Semiring Provenance for Guarded Logics

Katrin M. Dannert and Erich Grädel

1 Introduction

In this paper we bring together two different areas of mathematical logic, both of which are relevant for computer science: *guarded logics* and *provenance analysis on the basis of commutative semirings*. Guarded logics are fragments of standard logical systems such as first-order logic, fixed-point logic, or second-order logic, in which quantification is restricted in such a way that formulae can only talk about tuples whose elements are, in some sense, close together. In the standard guarded fragment of first-order logic (GF), which we shall discuss in more detail in the next section, these elements must co-exist in some atomic fact. In a graph this would mean that formulae can only refer to single nodes and to edges (or inverse edges), but neither to triples, quadruples etc. of elements, nor to pairs of nodes that are not adjacent. But of course, guarded logics are not restricted to graphs, and co-existence in an atomic fact becomes more interesting and more general in structures with relations of larger arity. Further there are also guarded logics where the notion of "close together" has a more general meaning, for instance that the tuple induces a clique in the Gaifman graph of the structure. A further rather powerful extension is based on the idea to guard negation rather than quantification. A strong reason for studying guarded logics is that they have very interesting and convenient algorithmic and model-theoretic properties. In this paper, we shall just consider guarded fragments of first-order logic, but in principle, our approach extends also to guarded fixed-point logics.

Katrin M. Dannert
Mathematische Grundlagen der Informatik, RWTH Aachen University, Germany, e-mail: dannert@logic.rwrth-aachen.de. Supported by the DFG RTG 2236 UnRAVeL.

Erich Grädel
Mathematische Grundlagen der Informatik, RWTH Aachen University, Germany, e-mail: graedel@logic.rwrth-aachen.de.

Provenance analysis on the other side is an approach that has originally been developed in database theory. It aims at understanding how the result of a computational process with a complex input, consisting of multiple items, depends on the various parts of this input. Specifically, provenance analysis based on interpretations in commutative semirings has been developed for positive database query languages, to understand which combinations of the atomic facts in a database can be used for deriving the result of a given query. In this approach, atomic facts are interpreted not just by true or false, but by values in an appropriate semiring. These values are then propagated from the atomic facts to arbitrary queries in the language, which permits to answer questions such as the minimal cost of a query evaluation, the confidence one can have that the result is true, or the clearance level that is required for obtaining the output. Semiring provenance has recently extended by Grädel and Tannen [12, 13] to logics with negation, notably first-order logic, dealing with negation by transformation into negation normal form and by semirings of polynomials with a duality on the indeterminates. Here we develop this approach further for the guarded fragment (GF), introduced by Andréka, van Benthem and Németi, as well as for the guarded negation fragment (GNF). Guarded quantification permits to control the complexity of the semiring computations since once has to take sums or products only over those tuples of elements that appear in the guards.

Provenance analysis of logics is intimately connected to provenance analysis of games. In the same way as formula evaluation or model checking can be formulated in game theoretic terms, also the propagation of provenance values from atomic facts to arbitrary formulae can be viewed as a process on the associated games. Moreover, provenance analysis of games is of independent interest, and provenance values of positions in a game provide detailed information about the number and properties of the strategies of the players, far beyond the question whether a player has a winning strategy from a given position. We discuss provenance of games here just in terms of a particularly simple case of games, namely finite acyclic games, which are sufficient for first-order logic and its fragments. Our approach relates the provenance analysis of modal and guarded logics to the provenance analysis of the associated games.

2 Modal logic and the guarded fragment

The guarded fragment (GF) of first-order logic has been introduced by Andréka, van Benthem and Németi in [1]. It is defined by restricting existential and universal quantification in such a way that formulae can only refer to *guarded tuples*, i.e., tuples of elements that occur together in some atomic fact. Syntactically, this means we consider first-order formulae over some relational vocabulary τ where quantifiers can be used only in the form $\exists \bar{y}(\alpha \wedge \varphi)$ or $\forall \bar{y}(\alpha \rightarrow \varphi)$ where α is an atomic formula that must contain *all* free variables of φ (and possibly more). The atom α is called the guard of the quantification. If φ contains only a single free variable x, then the guard may be the equality $x = x$.

An important motivation for introducing the guarded fragment was to explain and generalize the good algorithmic and model-theoretic properties of *modal logics* (see [6] for background on modal logic). Recall that the basic modal logic ML can be viewed as a fragment of first-order logic, via the standard translation that takes every modal formula $\psi \in \mathrm{ML}$ to a first-order formula $\psi^*(x)$ with one free variable, such that for every Kripke structure $\mathscr{K}$ with a distinguished node w we have that $\mathscr{K}, w \models \psi$ if, and only if, $\mathscr{K} \models \psi^*(w)$. This translation takes an atomic proposition P to the atom Px, it commutes with the Boolean connectives, and it translates the modal operators by quantifiers as follows: For $\psi = \Diamond\varphi$, we have $\psi^*(x) := \exists y(Exy \wedge \varphi^*(y))$ and $\psi = \Box\varphi$ is translated into $\psi^*(x) := \forall y(Exy \to \varphi^*(y))$, with the binary relation symbol E describing the accessibility between different nodes of the Kripke structure. The *modal fragment* of first-order logic is the image of propositional modal logic under this translation. Notice that formulae in the modal fragment can be written using just two variables x and y. The formula $\varphi^*(y)$, used in the translation of the modal operators, is obtained from $\varphi^*(x)$ by exchanging all occurrences of x by y and all occurrences of y by x. Further, 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.

It has turned out that almost all important algorithmic and model-theoretic properties of modal logic extend to the guarded fragment. In particular, the following properties of GF have been demonstrated in [1, 9]:

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: every satisfiable formula of the guarded fragment has a model of small tree width.
4. The notion of equivalence under guarded formulae can be characterized by a straightforward generalization of bisimulation, called guarded bisimulation. See [11] for a detailed discussion of guarded bisimulations in various contexts.

One aspect where, at first sight, modal logic and the guarded fragment seem to differ is the complexity of the satisfiability problem. It is well-known that satisfiablity for ML is a PSPACE-complete problem [15], whereas we have shown in [9] that 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 $p(n)$. But the reason for the double exponential time complexity of GF is essentially just the fact that predicates may have unbounded arity (wheras ML only expresses properties of labelled 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 [9], which is a complexity level that is reached already for rather weak

extensions of ML (e.g. by a universal modality) [16]. Thus, the complexity analysis does not really reveal a fundamental difference between modal and guarded logic, beyond the difference caused by the much wider scope of guarded formulae.

There is a further very important point that Moshe Vardi has called the *robust algorithmic properties of modal logic* [17]. The basic modal logic ML is a rather weak logic and the really interesting modal logics, as far as applications in computer science are concerned, extend ML by features such path quantification, temporal operators, least and greatest fixed points etc. Many of these extended modal logics are algorithmically still rather well manageable and actually of considerable practical importance. The most interesting of these extensions is the modal μ-calculus L_μ, which extends ML by least and greatest fixed points and subsumes most of the modal logics used for automatic verification including CTL, LTL, CTL*, and PDL. The satisfiability problem for L_μ is known to be decidable and complete for EXPTIME [8].

It has turned out that the guarded fragment shares this robustness of modal logic. If we extend GF by similar features as modal logic, in particular by least and greatest fixed points, we still get decidable logics, and in fact we do not even pay a prize in terms of the complexity classes in which we can place the satisfiability problem. Indeed, as we have shown in [14], the satisfiability problem for the guarded fixed point logic μGF 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.

Based on all these results it is indeed fair to say that it is the guarded nature of quantification that is the main reason for the good model-theoretic and algorithmic properties of modal logics. For more details, see [10], which can be seen as an answer to [17].

3 Semiring provenance for first-order logic and acyclic games

We present a brief survey on the use of commutative semirings for provenance analysis, both for first-order logic and for acyclic finite games.

3.1 Commutative semirings

Definition 1. A semiring is an algebraic structure $(K,+,\cdot,0,1)$, with $0 \neq 1$, such that $(K,+,0)$ is a commutative monoid, $(K,\cdot,1)$ is a monoid, $\cdot$ distributes over $+$, and

$0 \cdot a = a \cdot 0 = 0$. The semiring is *commutative* if $\cdot$ is commutative, and it is *idempotent* if $+$ is idempotent. All semirings considered in this paper are commutative.

Elements of a commutative semiring will be used as truth values for logical statements and as values for positions in games. The intuition is that + describes the *alternative use* of information, as in disjunctions or existential quantifications, or for different possible choices of a player in a game, whereas $\cdot$ stands for the *joint use* of information, as in conjunctions or universal quantifications, or for choices in a game that are controlled by the opponent of the given player. Further, 0 is the value of false statements or losing positions, whereas any element $a \neq 0$ of a semiring K stands for a "nuanced" interpretation of true or as a value of a non-losing position.

Examples. Every distributive lattice is an idempotent commutative semiring. Here are some commutative semirings of interest to us:

1. The Boolean semiring $\mathbb{B} = (\mathbb{B}, \vee, \wedge, \bot, \top)$ is the standard habitat of logical truth.
2. $\mathbb{N} = (\mathbb{N}, +, \cdot, 0, 1)$ is of importance for multiset semantics in logic and databases. We also use it here for counting winning strategies in model-checking games.
3. $\mathbb{T} = (\mathbb{R}^{\infty}_{+}, \min, +, \infty, 0)$ is called the *tropical* semiring and is idempotent but not a distributive lattice. It is used for cost-analysis in many areas of computer science. It is important also for analysing the costs of strategies in games.
4. $\mathbb{V} = ([0,1], \max, \cdot, 0, 1)$ is called the *Viterbi* semiring and is isomorphic to $\mathbb{T}$ via $x \mapsto e^{-x}$ and $y \mapsto -\ln y$. We think of the elements of $\mathbb{V}$ as *confidence scores*, for instance for the truth of a given statement, or the confidence of an agent that she can win a game from a given position.
5. For any set X, the semiring $\mathbb{N}[X] = (\mathbb{N}[X], +, \cdot, 0, 1)$ consists of the multivariate polynomials in indeterminates from X and with coefficients from $\mathbb{N}$. This is the commutative semiring that is freely generated by the set X. It is used for a general form of provenance.

3.2 Provenance for first-order logic

Given a finite relational vocabulary τ and a finite non-empty universe A, we denote by $\mathrm{Atoms}_A(\tau)$ the set of all atoms $R\bar{a}$ with $R \in \tau$ and $\bar{a} \in A^k$. Further let $\mathrm{NegAtoms}_A(\tau)$ be the set of all negated atoms $\neg R\bar{a}$ of the facts in $\mathrm{Atoms}_A(\tau)$, and consider the set of all τ-literals on A,

$$\mathrm{Lit}_A(\tau) := \mathrm{Atoms}_A(\tau) \cup \mathrm{NegAtoms}_A(\tau) \cup \{a \ \mathrm{op}\ b : a, b \in A\},$$

where op stands for $=$ or $\neq$.

Definition 3.1 Given any commutative semiring K, a *K-interpretation* (for τ and A) is a function $\pi : \mathrm{Lit}_A(\tau) \to K$ that maps all equality and inequality literals to their truth values 0 or 1.

As defined in [12] a semiring interpretation extends to a full valuation $\pi : \mathrm{FO}(\tau) \to K$ mapping any fully instantiated formula $\psi(\overline{a})$ (or equivalently, any first-order sentence of vocabulary $\tau \cup A$), to a value $\pi[\![\varphi]\!]$, by setting

$$\pi[\![\psi \vee \varphi]\!] := \pi[\![\psi]\!] + \pi[\![\varphi)]\!] \qquad \pi[\![\psi \wedge \varphi]\!] := \pi[\![\psi]\!] \cdot \pi[\![\varphi]\!]$$
$$\pi[\![\exists x \varphi(x)]\!] := \sum_{a \in A} \pi[\![\varphi(a)]\!] \qquad \pi[\![\forall x \varphi(x)]\!] := \prod_{a \in A} \pi[\![\varphi(a)]\!].$$

Negation is handled via negation normal forms: we set $\pi[\![\neg\varphi]\!] := \pi[\![\mathrm{nnf}(\neg\varphi)]\!]$ where $\mathrm{nnf}(\varphi)$ is the negation normal form of φ.

Definition 3.2 A semiring interpretation $\pi : \mathrm{Lit}_A(\tau) \to K$ is *model-defining* if for every atom $\varphi \in \mathrm{Atoms}_A(\tau)$ one of $\pi(\varphi)$ and $\pi(\neg\varphi)$ is 0, and the other is $\neq 0$. It uniquely defines the τ-structure $\mathfrak{A}_\pi$ that has universe A, and in which precisely those literals φ are true for which $\pi(\varphi) \neq 0$.

This definition is motivated by the interpretation described above, that a provenance value of 0 describes a false statement, whereas a non-zero value indicates some nuance of truth. Notice that, if K is not the Boolean semiring, then several different K interpretations may define the same structure. Further, K-interpretations are interesting, and have a number of applications, also in cases where they do not specify a single model, see [12] and the references given there.

Such valuations of first-order logic in a semiring K do in fact have an equivalent definition in terms of K-valuations of the usual model-checking games for first-order formulae. We next discuss the provenance approach to games, focussing for simplicity just on the case of acyclic games.

3.3 Provenance analysis for acyclic games

In this section we briefly describe the provenance approach to games as developed in [13], restricting attention to two-player turn-based games on acyclic directed graphs. Such a game is defined by the game graph on which it is played, and by the objectives of the players.

Definition 3.3 A *game graph* is a structure $\mathscr{G} = (V, V_0, V_1, T, E)$, where $V = V_0 \cup V_1 \cup T$ is the set of positions, partitioned into the sets V_0, V_1 of the two players and the set T of terminal positions, and where $E \subseteq V \times V$ is the set of moves. In the games considered here, the underlying graph $G = (V, E)$ is always acyclic and finite. We denote the set of immediate successors of a position v by $vE := \{w : (v,w) \in E\}$ and require that $vE = \emptyset$ if, and only if, $v \in T$. A play from an initial position v_0 is a path $v_0 v_1 v_2 \ldots v_m$ through $\mathscr{G}$ where the successor $v_{i+1} \in v_i E$ is chosen by Player 0 if $v_i \in V_0$ and by Player 1 if $v_i \in V_1$. A play ends when it reaches a terminal node $v_m \in T$.

A strategy for a player in a game is a function that selects moves at points that are controlled by that player. A strategy need not be defined at all positions of a player, but it must be closed in the sense that it defines a move from each position that is reachable by a play that is admitted by the strategy. There are several possibilities to define the notion of a strategy formally. For our purposes it is convenient to identify a strategy with the histories of plays that it admits.

Definition 3.4 For every game graph $\mathscr{G} = (V, V_0, V_1, T, E)$, and every initial position $v_0 \in V$, the *tree unraveling* of $\mathscr{G}$ from v_0 is the game tree $\mathscr{T}(\mathscr{G}, v_0)$ of all finite paths from v_0. More precisely, $\mathscr{T}(\mathscr{G}, v_0) = (V^\#, V_0^\#, V_1^\#, T^\#, E^\#)$, where $V^\#$ is the set of finite paths $\pi = v_0 v_1 \dots v_m$ from v_0 through $\mathscr{G}$, with $V_\sigma^\# = \{\pi v \in V^\# : v \in V_\sigma\}$, $T^\# = \{\pi t \in V^\# : t \in T\}$, and $E^\# = \{(\pi v, \pi v v') : (v, v') \in E\}$. For most game-theoretic considerations, the games played on $\mathscr{G}$ and its unravelings are equivalent, via the canonical projection $\rho : \mathscr{T}(\mathscr{G}, v_0) \to \mathscr{G}$ that maps every path πv to its end point v.

The elements of $\mathscr{T}(\mathscr{G}, v_0)$ are the finite initial segments or histories of all possible plays of $\mathscr{G}$ that start at v_0. A strategy of a player can now be viewed as an appropriate subtree of $\mathscr{T}(\mathscr{G}, v_0)$.

Definition 3.5 A *strategy* of Player σ from v_0 in a game $\mathscr{G}$ is a subtree of $\mathscr{T}(\mathscr{G}, v_0)$, of the form $S = (W, F)$ with $W \subseteq V^\#$ and $F \subseteq (W \times W) \cap E^\#$, satisfying the following conditions:

(1) W is closed under predecessors: if $\pi v \in W$ then also $\pi \in W$.
(2) If $\pi v \in W \cap V_\sigma^\#$, then $|(\pi v)F| = 1$.
(3) If $\pi v \in W \cap V_{1-\sigma}^\#$ then $(\pi v)F = (\pi v)E^\#$.

A strategy can also be viewed as a function $S : W \cap V_\sigma^\# \to V$ such that $S(\pi v) \in vE$ defines the node to which Player σ moves from πv.

Here W is the part of $\mathscr{T}(\mathscr{G}, v_0)$ on which the strategy is defined, and F is the set of moves that are admitted by the strategy. We define $\mathrm{Strat}_\sigma(v_0)$ to denote the set of all strategies of Player σ from v_0. A strategy $S \in \mathrm{Strat}_\sigma(v_0)$ induces the set $\mathrm{Plays}(S)$ of those plays from v_0 whose moves are consistent with S. We call S well-founded if it does not admit any infinite plays; this is always the case on finite acyclic game graphs, but need not be the case otherwise. The set of possible *outcomes* of a strategy S is the set of terminal nodes that are reachable by a play that is consistent with S.

Game valuations. Let $(K, +, \cdot, 0, 1)$ be a commutative semiring, and let $\mathscr{G} = (V, V_0, V_1, T, E)$ be a finite acyclic game graph. A K-valuation of $\mathscr{G}$ for Player σ provides a value $f_\sigma(v) \in K$ for every position $v \in V$.

Such a valuation is induced by its values on the terminal positions, i.e. by a function $f_\sigma : T \to K$, and by a valuation of the moves, i.e. by a function $h_\sigma : E \to (K \setminus \{0\})$. The function $f_\sigma : T \to K$ defines the value of every terminal position from the point of view of Player σ. Intuitively, $f_\sigma(t) = 0$ means that position t is losing for Player σ. For instance, we can specify reachability objectives T_σ by setting $f_\sigma(t) = 1$ for $t \in T_\sigma$ and $f_\sigma(t) = 0$ otherwise. But there are many other choices.

The functions $h_\sigma : E \to (K \setminus \{0\})$ provide a value (or cost) for Player σ of the moves. In many cases valuations of moves are not relevant; we then just put $h_\sigma(vw) = 1$ for all edges $(v,w) \in E$. When the functions for the two players are identical, i.e. $h_0 = h_1$, we often omit the subscripts.

The extension of the basic valuations $f_\sigma : T \to K$ and $h_\sigma : E \to K \setminus \{0\}$ to valuations $f_\sigma : V \to K$ for all positions relies on the idea that a move from v to w contributes to $f_\sigma(v)$ the value $h_\sigma(vw) \cdot f_\sigma(w)$. These contributions are summed up in the case that v is a position for Player σ (i.e. when she choses herself the successors), and multiplied in the case that v is a position of the opponent (i.e. when she has to cope with any of the possible successors). Thus

$$f_\sigma(v) := \begin{cases} \sum_{w \in vE} h_\sigma(vw) \cdot f_\sigma(w) & \text{if } v \in V_\sigma \\ \prod_{w \in vE} h_\sigma(vw) \cdot f_\sigma(w) & \text{if } v \in V_{1-\sigma}. \end{cases}$$

An equivalent characterization of the provenance values $f_\sigma(v)$ can be obtained by defining provenance values for plays and strategies.

Definition 3.6 For a play $x = v_0 v_1 \dots v_m$ from v_0 to a terminal node v_m, we define its valuation for Player σ as $f_\sigma(x) := h_\sigma(v_0v_1) \cdots h_\sigma(v_{m-1}v_m) \cdot f_\sigma(v_m)$. Let now $S = (W,F) \subseteq \mathscr{T}(\mathscr{G}, v_0)$ be a strategy for Player σ from v_0 and $\rho_S : (W,F) \to (V,E)$ be the restriction of of the canonical homomorphism $\rho : \mathscr{T}(\mathscr{G}, v_0) \to \mathscr{G}$ to S. For any position $v \in V$ and any move $e \in E$, the values

$$\#_S(v) := |\rho_S^{-1}(v)| \quad \text{and} \quad \#_S(e) := |\rho_S^{-1}(e)|$$

indicate how often the position v and the move e appear in the strategy S. We then define the provenance value $S \in \mathrm{Strat}_\sigma(v_0)$ as

$$F(S) := \prod_{e \in E} h_\sigma(e)^{\#_S(e)} \cdot \prod_{v \in T} f_\sigma(v)^{\#_S(v)}.$$

Theorem 3.7 *For any commutative semiring K and any finite acyclic game $\mathscr{G}$, let $f_\sigma : V \to K$ be the provenance valuation for Player σ, induced by the valuation $f_\sigma : T \to K$ of the terminal nodes and $h_\sigma : E \to K \setminus \{0\}$ of the moves. Then, for every position v*

$$f_\sigma(v) = \sum_{S \in \mathrm{Strat}_\sigma(v)} F(S).$$

If $h_\sigma(e) = 1$ for all moves $e \in E$, or if the underlying semiring is multiplicatively idempotent (i.e. $a^2 = a$ for all a), then we further have that

$$f_\sigma(v) = \sum_{S \in \mathrm{Strat}_\sigma(v)} \prod_{x \in \mathrm{Plays}(S)} f_\sigma(x).$$

Example: Cost of strategies. Given a game $\mathscr{G}$, we associate for Player 0 *cost functions* $f_0 : T \to \mathbb{R}_+$ and $h : E \to \mathbb{R}_+$ for the terminal positions and the moves. We

define the cost of a strategy $S \in \mathrm{Strat}_0(v)$ as the sum of the costs of all moves and outcomes that it admits, weighted by the number of their occurrences.

Proposition 3.8 *The* cost of an optimal strategy *from v in $\mathscr{G}$ is given by the valuation $f_0(v)$ in the tropical semiring $\mathbb{T} = (\mathbb{R}^\infty_+, \min, +, \infty, 0)$.*

Similarly, we can use game valuations in appropriate semirings for computing *confidence scores* for positions (decribing the confindence of a player to win from that position) or *minimal clearance levels* that a player needs to win from a position, assuming that the possible moves have access restrictions ("confidential", "secret","top secret"). For details, see [13].

Counting winning strategies. Consider a game graph $\mathscr{G} = (V, V_0, V_1, T, E)$ with a set T of terminal positions and with trivial valuations for the moves, i.e. $h_\sigma(vw) = 1$ for all edges (v,w). A general provenance analysis for acyclic games is based in the semiring $\mathbb{N}[T]$ of polynomials over the indeterminates $t \in T$, the semiring that is freely generated by the set of terminal positions.

We define $f_\sigma : V \to \mathbb{N}[T]$ as the valuations induced by setting $f_\sigma(t) = t$ for $t \in T$. Clearly, we can write $f_\sigma(v)$ as a sum of monomials $m \cdot t_1^{j_1} \dots t_k^{j_k}$. This provides a detailed description of the number and properties of the strategies that Player σ has from position v.

Theorem 3.9 *Every monomial $m \cdot t_1^{j_1} \dots t_k^{j_k}$ in $f_\sigma(v)$ (with $m \in \mathbb{N}$ and $j_i > 0$) indicates that Player σ has precisely m strategies S from v with the property that the set of possible outcomes for S is precisely $\{t_1, \dots, t_k\}$, and precisely j_i plays that are consistent with S have the outcome t_i.*

If we fix any reachability objective $W \subseteq T$ for Player σ, we can write the polynomial $f_\sigma(v)$ as a sum $f_\sigma(v) = f^W_\sigma(v) + g^W_\sigma(v)$ where $f^W_\sigma(v)$ is the sum of those monomials that only contain indeterminates in W (i.e. for which $j(t) = 0$ whenever $t \in T \setminus W$), and $g^W_\sigma(v)$ contains the rest.

Theorem 3.10 *For every subset $W \subseteq T$ and every $v \in V$, Player σ has a strategy to reach W from v if, and only if, $f^W_\sigma(v) \neq 0$. Moreover, if*

$$f^W_\sigma(v) = \sum_{j \in J} m_j \prod_{t \in W} t^{j(t)}$$

then $\sum_{j \in J} c_j$ is the number of distinct deterministic strategies from v that Player σ has for this objective.

3.4 Provenance analysis via model-checking games

Let $\mathfrak{A}$ be a finite relational τ-structure and ψ be a first-order formula in negation normal form. The model checking game $\mathscr{G}(\mathfrak{A}, \psi)$ is defined in the usual way. The

positions are expressions $\varphi(\bar{a})$, obtained from a subformula $\varphi(\bar{x})$ of ψ, by instantiating the free variables $\bar{x}$ by a tuple $\bar{a}$ of elements of $\mathfrak{A}$. At a disjunction $(\psi \vee \varphi)$, Player 0 (Verifier) moves to either ψ or φ, and at a conjunction, Player 1 (Falsifier) makes an analogous move. At a position $\exists x\varphi(\bar{a},x)$, Verifier selects an element b and moves to $\varphi(\bar{a},b)$, whereas at positions $\forall x\varphi(\bar{a},x)$ the move to to the next position $\varphi(\bar{a},b)$ is done by Falsifier. The terminal positions of $\mathscr{G}(\mathfrak{A},\psi)$ are the literals in $\mathrm{Lit}_A(\tau)$. Literals $\varphi \in \mathrm{Lit}_A(\tau)$ that are true in the given structure $\mathfrak{A}$ are the winning terminal positions for Verifier in $\mathscr{G}(\mathfrak{A},\psi)$ (and the losing ones for Falsifier); for the literals that are false in $\mathfrak{A}$ it is the other way round.

The central observation concerning these games is that, for any structure $\mathfrak{A}$ and any position φ of a model checking game $\mathscr{G}(\mathfrak{A},\psi)$, Verifier has a winning strategy from φ if, and only if, $\mathfrak{A} \models \varphi$. Moreover, by duality, or by the determinacy of well-founded games, Falsifier has a winning strategy from φ if, and only if, $\mathfrak{A} \not\models \varphi$ which, of course, is the case if, and only if, $\mathfrak{A} \models \neg\varphi$.

Provenance analysis provides a broader view on both logic and games. Notice that up to the labelling of the terminal positions as winning or losing for Verifier (Player 0) and Falsifier (Player 1), the model checking game $\mathscr{G}(\mathfrak{A},\psi)$ only depends on the formula ψ and on the *universe* A of the structure. Thus, we have a game graph $\mathscr{G}(A,\psi)$, and separately a valuation $\pi : \mathrm{Lit}_A(\tau) \to \{0,1\}$.

From Boolean valuations of literals (and of terminal positions of games) we can now move to K-valuations for an arbitrary commutative semiring K, and study the connection between logic and games in this broader context.

Let $\pi : \mathrm{Lit}_A(\tau) \to K$ be any K-interpretation of the τ-literals on A in a semiring K. We can view π as a K-valuation $f_0 : T \to K$ of the set of terminal positions of any model-checking game $\mathscr{G}(\mathfrak{A},\psi)$ (for a τ structure with universe A and a first-order formula ψ) from the point of view of Player 0. The dual valuation $f_1 : T \to K$ for Player 1 is obtained by putting $f_1(\varphi) = \pi[\![\varphi^\neg]\!]$ where $\varphi^\neg \equiv \neg\varphi$ is the complementary literal to φ. Then both f_0 and f_1 extend to valuations $f_0 : V \to K$ and $f_1 : V \to K$ of all positions of $\mathscr{G}(\mathfrak{A},\psi)$. In particular, we obtain valuations $f_0(\psi)$ and $f_1(\psi)$ for the initial position ψ.

Proposition 3.11 *Suppose that π is model-defining, and hence completely specifies a structure $\mathfrak{A}_\pi$. For every first-order formula ψ and every position $\varphi(\bar{a})$ in $\mathscr{G}(\mathfrak{A},\psi)$ we have that $\pi[\![\varphi(\bar{a})]\!] = f_0(\varphi(\bar{a}))$ and $\pi[\![\neg\varphi(\bar{a})]\!] = f_1(\varphi(\bar{a}))$. In particular $\mathfrak{A}_\pi \models \psi$ if, and only if, $f_0(\psi) \neq 0$.*

4 Provenance analysis for modal logic and the guarded fragment

Recall that modal logic, for a fixed vocabulary $\{P_i : i \in I\}$ of atomic propositions, is given by the grammar

$$\varphi ::= \bot \mid \top \mid P_i \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \neg\varphi \mid \Diamond\varphi \mid \Box\varphi.$$

A transition system (or Kripke structure) for this vocabulary is a labelled directed graph $\mathscr{K} = (V, E, (P_i)_{i\in I})$ with $E \subseteq V \times V$ and $P_i \subseteq V$, and we write $\mathscr{K}, v \models \varphi$ if φ holds at state v in the transition system $\mathscr{K}$. The set of modal literals for V and τ, denoted MLit_V consists of the atoms $P_i v$ and their negations $\neg P_i v$ (for $v \in V$, $i \in I$), and the edge atoms Evw for $v, w \in V$. Note that literals $\neg Evw$ for the absence of edges are not included.

Definition 4.1 Let K be a semiring. A *modal K-interpretation* for V is a function $\pi : \mathrm{MLit}_V \to K$.

Similar to the case of first-order logic, a modal K-interpretation extends to a K-valuation $\pi : \mathrm{ML} \times V \to K$ by

$$\pi[\![\bot, v]\!] := 0 \qquad \pi[\![\top, v]\!] := 1$$
$$\pi[\![P_i, v]\!] := \pi(P_i v) \qquad \pi[\![\neg P_i, v]\!] := \pi(\neg P_i v)$$
$$\pi[\![\psi \vee \varphi, v]\!] := \pi[\![\psi, v]\!] + \pi[\![\varphi, v]\!] \qquad \pi[\![\psi \wedge \varphi, v]\!] := \pi[\![\psi, v]\!] \cdot \pi[\![\varphi, v]\!]$$
$$\pi[\![\Diamond\varphi, v]\!] := \sum_{w \in vE} \pi(Evw) \cdot \pi[\![\varphi, w]\!] \qquad \pi[\![\Box\varphi, v]\!] := \prod_{w \in vE} \pi(Evw) \cdot \pi[\![\varphi, w]\!]$$
$$\pi[\![\neg\varphi, v]\!] := \pi[\![\mathrm{nnf}(\neg\varphi), v]\!].$$

This valuation is usually not the same as the first-order valuation for the translation of modal logic formulae into first-order logic. Notice, that the standard translation of modal logic into first-order logic maps a formula $\psi = \Box\varphi \in \mathrm{ML}$ to the first-order formula $\psi^*(x) := \forall y(Exy \to \varphi^*(y))$. Rewriting $\psi^*(x)$ as $\forall y(\neg Exy \vee \varphi^*(y))$ we see that a K-interpretation of $\psi^*(v) \in \mathrm{FO}$ requires a basic K-interpretation of $\mathrm{Lit}_V(\tau)$ for $\tau = \{P_i : i \in I\} \cup \{E\}$, which needs to provide values also for the negative literals $\neg Evw$. Even if we extend the given modal K-interpretation $\pi : \mathrm{MLit}_V \to K$ in the simplest possible way, by setting $\pi(\neg Evw) = 1$ if $\pi(Evw) = 0$ and $\pi(\neg Evw) = 0$ otherwise, the resulting valuation will have the property that

$$\begin{aligned}\pi[\![\psi^*(v)]\!] = \pi[\![\forall y(\neg Evy \vee \varphi^*(y))]\!] &= \prod_{w \in V} (\pi(\neg Evw) + \pi[\![\varphi^*(w)]\!]) \\ &= \prod_{w \in vE} \pi[\![\varphi^*(w)]\!] \cdot \prod_{w \notin vE} (1 + \pi[\![\varphi^*(w)]\!])\end{aligned}$$

which is, in general something quite different than the value $\pi[\![\psi, v]\!]$ of the corresponding modal K-interpretation. A sufficient condition that the two valuations coincide would be that the basic modal valuation maps edge atoms only to 0 and 1, and that in the given semiring $1 \in K$ is an absorbing element, i.e. $1 + a = 1$ for all a.

A justification for our proposed valuation, despite the difference to first-order logic, is that it is more in line with the intuitive meaning of the modal logic formula $\Box\varphi$. When we think of the meaning of $\Box\varphi$, we think of it as "φ holds at all successors of the current node", which corresponds to the provenance valuation as defined above. We do not usally interpret $\Box\varphi$ as the statement "for all nodes w in the Kripke-structure, either there is no edge to w from the current node or φ holds

at w". This interpretation corresponds to the first-order translation of $\Box\varphi$ but it goes against the local nature of modal logic and is therefore far less intuitive.

Additionally, our proposed provenance valuation for modal logic is completely in line with K-valuations for the standard model-checking games for modal logic. Indeed, the model-checking game $\mathscr{G}(\mathscr{K},\psi)$ for a transition system $\mathscr{K}$ with frame (V,E) and a modal formula ψ has positions (φ,v) where φ is a subformula of ψ and v is a state of $\mathscr{K}$. From positions $(\varphi_1\vee\varphi_2,v)$, Player 0 can move to (φ_1,v) or (φ_2,v) and dually Player 1 moves from $(\varphi_1\wedge\varphi_2,v)$ to either (φ_1,v) or (φ_2,v). At positions $(\Diamond\varphi,v)$ Player 0 can move to any position (φ,w) such that $(v,w)\in E$, and there are analogous moves for Player 1 at positions $(\Box\varphi,v)$.

If we define the basic valuations for Player 0 of the terminal positions by $f_0(P_i,v)=\pi(P_iv)$, $f_0(\neg P_i,v)=\pi(\neg P_iv)$, $f_0(\bot,v)=0$ and $f_0(\top,v)=1$, and give any move from a position $(\Diamond\varphi,v)$ or $(\Box\varphi,v)$ to (φ,w) the value $\pi(Evw)$, then we obtain, for any formula $\psi\in\mathrm{ML}$ and every $v\in V$ a valuation $f_0(\psi,v)\in K$ such that $f_0(\psi,v)=\pi[\![\psi,v]\!]$.

We now move to a provenance analysis for the guarded fragment GF. We start with a more explicit definition of GF.

Definition 4.2 Given a relational vocabulary τ, the set of guarded formulae in $\mathrm{GF}(\tau)$ is defined inductively by the following rules:

(1) Every atomic τ-formula belongs to $\mathrm{GF}(\tau)$;
(2) $\mathrm{GF}(\tau)$ is closed under $\wedge$, $\vee$, and $\neg$;
(3) $\mathrm{GF}(\tau)$ is closed under guarded quantification: For every formula $\varphi\in\mathrm{GF}(\tau)$, every τ-atom α such that *all* free variables of φ occur in α, and every tuple $\bar{y}$ of variables occurring in α, $\mathrm{GF}(\tau)$ contains formulae $(\exists\bar{y}.\,\alpha)\varphi$ and $(\forall\bar{y}.\,\alpha)\varphi$.

Here (3) is the rule of *guarded quantification* and the atom α is called the *guard* of the quantification. The semantics of guarded quantification is defined as follows. Let $\mathrm{var}(\alpha)$ be set of all variables occurring in the atom α, let $\mathfrak{A}$ be a τ-structure and let s be a valuation, mapping the free variables of $(Q\bar{y}.\,\alpha)\varphi$ into A. Then we denote by $T_{s,\alpha}$ the set of all valuations $t:\mathrm{var}(\alpha)\to A$ such that $\mathfrak{A}\models_t\alpha$ and t coincides with s on the common variables. Then we put

$$\mathfrak{A}\models_s(\exists\bar{y}.\,\alpha)\varphi \iff \mathfrak{A}\models_t\varphi \quad \text{for } \textit{some } t\in T_{s,\alpha}$$
$$\mathfrak{A}\models_s(\forall\bar{y}.\,\alpha)\varphi \iff \mathfrak{A}\models_t\varphi \quad \text{for } \textit{all } t\in T_{s,\alpha}$$

Notice that in a formula $\psi:=(Q\bar{y}.\,\alpha)\varphi$, the only requirement is that α contains all free variables of φ. But it may contain more variables, and contrary to unguarded quantification in FO, among the free variables of ψ (i.e. the variables occurring in α but not in $\bar{y}$) there may be some that do not occur in φ.

To see what happens precisely in an evaluation of a guarded formula, say in a model-checking game or when we define K-interpretations in a semiring K, it is therefore instructive to rewrite the rule of guarded quantification in a way that makes all free variables explicit. In the following, $\bar{x}$, $\bar{x}'$, $\bar{y}$, and $\bar{z}$ are disjoint (and possibly

empty) sequences of variables, and the free variables of the formulae are precisely as displayed.

(3) For every formula $\varphi(\overline{xy}) \in \mathrm{GF}(\tau)$ and every atomic τ-formula $\alpha(\overline{xx'yz})$, we can build in $\mathrm{GF}(\tau)$ the formulae

$$\psi(\overline{xx'}) := (\exists \overline{yz}\,.\,\alpha(\overline{xx'yz}))\varphi(\overline{xy}) \text{ and } \psi(\overline{xx'}) := (\forall \overline{yz}\,.\,\alpha(\overline{xx'yz}))\varphi(\overline{xy}).$$

Thus, the natural model-checking games for GF-formulae are modifications of the first-order model checking games where, for a formula $\psi(\overline{xx'}) := (Q\overline{yz}\,.\,\alpha)\varphi(\overline{xy})$, we have a move from $\psi(\overline{aa'})$ to $\varphi(\overline{ab})$ for every tuple $\overline{c}$ such that $\alpha(\overline{aa'bc})$ holds. Notice that this means that there may actually be more than one move from $\psi(\overline{aa'})$ to $\varphi(\overline{ab})$.

Let $\pi : \mathrm{Lit}_A(\tau) \to K$ be a K-interpretation for A and τ. It provides, for every terminal position φ of a GF-model-checking game on A, the basic valuation $f_0(\varphi) = \pi(\varphi)$. The valuations of the moves are defined as follows. Every move associated with a disjunction or conjunction, going from $(\varphi_1 \vee \varphi_2)$ or $(\varphi_1 \wedge \varphi_2)$ to φ_1 or φ_2 has value 1. Every move associated with a guarded quantification, that is from $\psi(\overline{aa'}) := (Q\overline{yz}\,.\,\alpha)\varphi(\overline{a},\overline{y})$ to $\varphi(\overline{ab})$, which is witnessed by the atom $\alpha(\overline{aa'bc})$ has value $\pi(\alpha(\overline{aa'bc}))$.

This induces a valuation $f_0(\varphi) \in K$ for every position φ in the game. This coincides with the extension of $\pi : \mathrm{Lit}_A(\tau) \to K$ to $\pi : \mathrm{GF}(\tau) \to K$ by the straightforward induction, setting

$$\pi[\![\exists \overline{yz}\,\alpha(\overline{aa'},\overline{y},\overline{z})\varphi(\overline{a},\overline{y})]\!] := \sum_{\overline{bc}\colon\, \mathfrak{A}\models\alpha(\overline{aa'bc})} \pi(\alpha(\overline{aa'bc})) \cdot \pi[\![\varphi(\overline{ab})]\!]$$

$$\pi[\![\forall \overline{yz}\,\alpha(\overline{aa'yz})\varphi(\overline{a},\overline{y})]\!] := \prod_{\overline{bc}\colon\, \mathfrak{A}\models\alpha(\overline{aa'bc})} \pi(\alpha(\overline{aa'bc})) \cdot \pi[\![\varphi(\overline{ab})]\!]$$

Again negation is handled via negation normal form, where $\mathrm{nnf}(\neg(\exists \overline{y}\,.\,\alpha)\varphi) = (\forall \overline{y}\,.\,\alpha)\,\mathrm{nnf}(\neg\varphi)$.

As in the case of modal logic, the standard translation of guarded universal quantification into usual first-order syntax taking $(\forall \overline{y}\,.\,\alpha)\varphi$ to $\forall \overline{y}(\neg\alpha \vee \varphi)$ produces in general formulae that have not the same K-valuations.

5 Algorithmic Analysis

It is well known that the model checking problems for both modal logic and the guarded fragment can be solved in polynomial time, whereas the corresponding for full first-order logic is PSPACE-complete. One way to prove this, and to understand the differences between modal and guarded logic on one side, and full FO on the other side, is to compute the size of the model-checking games. An arbitrary first-order sentence ψ on a finite structure $\mathfrak{A}$ has a model checking game $\mathscr{G}(\mathfrak{A},\psi)$ of

size $O(|\psi| \cdot |A|^{\text{width}(\psi)})$, where $|A|$ is the number of elements of $\mathfrak{A}$ and $\text{width}(\psi)$ is the maximal number of free variables in subformulae of ψ. However, for a formula $\psi \in \text{GF}$ the size of $\mathscr{G}(\mathfrak{A}, \psi)$ is only $O(\psi \cdot \|\mathfrak{A}\|)$ where $\|\mathfrak{A}\|$ is the length of a natural representation of $\mathfrak{A}$ (listing all atomic facts). A similar bound applies for the size of model checking games for modal logic.

This poses the natural question whether there is a similar difference in the complexities of computing provenance values $\pi[\![\psi]\!]$, given a formula ψ from ML, GF, or FO, and a K-interpretation $\pi : \text{Lit}_A(\tau) \to K$, for some fixed semiring K.

Of course, this question may strongly depend on the semiring K that we consider, and how we measure the complexity of addition and multiplication in K. In the Boolean semiring we simply count the number of operations needed to compute the truth value of a given formula. We can take an analogous approach in an arbitrary semiring and just count the number of arithmetic operations needed to compute a provenance value. This would mean that we abstract from the computational difficulties that arise from representing semiring elements as words over some fixed alphabet and computing sums and products on such representations. This *unit cost model* is certainly appropriate for finite semirings, but gives relevant insights also in other cases, specifically for an abstract approach over, say, uncountable semirings. In the unit cost model, the total cost $|\pi|$ of a semiring interpretation $\pi : \text{Lit}_A(\tau) \to K$ is just the number of literals in $\text{Lit}_A(\tau)$.

It is easy to see that provenance values for positions in an acyclic game $\mathscr{G}$ can be computed with $O(\|\mathscr{G}\|)$ semiring operations (since every edge of the game graph needs to be processed only once).

Proposition 5.1 *Let K be an arbitrary semiring. Given a formula $\psi \in \text{ML}$ or $\psi \in \text{GF}$ and a corresponding K-interpretation $\pi : \text{Lit}_A(\tau) \to K$, the provenance value $\pi[\![\psi]\!]$ can be computed with $O(|\psi| \cdot |\pi|)$ semiring operations.*

It is very unlikely that this is also holds for arbitrary first-order formulae since, by taking the Boolean semiring, and the PSPACE-completeness of first-order model checking, this would imply that P = PSPACE.

Of course, the unit cost model is unrealistic for many algorithmic applications, in particular if we are interested in practical complexity considerations for computing provenance values in $\mathbb{N}$, or in the semiring of polynomials $\mathbb{N}[X]$. We therefore aim at a more general approach, assuming that we have a semiring K together with cost functions $|\ | : K \to \mathbb{N}$ for the elements and $|\ |_+ : K \times K \to \mathbb{N}$ and $|\ |_\cdot : K \times K \to \mathbb{N}$ for the two semiring operations. We always assume that $|a+b| \leq |(a,b)|_+$, and $|a \cdot b| \leq |(a,b)|_\cdot$ for all $a, b \in K$. In fact, over any semiring, one possibility of cost functions for $+$ and $\cdot$ is to simply take the element costs of the sum and the product, respectively.

In the case of $K = \mathbb{N}$, one approach is to set $|a| = a$, and then $|(a,b)|_+ = a+b$ and $|(a,b)|_\cdot = ab$, which are the cost functions for the *unary representation* of natural numbers. Another, in most cases more natural possibility is the *logarithmic cost model*, with $|a| = \lceil \log a \rceil$, and the cost of addition and multiplication as $|(a,b)|_+ := \max\{|a|, |b|\} + 1$ and $|(a,b)|_\cdot := |a| + |b|$. Here $|(a,b)|_+$ and $|(a,b)|_\cdot$ are

upper bounds for $|a+b|$ and $|a \cdot b|$. For the semiring $\mathbb{N}[X]$ of polynomials over X we can define the cost of a monomial as $|a\prod_{x\in X} x^{e_x}| := \lceil \log a \rceil + \sum_{x\in X} \lceil \log e_x \rceil$ and the cost of a polynomial as the sum of the costs of its monomials or, alternatively, set the cost of each monomial to one and only consider the number of distinct monomials in a polynomial. In general there are several natural cost functions for a semiring, and the unit cost models (setting all costs to 1) is one of them.

We call a cost function $|\ | : K \to \mathbb{N}$ *additively bounded* if $|a+b|, |a\cdot b| = O(|a|+|b|)$, and *multiplicatively bounded* if $|a+b|, |a\cdot b| = O(|a|\cdot|b|)$. Clearly, the logarithmic cost model on $\mathbb{N}$ is additively bounded whereas the costs of unary representations are multiplicatively bounded.

Given a semiring K with associated cost functions, one important complexity parameter for a semiring interpretation $\pi : \mathrm{Lit}_A(\tau) \to K$ is the maximal cost of its values, i.e.

$$\max \pi := \max\{|\pi(\alpha)| : \alpha \in \mathrm{Lit}_A(\tau)\}.$$

Consider any first-order formula $\psi \in \mathrm{FO}(\tau)$ and let $d(\psi)$ be the nesting depth of the logical operators in ψ or, equivalently, the maximal length of plays in model checking games for ψ. We can calculate upper bounds for the costs of provenance values $\pi[\![\psi]\!]$ by looking, again, at the associated model checking game. At the terminal nodes, the costs are bounded by $\max \pi$. Each non-terminal node has at most $|A|$ immediate successors so we perform a sum or product of at most $|A|$ values that may appear at lower levels.

Proposition 5.2 *Let K be an arbitrary semiring with an associated additively bounded cost function, and let $\pi : \mathrm{Lit}_A(\tau) \to K$ be a semiring interpretation with $m = \max \pi$ and $n = |A|$. For any first-order formula ψ of depth $d = d(\psi)$, we then have that*

$$|\pi[\![\psi]\!]| \leq m \cdot n^d.$$

In the case that the cost function of K is multiplicatively bounded, we instead have that

$$|\pi[\![\psi]\!]| \leq m^{n^d}.$$

We claim that the maximal size bounds of Proposition 5.2 can actually be realized, and in fact even by formulae of modal logic and hence also by guarded formulae in GF. For proper comparison, a modal or guarded quantification should count as an operation of depth two, since provenance values for $(\Diamond\varphi, v)$ or $(\Box\varphi, v)$ take into account values of edges (v,w) given by a modal K-interpretation; similarly at a guarded quantification the values of the guard atoms are used in the computation of the provenance values.

Let π be a modal K-interpretation on a completely connected frame with n nodes giving to all P-atoms and all edges the same value a, i.e. $\pi(Pi) = \pi(Eij) = a$ for all $i, j < n$. It follows that

$$\pi[\![(\Box^k P, i)]\!] = a^{n^{2k}}.$$

Taking, for instance, $K = \mathbb{N}$ with an additively bounded cost measure, putting $m = |a| = \max \pi$ and $d = 2k$ we indeed get $|\pi[\![(\Box^k P, i)]\!]| = m \cdot n^d$, and for a multiplica-

tively bounded cost measure (such as unary representation), we get $|\pi[\![(\Box^k P,i)]\!]| = m^{n^d}$. This exponential cost is not unique to $\mathbb{N}$ of course. Consider the polynomial semiring $K = \mathbb{N}[x,y]$ and very simple modal K-interpretation on a universe with just two nodes u,v, such that all four possible edges have value 1, and further $\pi(Pu) = \pi(Pv) = x$ and $\pi(Pu) = \pi(Pv) = y$. Then, on both nodes, the formula $\Box^k(P \vee Q)$ has provenance value $(x+y)^{2^k}$ which has cost 2^k+1 even if we set monomial costs to 1.

This shows us that there is no essential difference between modal, guarded, and arbitrary first-order formulae ψ with respect to the maximal possible (cost of) provenance values $\pi[\![\psi]\!]$, for arbitrary K-interpretations over the same universes and with the same bounds for literals.

Nevertheless there can be huge differences for particular K-interpretations π of the maximal provenance values $\pi[\![\psi]\!]$ of modal, guarded, and first-order formulae. Intuitively this arises in cases where the K-interpretations provide very few connections between different elements, which makes the power of modal and guarded quantification very weak, but does not affect arbitrary first-order quantification.

Proposition 5.3 *There are model-defining K-interpretations π such that the model defined by π is a Kripke structure, with the property that the provenance values $\pi[\![\psi]\!]$ of certain guarded first-order formulae are arbitrarily larger that the maximal provenance values of modal formulae φ of the same depth. Similarly, there are K-interpretations with arbitrarily large differences between the provenance values of certain first-order formulae and maximal provenance values of guarded formulae of the same depth.*

Proof. Let $K = \mathbb{N}$ and consider modal K-interpretations π on a (large) universe V that gives value 0 to all potential edges Euv and to all positive atoms Pv, and a value $m \geq 2$ to all negated atoms $\neg Pv$. The maximal provenance values for modal formulae of depth d is m^{2^d} (independent of V); this is achieved by formulae of form $\neg P \wedge \neg P \wedge \cdots \wedge \neg P$. However guarded formulae of form $\psi := (\forall x_1 . x_1 = x_1) \cdots (\forall x_d\, x_d = x_d) \neg P x_d$ have provenance values $\pi[\![\psi]\!] = m^{n^d}$.

Large differences between provenance values of guarded formulae and unrestricted first-order formulae are, for instance, witnessed by K-interpretations $\pi : \mathrm{Lit}_A(\{P,E\}) \to \mathbb{N}$ that give value 0 to all positive literals, small values, say 1 or 2, to literals $\neg Pa$ but a very large value m to literals $\neg Eab$. Maximal provenance values for guarded sentences are achieved by formulae $\psi := (\forall x_1 . x_1 = x_1) \cdots (\forall x_d\, x_d = x_d) \neg P x_d$ and these values are bound by 2^{n^d} (where $n = |A|$). Indeed, every guarded formula under the scope of a quantifier that has more than one free variable has provenance value 0. Further, if $\varphi(x_1,\ldots,x_k)$ is a quantifier-free formulae of depth d, then provenance values of its instantiations $\varphi(a_1,\ldots,a_k)$ are bound by m^{2^d}, independent of n. Even any Boolean combination of these two types of guarded formulae cannot achieve the provenance values of first-order formula of form $\forall x_1 \ldots \forall x_k \bigwedge_{i,j \leq k} \neg E x_i x_j$, which are $m^{k^2 n^k}$. □

We conclude that, as in the case of model checking, there is also in the computation of provenance values an exponential gap between the number of semiring

operations required for a model or guarded formula on one side, and for an arbitrary first-order formula on the other side. However, in cases where the size of (representations of) semiring elements is the main source of complexity, this difference tends to become less important, or even to disappear. The order of magnitude of provenance values realizable by modal or guarded formulae is similar to those for arbitrary first-order formulae, and the size of these representations dominates in many cases the number of semiring operations.

6 A more abstract view of guarded logics

Instead of the syntactic definition given above for the guarded fragment, one may use a different presentation that is based on classical first-order syntax (without relativization of quantifiers), but uses a guarded modification of the semantics which restricts all valuations of the free variables to guarded tuples. This leads to a more semantic view of guardedness which is also more flexible and easily adapts to other variants of guarded logics.

Definition 2. Let $\mathfrak{A}$ be a τ-structure with finite relational vocabulary τ and universe A. A *guard system* $\mathbb{G}$ for $\mathfrak{A}$ is a collection $\mathbb{G} \subseteq \mathscr{P}(A)$ of subsets of A, which is downwards closed in the sense that $g' \subseteq g \in \mathbb{G}$ implies $g' \in \mathbb{G}$. A tuple $\overline{a} = (a_1, \ldots, a_k) \in A^k$ is $\mathbb{G}$-guarded if the set of its components, $[\overline{a}] := \{a_1, \ldots, a_k\}$, belongs to $\mathbb{G}$. The $\mathbb{G}$-semantics for first-order logic on $\mathfrak{A}$ is defined by inductive rules for $\mathfrak{A} \models_{\mathbb{G}} \varphi(\overline{a})$, for $\mathbb{G}$-guarded $\overline{a}$, which are the usual ones for first-order logic, except that for quantifiers we have

$$\begin{aligned} \mathfrak{A} \models_{\mathbb{G}} \exists y \varphi(\overline{a}, y) \quad &:\Longleftrightarrow \quad \mathfrak{A} \models \varphi(\overline{a}, b) \text{ for some } b \text{ such that } [\overline{a}, b] \in \mathbb{G} \\ \mathfrak{A} \models_{\mathbb{G}} \forall y \varphi(\overline{a}, y) \quad &:\Longleftrightarrow \quad \mathfrak{A} \models \varphi(\overline{a}, b) \text{ for all } b \text{ such that } [\overline{a}, b] \in \mathbb{G} \end{aligned}$$

We now consider the specific guard system $\mathbb{G} \subseteq \mathscr{P}(A)$ that consist of those sets $g \subseteq A$ that are guarded in $\mathfrak{A}$, in the sense that there is an atomic fact $\mathfrak{A} \models R\overline{a}$ such that $g \subseteq [\overline{a}]$. Thus, the guarded tuples are obtained from tuples that occur in some atomic fact by copying, permuting or deleting components. It is not difficult to see that the guarded fragment is equivalent to the $\mathbb{G}$-semantics of first-order logic, for this particular guard system $\mathbb{G}$.

Theorem 6.1 *There exists a translation $\varphi \mapsto \varphi^g$ from* FO *to* GF*, such that, for every formula $\varphi(\overline{x}) \in \mathrm{FO}(\tau)$, every τ-structure $\mathfrak{A}$, and every tuple $\overline{a}$ with $[\overline{a}] \in \mathbb{G}$ we have that*

$$\mathfrak{A} \models_{\mathbb{G}} \varphi(\overline{a}) \quad \Longleftrightarrow \quad \mathfrak{A} \models \varphi^g(\overline{a}).$$

Further, for every $\varphi(\overline{x}) \in \mathrm{GF}$, the $\mathbb{G}$-semantics coincides with the usual first-order semantics.

Notice that the natural model checking game for $\mathbb{G}$-semantics of ψ on $\mathfrak{A}$, $\mathscr{G}_{\mathbb{G}}(\mathfrak{A}, \psi)$ is obtained as the restriction of usual model checking game $\mathscr{G}(\mathfrak{A}, \psi)$ to

the positions $\varphi(\bar{a})$ for which $[\bar{a}] \in \mathbb{G}$. In particular moves from $Qy\varphi(\bar{a})$ to $\varphi(\bar{a},b)$ where $[\bar{a},b] \notin \mathbb{G}$ are no longer available.

This abstract view of guarded logics also leads to a somewhat different view of provenance, based on K-interpretations that give values not only to the literals, but separately also to the guard system.

Definition 6.2 Let K be a commutative semiring K, τ a relational vocabulary, and $\mathbb{G} \subseteq \mathscr{P}(A)$ be a guard system for the set A. A *K-interpretation* for τ, A, and $\mathbb{G}$ consists of a function $\pi : \mathrm{Lit}_A(\tau) \to K$ (that maps all equality and inequality literals to their truth values 0 or 1) and a function $h : \mathbb{G} \to K \setminus \{0\}$. The *$\mathbb{G}$-provenance semantics* of first-order logic then extends π to a function $\pi_{\mathbb{G}} : \mathrm{FO}(\tau) \to K$ giving to each sentence $\varphi(\bar{a})$ with $[\bar{a}] \in \mathbb{G}$ the value $\pi_{\mathbb{G}}[\![\varphi(\bar{a})]\!]$, where quantified formulae now are treated according to the rules

$$\pi_{\mathbb{G}}[\![\exists y\varphi(\bar{a},y)]\!] := \sum_{b:[\bar{a},b]\in\mathbb{G}} h([\bar{a},b]) \cdot \pi_{\mathbb{G}}[\![\varphi(\bar{a})]\!]$$

$$\pi_{\mathbb{G}}[\![\forall y\varphi(\bar{a},y)]\!] := \prod_{b:[\bar{a},b]\in\mathbb{G}} h([\bar{a},b]) \cdot \pi_{\mathbb{G}}[\![\varphi(\bar{a})]\!].$$

Notice that a *K-interpretation* for τ, A, and $\mathbb{G}$ also gives basic valuations for the terminal positions and the moves of the model checking games $\mathscr{G}_{\mathbb{G}}(A,\psi)$, for every first-order sentence ψ, which by the rules given in Sect. 3 extends to valuations f_0 and f_1 for all positions of that game.

Proposition 6.3 *For every position φ of the model checking game $\mathscr{G}_{\mathbb{G}}(A,\psi)$ and every K-*interpretation *for τ, A, and $\mathbb{G}$ we have that $f_0(\varphi) = \pi_{\mathbb{G}}[\![\varphi]\!]$ and $f_1(\varphi) = \pi_{\mathbb{G}}[\![\neg\varphi]\!]$.*

Despite Theorem 6.1, the provenance values defined by this "semantic approach" to the guarded fragment may be different from the provenance values for the syntactic presentation of GF. Indeed, we here give provenance values to the guarded tuples themselves, not to their presentations by an atomic statement. Since a guarded tuple may admit several different syntactic guards, the syntactic approach does not provide a unique provenance value for it. Morerover, even in the case where a guarded tuple has a unique guard, the semantic approach admits to separate the provenance value of its use as guard for a quantifier from its use as an atomic statement as such.

However, Theorem 6.1 implies that a formula has a non-zero provenance value in the semantic approach if, and only if, it has a non-zero value in the traditional syntactic approach.

7 Guarded negation first-order logic

Guarded negation first-order logic, denoted GNF, is a fragment of first-order logic introduced by Bárány, ten Cate and Segoufin [3], which applies the concept of

guards not to quantifiers but to negation. As we will see it in some sense generalizes the guarded logics considered so far. Guarded negation first-order logic can be defined by the grammar

$$\varphi ::= R(\bar{x}) \mid x = y \mid \exists x\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \alpha(\overline{xy}) \wedge \neg\varphi(\bar{y}),$$

where R is a relation symbol and $\alpha(\overline{xy})$ is an atomic formula that contains all free variables of $\varphi(\bar{y})$.

This yields a logic that contains the existential positive fragment of first-order logic, but also allows for some restricted negation. Further it also generalizes the traditional guarded fragment GF in the sense that every formula $\varphi(\bar{x})$ can be translated into a formula $\varphi^*(\bar{x})$ of GNF such that for every *guarded tuple* $\bar{a}$ of a structure $\mathfrak{A}$ we have that $\mathfrak{A} \models \varphi(\bar{a})$ if, and only if, $\mathfrak{A} \models \varphi^*(\bar{a})$. In particular, every sentence of GF is equivalent to a sentence of GNF. Additionally, many of the desirable properties of GF survive also for GNF; in particular this holds for the decidability of satisfiability and finite satisfiability, even for the fixed-point extension of GNF. A more detailed model-theoretic analysis of GNF has been presented in [5] and [4].

We discuss the embedding of GF into GNF established in [3]. There is one minor point that one has to take care of in such translations. In GF, a formula $\varphi(\bar{x})$ can also be used to express a property of tuples that are not necessarily guarded, because the restriction to guarded tuples appears only inside the scope of a quantifier, but not necessarily on the top level of the formula, and as far as unguarded tuples are concerned, translations from GF into other logics may be problematic. We therefore restrict formulae to talk only about guarded tuples, for instance by attaching an explicit guard atom: we say that a formula is *answer guarded*, if it is either a sentence, or of the form $\alpha(\overline{xy}) \wedge \varphi(\bar{y})$, where $\alpha(\overline{xy})$ an atomic formula containing all free variables of $\varphi(\bar{y})$.

Proposition 7.1 *Every answer guarded formula in* GF *can be translated into an equivalent* GNF*-formula via polynomial time transformation.*

Proof. Let φ be an answer guarded formula in GF. First, transform every subformula of the form $(\forall\bar{x}.\,\alpha)\psi$ into $\neg\exists\bar{x}(\alpha \wedge \neg\psi)$ and every subformula of the form $(\exists\bar{x}.\,\alpha)\psi$ into $\exists\bar{x}(\alpha \wedge \psi)$. Then consider the subformulae of φ that are of the form $\neg\vartheta$, starting with the literals. If $\neg\vartheta$ does not have any free variables, we can replace it by $\exists x(x = x \wedge \neg\vartheta)$. Now suppose that ϑ has the free variables $\bar{y}$. Then ϑ is in the scope of an innermost guard atom α and we can replace $\neg\vartheta$ by $\alpha \wedge \neg\vartheta$ since α must contain all free variables of ϑ. Implementing these replacements for every negated subformula of φ (including possibly φ itself) yields a formula φ' which is equivalent to φ and in GNF by construction. □

Bárány, ten Cate and Segoufin [3] have shown that GNF has the finite model property and have determined the complexity of the satisfiability problem, based on a reduction to GF using Rosati covers [2].

Theorem 7.2 *1. The satisfiability problem for* GNF *is* 2ExpTime*-complete.*

2. *Every satisfiable* GNF*-sentence has a finite model of size* $2^{2^{|\varphi|^{O(1)}}}$.

In the same paper, they have also shown that the evaluation problem for GNF on finite structures has a higher complexity level than GF (which is in polynomial time) but lower than full first-order logic (which is PSPACE-complete).

Theorem 7.3 *The model checking problem for* GNF *is* $\mathrm{P}^{\mathrm{NP}[O(\log^2(n))]}$*-complete.*

We shall discuss this result below.

7.1 Provenance analysis for GNF

We want to provide appropriate definitions for a provenance analysis for guarded negation first-order logic. To do this along the lines described above for other logics, we need a negation normal form for GNF. This poses a problem however, since a negation cannot be simply "pushed through" an existential quantifier; this would lead to universal quantification, which is not allowed in the syntax. To solve this problem, one could modify the syntax to allow for the use of universal quantifiers, but only in the cases where the formula can be translated back into a formula from the original syntax. However this approach leads to an artificial syntax which would still be asymmetric in the treatment of existential and universal quantifiers.

For this reason, we introduce a new variant GNF* of guarded negation first-order logic, which is equivalent to GNF for sentences and answer guarded formulae, but permits a few more formulae in order to allow symmetric use of the two quantifiers. Therefore we will be able to retain all the desirable properties of GNF for sentences and for formulae on guarded tuples, but we will have a more general syntax that has the required dualities to be amenable to a game-based approach and to semiring provenance.

Definition 3. We define GNF* as the union of two fragments GNF$^+$ and GNF$^-$ which are defined by the mutual induction

$$\varphi^+ ::= R(\bar{x}) \mid x = y \mid \exists x\varphi^+ \mid \varphi^+ \vee \varphi^+ \mid \varphi^+ \wedge \varphi^+ \mid \alpha(\overline{xy}) \wedge \varphi^-(\bar{y}) \mid \neg\varphi^-$$
$$\varphi^- ::= \neg R(\bar{x}) \mid x \neq y \mid \forall x\varphi^- \mid \varphi^- \wedge \varphi^- \mid \varphi^- \vee \varphi^- \mid \alpha(\overline{xy}) \to \varphi^+(\bar{y}) \mid \neg\varphi^+,$$

where R is a relation symbol and $\alpha(\overline{xy})$ is an atomic formula containing all free variables of the formula it is used with. The formulae $\varphi^+ \in \mathrm{GNF}^+$ are called positive, the formulae $\varphi^- \in \mathrm{GNF}^-$ are negative.

Remark. It might also be interesting to consider the fragment of this logic, where we disallow formulae of the form $\varphi^+ \wedge \psi^+$ in GNF$^+$ and $\varphi^- \vee \psi^-$ in GNF$^-$. This would lead to a nice simplification of the model checking game: between guards, only one player would move, until control switches to the other player at the next guard. But even in the model checking game of the full logic GNF* as defined above we have the useful property that only one player is in control of the assignments to

the variables, and again this control only switches at guards. We shall return to model checking games later in the section, after introducing provenance for GNF*. Before doing that, we note some interesting properties of GNF*.

Proposition 7.4 *1. Every formula in* GNF$^-$ *is equivalent to the negation of a formula from* GNF$^+$ *and vice versa. In particular,* GNF* *is closed under transformation to negation normal form.*
2. Syntactically, $\mathrm{GNF}^+ \cap \mathrm{GNF}^- = \emptyset$*. In particular every formula can be uniquely identified as positive or negative.*
3. GNF$^+$ *is equivalent to* GNF.
4. Every answer guarded formula in GNF* *is equivalent to a formula in* GNF.

Proof. If φ is in GNF$^+$, rewrite every subformula of φ that is in GNF$^-$ as the negation of a formula in GNF$^+$, starting with the outermost subformulae in GNF$^-$. Because formulae of GNF$^-$ only occur in answer guarded formulae, applying these transformations from the outside in yields a formula in GNF.

If on the other hand $\varphi(\bar{a})$ is in GNF$^-$, rewrite $\varphi(\bar{a})$ as the negation of a formula $\varphi^*(\bar{a})$ in GNF$^+$. Because $\bar{a}$ is guarded by γ, $\neg\varphi^*$ and therefore φ are equivalent to $\gamma(\bar{a}) \wedge \neg\varphi^*(\bar{a})$ and $\varphi^*(\bar{a}) \in \mathrm{GNF}^+$ is equivalent to a formula from GNF by the argument above. Therefore, φ is equivalent to a formula from GNF. □

While GNF consists of clearly positive formulae which allow negative subformulae, but only applied to a guarded tuple, GNF* also permits negative fomulae. However, unlike in first-order logic, every formula is clearly either positive, if it is in GNF$^+$, or negative, if it is in GNF$^-$. A positive formula, like a GNF-formula, is a positive or existential statement about the whole structure, possibly with negative or universal statements about guarded tuples. In contrast to that, a negative formula is a negative or universal statement about the structure, where positive or existential statements only apply if a guarding condition is fullfilled (remember that in negative formulae, the guarded subformulae have the form $\alpha \to \varphi$).

We are now ready to provide a notion of provenance for GNF*.

Definition 7.5 A *K-interpretation* $\pi : \mathrm{Lit}_A(\tau) \to K$ (for a commutative semiring K, a universe A and a relational vocabulary τ) extends to a valuation $\pi : \mathrm{GNF}^*(\tau) \to K$ by the following rules, where φ, ψ are arbitrary formulae in GNF*, whereas $\varphi^+ \in$ GNF$^+$ and $\varphi^- \in$ GNF$^-$:

$$\pi[\![\psi \vee \varphi]\!] := \pi[\![\psi]\!] + \pi[\![\varphi]\!] \qquad \pi[\![\psi \wedge \varphi]\!] := \pi[\![\psi]\!] \cdot \pi[\![\varphi]\!]$$
$$\pi[\![\exists x \varphi^+(x)]\!] := \sum_{a \in A} \pi[\![\varphi^+(a)]\!] \qquad \pi[\![\forall x \varphi^-(x)]\!] := \prod_{a \in A} \pi[\![\varphi^-(a)]\!].$$
$$\pi[\![\alpha \wedge \varphi^-]\!] := \pi[\![\alpha]\!] \cdot \pi[\![\varphi^-]\!] \qquad \pi[\![\alpha \to \varphi^+]\!] := \begin{cases} 1, & \pi[\![\alpha]\!] = 0 \\ \pi[\![\alpha]\!] \cdot \pi[\![\varphi^+]\!], & \text{otherwise} \end{cases}$$

As before, negation is handled via negation normal form: $\pi[\![\neg\varphi]\!] := \pi[\![\mathrm{nnf}(\neg\varphi)]\!]$.

As for GF, a K-valuation for GNF* may assign a different value to a a formula than the corresponding first-order K-valuation would assign to the standard rewriting in traditional first-order syntax.

7.2 Model checking games for GNF* and their provenance analysis

We define model checking games for GNF* in a similar way to those of first-order logic. To avoid player switches, which would interfere with defining a notion of provenance, we only consider formulae in negation normal form. The only difference to the model checking games for first-order logic concerns the rules for answer guarded formulae of form $\alpha \wedge \varphi^-$ in GNF$^+$ or $\alpha \to \varphi^+$ in GNF$^-$, i.e. the positions where a play switches between GNF$^+$ and GNF$^-$.

At a position $\alpha \wedge \varphi^-$ the guard α is evaluated. If it is false then Player 1 has won. If it is true then the play proceeds to φ^-. Dually, at a position $\alpha \to \varphi^+$, Player 0 has won if the guard α is false and if it is true the play proceeds to φ^+.

Proposition 7.6 *Let $\mathfrak{A}$ be a τ-structure, $\bar{a}$ a guarded tuple in $\mathfrak{A}$, and $\psi(\bar{x})$ a* GNF*-*formula. Then $\mathfrak{A} \models \psi(\bar{a})$ if, and only if, Player 0 wins the model checking game from $\psi(\bar{a})$.*

Proof. The argument, by induction on the formula, proceeds as for the general first-order model-checking game. It only remains to consider positions of the form $\alpha \wedge \varphi^-$ in GNF$^+$ and $\alpha \to \varphi^+$ in GNF$^-$ for which we have modified the rules.

If $\mathfrak{A} \models \alpha \wedge \varphi^-$ then the guard α evaluates to true and the play proceeds to position φ^- from which Player 0 wins by induction. If $\mathfrak{A} \not\models \alpha \wedge \varphi^-$ then either the guard α evaluates to false, and Player 1 wins by definition, or α evaluates to true and the play continues at φ^-. Since $\mathfrak{A} \not\models \varphi^-$ Player 1 wins by induction.

If $\mathfrak{A} \models \alpha \to \varphi^+$ then either the guard α evaluates to false, in which case Player 0 wins by definition or, if the play proceeds to φ^+, then $\mathfrak{A} \models \varphi^+$ so Player 0 wins by induction hypothesis. If $\mathfrak{A} \not\models \alpha \to \varphi^+$ then the guard α evaluates to true, so the play proceeds to φ^+ with $\mathfrak{A} \not\models \varphi^+$ and Player 1 wins by induction hypothesis. □

We next consider natural provenance evaluations for the model checking games for GNF* and show that they are compatible with the provenance definitions given above.

In a model checking game $\mathscr{G}(\mathfrak{A}, \psi)$ for a finite τ-structure $\mathfrak{A}$ and $\psi \in$ GNF* there are two kinds of terminal positions: either they are literals $\varphi \in \mathrm{Lit}_A(\tau)$ or they correspond to an answer-guarded formula where the guard is not satisfied. Given a commutative semiring K and a model-defining K-interpretation $\pi : \mathrm{Lit}_A(\tau) \to K$ with $\mathfrak{A}_\pi = \mathfrak{A}$ we obtain valuations f_0, f_1 for the terminal positions $\mathscr{G}(\mathfrak{A}, \psi)$ as follows. For $\varphi \in \mathrm{Lit}_A(\tau)$ we put $f_0(\varphi) = \pi[\![\varphi]\!]$ and $f_1(\varphi) = \pi[\![\neg\varphi]\!]$. For an answer guarded formula $\alpha \wedge \varphi^-$ with $\pi[\![\alpha]\!] = 0$ we put $f_0(\alpha \wedge \varphi^-) = 0$ and $f_1(\alpha \wedge \varphi^-) = 1$. Dually, for an answer guarded formula $(\alpha \to \varphi^+)$ with $\pi[\![\alpha]\!] = 0$ we put $f_0(\alpha \to \varphi^+) = 1$ and $f_1(\alpha \to \varphi^+) = 0$. Further we defined a valuation $h : E \to K$ of the edges of $\mathscr{G}(\mathfrak{A}, \psi)$ as follows. For every move from a node $v = (\alpha \wedge \varphi^-)$ to $w = \varphi^-$, or from $v = (\alpha \to \varphi^+)$ to $w = \varphi^+$, we put $h(vw) := \pi[\![\alpha]\!]$. For all other moves $(v, w) \in E$ we put $h(vw) := 1$.

The model checking games for GNF* do not have cycles. Therefore we can inductively extend f_0 and f_1 from the terminal position to the entire game graph as defined in Sect. 3.3, by the rules

$$f_\sigma(v) := \begin{cases} \sum_{w \in vE} h(vw) \cdot f_\sigma(w) & \text{if } v \in V_\sigma \\ \prod_{w \in vE} h(vw) \cdot f_\sigma(w) & \text{if } v \in V_{1-\sigma}. \end{cases}$$

Proposition 7.7 *Let φ be any position in the model checking game $\mathscr{G}(\mathfrak{A}, \psi)$. Then $f_0(\varphi) = \pi[\![\varphi]\!]$ and $f_1(\varphi) = \pi[\![\neg\varphi]\!]$.*

Proof. The proof is an obvious induction on φ and the only difference to the arguments for arbitrary first-order model checking games concerns the cases where φ is of form $\alpha \wedge \varphi^-$ or $\alpha \to \varphi^+$.

For $\varphi = \alpha \wedge \varphi^-$ we have that

$$f_0(\varphi) = \pi[\![\alpha]\!] \cdot f_0(\varphi^-) = \pi[\![\alpha]\!] \cdot \pi[\![\varphi^-]\!] = \pi[\![\varphi]\!].$$

Further

$$f_1(\varphi) = \begin{cases} 1 & \text{if } \pi[\![\alpha]\!] = 0 \\ \pi[\![\alpha]\!] \cdot f_1(\varphi^-) & \text{otherwise.} \end{cases}$$

On the other side

$$\pi[\![\neg\varphi]\!] = \pi[\![\alpha \to \neg\varphi^-]\!] = \begin{cases} 1 & \text{if } \pi[\![\alpha]\!] = 0 \\ \pi[\![\alpha]\!] \cdot \pi[\![\neg\varphi^-]\!] & \text{otherwise.} \end{cases}$$

Since, by induction hypothesis, $f_1(\varphi^-) = \pi[\![\neg\varphi^-]\!]$, we also have $f_1(\varphi) = \pi[\![\neg\varphi]\!]$. The arguments for $\varphi = \alpha \to \varphi^+$ are analogous. □

7.3 Algorithmic analysis

To analyze model-checking and provenance for guarded negation first-order logic algorithmically it is useful to consider its stratification via guarded negation: we define fragments GNF_k^+ and GNF_k^-, for $k \geq 1$, where GNF_1^+ is the existential positive fragment of FO, built from atomic formulae $R\overline{x}$ and $x = y$ by means of disjunction, conjunction, and existential quantification. Similarly GNF_1^- is the universal negative fragment, built from negated atoms by conjunction, disjunction, and universal quantification. For $k \geq 2$, the formulae in GNF_k^+ are defined by the grammar

$$\varphi ::= \varphi^+ \mid \exists x\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \alpha \wedge \varphi^-$$

where $\varphi^+ \in \mathrm{GF}_{k-1}^+$, $\varphi^- \in \mathrm{GNF}_{k-1}^-$ and α is a guard for φ^-. Analogously, GNF_k^- is defined by

$$\varphi ::= \varphi^- \mid \forall x\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \alpha \to \varphi^+.$$

It is well-known that the model-checking problem for existential positive formulae, and in fact even for conjunctive queries, is NP-complete. In game theoretic terms, we analyze this problem as follows. For a finite structure $\mathfrak{A}$ of size m, a tuple $\overline{a}$ that is guarded in $\mathfrak{A}$, and an existential positive formula $\psi(\overline{a})$ of size n the model

checking game $\mathscr{G}(\mathfrak{A}, \psi)$ has, in general, exponential size $O(n \cdot m^n)$ and can therefore not be explicitely constructed in an efficient way. However, the game has only a polynomial number of terminal positions, and, in contrast to games for general first-order formulae, it has the property that the exponential branching of the game is only caused by Player 0. This means that once Player 0 has committed herself to a strategy f, the reduced game graph $\mathscr{G}_f$, admitting only the plays that are consistent with f, has only polynomial size.

As a consequence model-checking games for existential positive formulae can be solved in NP, avoiding an explicit construction.

Proposition 7.8 *There is a nondeterministic polynomial-time algorithm that given a finite structure $\mathfrak{A}$, a guarded tuple $\bar{a}$, and a formula $\psi(\bar{x}) \in \mathrm{GNF}_1^+$ guesses, on the fly, a strategy f for Player 0, constructs the reduced game graph $\mathscr{G}_f$ of $\mathscr{G}(\mathfrak{A}, \psi(\bar{a}))$, and determines whether Player 0 has a winning strategy.*

This analysis can be extended to arbitrary formulae of GNF^*.

Theorem 7.9 *Given a finite structure $\mathfrak{A}$, a guarded tuple $\bar{a}$, a formula $\psi(\bar{x}) \in \mathrm{GNF}^*$, and $\sigma \in \{0,1\}$, the problem whether Player σ wins the model-checking game $\mathscr{G}(\mathfrak{A}, \psi(\bar{a}))$ is in P^{NP}.*

Proof. For $\psi \in \mathrm{GNF}_1^+$ the problem is in NP and hence also in P^{NP}. Let now $\psi \in \mathrm{GNF}_{k+1}^+$ and assume that the result has already been established for formulae in GNF_k^- and hence, by the closure of P^{NP} under complements, also for formulae in GNF_k^+. The game for a formula in GNF_{k+1}^+ can be viewed as a game for the top level existential positive formula whose terminal positions are either literals, or answer guarded formulae $\alpha \wedge \varphi^-$ with $\varphi^- \in \mathrm{GNF}_k^-$. There are only a polynomial number of such terminal nodes and we can construct them efficiently. For nodes of the form $\alpha \wedge \varphi^-$ we distinguish two cases: if α evaluates to false, we label the node as winning for Player 1. If α evaluates to true, the node is the root of a new game $\mathscr{G}(\mathfrak{A}, \varphi^-)$. In this case we apply the already established P^{NP}-algorithm to determine the winner, and label the node accordingly. We are left with a game for an existential positive formula that we can solve in P^{NP}. Altogether this is a polynomial composition of P^{NP}-algorithms which is again a P^{NP} algorithm. □

A more sophisticated implementation of this algorithmic idea results in a P^{NP} algorithm that queries the NP-oracle in a restricted way. Indeed one can make sure that each query to the oracle depends only on the answers to a logarithmic number (with respect to the the length of the formula) of previously asked queries, i.e., the problem is in the class $\mathrm{P}^{\mathrm{NP}}_{||O(\log n)}$ which, by a result due to [7], is the same as $\mathrm{P}^{\mathrm{NP}[O(\log^2 n)]}$ the class of problems solvable by a polyomial-time algorithm with at most $O(\log^2 n)$ calls to an oracle in NP. As shown by Bárány, ten Cate, and Segoufin [3] the model checking problem for GNF is actually complete for this complexity class.

For the provenance analysis of an existential positive formula in a semiring K, it does not suffice to guess a strategy and check whether it is winning. Instead one

has to sum up the provenance values of all possible strategies. For each individual strategy it is still possible to compute the value in polynomial time, provided that this is the case for the basic semiring operations, and that we have an additive cost measure. For provenance values in the semiring of natural numbers we hence have a summation over exponentially many values of a polynomial-time computable function into $\mathbb{N}$; this can be done by a #P-algorithm.

Theorem 7.10 *The problem of computing provenance values in $\mathbb{N}$ (with the standard logarithmic cost measure) for existential positive first-order formulae is #P-complete.*

It is open to what extent this result can be generalized beyond existential positive formulae, i.e. formulae in GNF_1^+, to higher levels of GNF^*.

References

1. H. Andréka, J. van Benthem, and I. Németi. Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic*, 27:217–274, 1998.
2. V. Bárány, G. Gottlob, and M. Otto. Querying the guarded fragment. In *Proceedings of the 2010 25th Annual IEEE Symposium on Logic in Computer Science*, LICS '10, pages 1–10, 2010.
3. V. Bárány, B. ten Cate, and L. Segoufin. Guarded negation. *J. ACM*, 62(3):22:1–22:26, 2015.
4. M. Benedikt, P. Bourhis, and M. Vanden Boom. Characterizing definability in decidable fixpoint logics. In *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017*, pages 107:1–107:14, 2017.
5. M. Benedikt, B. ten Cate, and M. Vanden Boom. Effective interpolation and preservation in guarded logics. *ACM Trans. Comput. Log.*, 17(2):8, 2016.
6. P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge University Press, 2001.
7. J. Castro and C. Seara. Complexity classes between Θ_k^P and Δ_k^P. *RAIRO - Theoretical Informatics and Applications - Informatique Théorique et Applications*, 30(2):101–121, 1996.
8. A. Emerson and C. Jutla. The complexity of tree automata and logics of programs. In *Proceedings of FOCS 1988*, pages 328–337, 1988.
9. E. Grädel. On the restraining power of guards. *Journal of Symbolic Logic*, 64:1719–1742, 1999.
10. E. Grädel. Why are modal logics so robustly decidable? *Bulletin of the European Association for Theoretical Computer Science*, 68:90–103, 1999.
11. E. Grädel and M.Otto. The freedoms of (guarded) bisimulation. In *Trends in Logic: Johan van Benthem on Logical and Informational Dynamics*, pages 3–31. Springer, 2014.
12. E. Grädel and V. Tannen. Semiring provenance for first-order model checking. arXiv:1712.01980 [cs.LO], 2017.
13. E. Grädel and V. Tannen. Provenance analysis for logic and games. arXiv: 1907.08470 [cs.LO], 2019.
14. E. Grädel and I. Walukiewicz. Guarded fixed point logic. In *Proceedings of LICS 1999*, pages 45–54, 1999.
15. R. Ladner. The computational complexity of provability in systems of propositional modal logic. *SIAM Journal on Computing*, 6:467–480, 1977.
16. E. Spaan. *Complexity of modal logics*. PhD thesis, University of Amsterdam, Institute for Logic, Language and Computation, 1993.
17. M. Vardi. Why is modal logic so robustly decidable? In N. Immerman and P. Kolaitis, editors, *Descriptive Complexity and Finite Models*, volume 31 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 149–184. AMS, 1997.