that can be shown to be in EXPTIME.

But the intuitive reason for the nice behaviour of μGF is that it has a generalized variant of the *tree model property*. Of course, guarded logics do not talk about graphs, but about general relational structures, so the tree model property in its strict form would not even make sense. The generalised variant of the tree model property is formulated in terms of *tree width*.

A structure has tree width k if it can be covered by a tree-shaped arrangement of (in general overlapping) substructures of size at most $k+1$. More formally, if there exists a tree T and for every node v of T a substructure $\mathfrak{F}(v) \subseteq \mathfrak{A}$ with no more than $k+1$ elements such that (1) $\mathfrak{A} = \bigcup_{v \in T} \mathfrak{F}(v)$ and (2) for each element a of $\mathfrak{A}$, the set of nodes v such that $a \in \mathfrak{F}(v)$ is connected (i.e. induces a subtree of T). Trees and forests have tree width 1. Cycles have tree width 2. A simple class of structures with unbounded tree width are finite rectangular grids. Intuitively, the tree width of structure measures how closely the structure ressembles a tree.

Theorem 5.3 (Tree model property). *Every satisfiable μGF-formula of width k has a model of tree width at most $k-1$.*

Recall that to prove that a problem is undecidable one usually encodes in it some undecidable tiling problem on grids. In a logic with this generalized tree model property we cannot axiomatize grids since these do not have bounded tree width. Further, structures of bounded tree width can be interpreted in trees with a finite set of labels. Then again one can use Rabin's Theorem to prove decidability, and tree automata to get good complexity bounds.

But how can one see that guarded fixed point logic has the generalised tree model property? One again generalises the methodology from modal logic. Modal logics have the tree model property because they are invariant under bisimulations and because every transition system can be unravelled to a bisimilar tree. The notion of bisimulations can be generalized to guarded bisimulations. In [1] it is shown that the properties expressible in GF are precisely the first-order definable properties that are invariant under guarded bisimulations. Also guarded fixed point logic reamins invariant under guarded bisimulations. Then one can proceed as in the modal case and 'unravel' a structure in this case not to a tree but to a structure of small tree width. There is a guarded bisimulation between a structure and its unraveling. Hence if there is any model of a guarded fixed point sentence then there also is one of small tree width.

In view of the positive decidability results and the model-theoretic properties of guarded logics it indeed seems that the guarded nature of quantification in modal logics provides a good explanation for their good algorithmic properties. Guarded logics provide an interesting and general class of logics that preserve most of these properties on a higher level of expressiveness and within the more general framework of first-order logic and fixed-point logic on arbitrary relational structures.

References

[1] H. Andréka, J. van Benthem, and I. Németi, *Modal languages and bounded fragments of predicate logic*, Journal of Philosophical Logic, 27 (1998), pp. 217–274.

[2] J. van Benthem, *Dynamic bits and pieces*, ILLC research report, University of Amsterdam, 1997.

[3] J. van Benthem, *Modal logic in two gestalts*, in Advances in Modal Logic. Proceedings of AiML 98, CSLI Publications, 2000.

[4] P. Blackburn, M. de Rijke, and Y. Venema, *Modal Logic*, Cambridge University Press, 2001.

[5] E. Börger, E. Grädel, and Y. Gurevich, *The Classical Decision Problem*, Springer Verlag, 1997.

[6] F. Donnini, M. Lenzerini, D. Nardi, and A. Schaerf, *Reasoning in description logics*, in Principles of Knowledge Representation, G. Brewka, ed., CSLI Publications, 1996, pp. 193–238.

[7] A. Emerson and J. Halpern, *Decision procedures and expressiveness in the temporal logic of branching time*, Journal of Computer and System Sciences, 30 (1985), pp. 1–24.

[8] A. Emerson and C. Jutla, *The complexity of tree automata and logics of programs*, in Proc. 29th IEEE Symp. on Foundations of Computer Science, 1988, pp. 328–337.

[9] K. Gödel, *Zum Entscheidungsproblem des logischen Funktionenkalküls*, Monatshefte f. Mathematik u. Physik, 40 (1933), pp. 433–443. Reprinted in [10, pp. 306–326].

[10] K. Gödel, *Collected Works, Vol. I: Publications 1929–1936*, Oxford University Press, 1986. Edited by S. Feferman, J. Dawson jr., S. Kleene, G. Moore, R. Solovay and J. van Heijenoort.

[11] E. Goncalves and E. Grädel, *Decidability issues for action guarded logics*, in Proceedings of 2000 International Workshop on Description Logics - DL2000, 2000, pp. 123–132.

[12] E. Grädel, *On the restraining power of guards*, Journal of Symbolic Logic, 64 (1999), pp. 1719–1742.

[13] E. Grädel, *Guarded fixed point logics and the monadic theory of countable trees*, Theoretical Computer Science, 288 (2002), pp. 129 – 152.

[14] E. Grädel, C. Hirsch, and M. Otto, *Back and forth between guarded and modal logics*, ACM Transactions on Computational Logic, 3 (2002), pp. 418–463.

[15] E. Grädel, P. Kolaitis, and M. Vardi, *On the decision problem for two-variable first-order logic*, The Bulletin of Symbolic Logic, 3 (1997), pp. 53–69.

[16] E. Grädel and M. Otto, *On logics with two variables*, Theoretical Computer Science, 224 (1999), pp. 73–113.

[17] E. Grädel, M. Otto, and E. Rosen, *Two-Variable Logic with Counting is Decidable*, in Proceedings of 12th IEEE Symposium on Logic in Computer Science LICS '97, Warschau, 1997.

[18] E. Grädel and E. Rosen, *Preservation theorems for two-variable logic*, Mathematical Logic Quarterly, 45 (1999), pp. 315–325.

[19] E. Grädel, W. Thomas, and T. Wilke, eds., *Automata, Logics, and Infinite Games*, Lecture Notes in Computer Science Nr. 2500, Springer, 2002.

[20] E. Grädel and I. Walukiewicz, *Guarded fixed point logic*, in Proc. 14th IEEE Symp. on Logic in Computer Science, 1999, pp. 45–54.

[21] L. Kalmár, *Eine Bemerkung zur Entscheidungstheorie*, Acta Scientiarum Mathematicarum Universitatis Szegediensis, 4 (1929), pp. 248–252.

[22] L. Kalmár, *Ein Beitrag zum Entscheidungsproblem*, Acta Scientiarum Mathematicarum Universitatis Szegediensis, 5 (1932), pp. 222–236.

[23] L. Kalmár, *Zum Entscheidungsproblem der mathematischen Logik*, in Verhandlungen des Internationalen Mathematischen Kongresses, vol. II, Zürich, 1932, pp. 337–338.

[24] L. Kalmár, *Über die Erfüllbarkeit derjenigen Zählausdrücke, welche in der Normalform zwei benachbarte Allzeichen enthalten*, Mathematische Annalen, 108 (1933), pp. 466–484.

[25] L. Kalmár, *Über einen Löwenheimschen Satz*, Acta Scientiarum Mathematicarum Universitatis Szegediensis, 7 (1934), pp. 112–121.

[26] L. Kalmár, *Zur Reduktion des Entscheidungsproblems*, Norsk Mat. Tidsskrift, (1937), pp. 1–10.

[27] L. Kalmár, *On the reduction of the decision problem*, Journal of Symbolic Logic, 4 (1939), pp. 1–9.

[28] L. Kalmár, *Contributions to the reduction theory of the decision problem*, Acta Mathematica Academiae Scientiarunm Hunagricae, 1 (1950), pp. 64–73. First paper: Prefix $(x_1)(x_2)(Ex_3)\cdots(Ex_{n-1})(x_n)$, a single binary predicate.

[29] L. Kalmár, *Contributions to the reduction theory of the decision problem*, Acta Mathematica Academiae Scientiarunm Hunagricae, 2 (1951), pp. 19–38. Third paper: Prefix $(x_1)(Ey_1)\cdots(Ey_n)(x_2)(x_3)$, a single binary predicate.

[30] L. Kalmár, *Contributions to the reduction theory of the decision problem*, Acta Mathematica Academiae Scientiarunm Hunagricae, 2 (1951), pp. 125–142. Fourth paper: Reduction to the case of a finite set of individuals.

[31] L. Kalmár, *Ein direkter Beweis für die allgemein-rekursive Unlösbarkeit des Entscheidungsproblems*, Zeitschr. f. math. Logik u. Grundlagen d. Math., 2 (1956), pp. 1–14.

[32] L. Kalmár and J. Surányi, *On the reduction of the decision problem, second paper: Gödel prefix, a single binary predicate*, Journal of Symbolic Logic, 12 (1947), pp. 65–73.

[33] L. Kalmár and J. Surányi, *On the reduction of the decision problem, third paper: Pepis prefix, a single binary predicate*, Journal of Symbolic Logic, 15 (1950), pp. 161–173.

[34] R. Ladner, *The computational complexity of provability in systems of propositional modal logic*, SIAM Journal on Computing, 6 (1977), pp. 467–480.

[35] M. Marx, *Tolerance logic*, Tech. Rep. IR-469, Faculteit der Exacte Wetenschappen, Vrije Universiteit Amsterdam, 1999.

[36] M. Mortimer, *On languages with two variables*, Zeitschrift für Mathematische Logik und Grundlagen der Mathematik, 21 (1975), pp. 135–140.

[37] S. Popcorn, *First Steps in Modal Logic*, Cambridge University Press, 1994.

[38] M. Rabin, *Decidability of second-order theories and automata on infinite trees*, Transactions of the AMS, 141 (1969), pp. 1–35.

[39] M. Vardi, *Why is modal logic so robustly decidable?*, in Descriptive Complexity and Finite Models, N. Immerman and P. Kolaitis, eds., vol. 31 of DIMACS Series in Discrete Mathematics and Theoretical Computer Science, AMS, 1997, pp. 149–184.