Decidability issues for action guarded logics

Elisabeth Goncalves* Erich Grädel*

Abstract

A general criterion for establishing the decidability of a logic is presented, based on a guarded fragment of second-order logic and the notion of tree width. The criterion is used to establish decidability results for action guarded logics (including counting quantifiers, inverse actions and/or least fixed points).

1 Motivation

Description logics come in many different varieties and there are many constructors that are used in different combinations to build description logics. To choose a particular DL (or similar formalism) for a practical application there is, as always, the trade-off between expressiveness and manageability to consider. Your particular DL should be tailored carefully such that it is (just) expressive enough for the application at hand, but nevertheless admits practical algorithms for the reasoning tasks required by the application.

It is known that some combinations of the constructors used for building DLs lead to undecidable formalisms. In that case you should probably restrict your formalism rather than spending much time in trying to build a system for reasoning about it. Hence a quick way to dermine whether a proposed DL is decidable can be useful in some cases.

We present a very general criterion for establishing decidability of a logic. Rather than building up DL-like formalisms from below (e.g. by extending $\mathcal{ALC}$ with more and more powerful operators) and proving that these remain decidable, we consider a very powerful logic, called *guarded second-order logic* (GSO) that can be shown to be decidable under the semantic restriction that only structures of bounded tree width are considered. We will then introduce some logical systems that encompass most of the description logics in common use, but that are still decidable, because (1) they can be embedded into GSO

*Mathematische Grundlagen der Informatik, RWTH Aachen, D-52065 Aachen, {goncalves,graedel}@informatik.rwth-aachen.de, WWW: http://www-mgi.informatik.rwth-aachen.de

and (2) they have a *generalized tree model property* saying that it suffices to consider models of bounded tree width. If your DL can be embedded in one of these logics (which is usually easy to see), then you immediately know that it is decidable.

In fact, some of the logics that we discuss here behave also quite reasonably in terms of complexity, in the sense that they are complete for the same complexity classes as many of the DLs that are in common use. Admittedly, this does not necessarily say very much about practical algorithms, but given that our general approach shows the applicability of the automata theoretic method for these logics, we believe that efficient and practical algorithms are possible, at least in some cases. However, for lack of space, we defer the discussion of complexity and of automata based procedures to a future paper.

Beyond this (limited) relevance of our results for designers of description logics, we find it a theoretically very intriguing problem to explore the limits of decidability for DL-like logical formalisms. Our work continues the research on *guarded logics* that have first been put forward by Andréka, van Benthem and Németi [1] in order to explain and generalize the good model-theoretic and algorithmic properties of modal logics (and hence also description logics). Guarded logics have received considerable attention in the last years and it seems that the notion of guarded quantification is indeed very important for the design of logics that are both expressive and algorithmically manageable. See [1, 2, 3, 5, 6, 10, 9, 8, 12, 13] for background and further results on guarded logics and [11] for an informal discussion.

2 A general criterion for decidability

Guarded second-order logic, abbreviated GSO, has the same syntax as ordinary second-order logic, but second-order quantifiers are restricted semantically to range over *guarded relations* only, rather than over arbitrary relations. Guarded relations contain only tuples $(b_1, \dots, b_n)$ that are covered by some ground atom (atomic fact) that holds in the structure under consideration.

Definition 2.1. Let $\mathfrak{B} = (B, R_1, \dots, R_t)$ a relational structure.

(1) A set $X = \{b_1, \dots, b_n\} \subseteq B$ is *guarded* in $\mathfrak{B}$ if there exists an atomic formula $\alpha(x_1, \dots, x_n)$ such that $\mathfrak{B} \models \alpha(b_1, \dots, b_n)$. Note that every singleton set $X = \{b\}$ is guarded (by the atom $b = b$).

(2) A tuple $(b_1, \dots, b_n) \in B^n$ is *guarded* if $\{b_1, \dots, b_n\} \subseteq X$ for some guarded set X.

(3) A *guarded m-ary relation* on $\mathfrak{B}$ is a relation $R \subseteq B^m$ that only contains guarded tuples.

(4) GSO is second-order logic (with equality, without function symbols) with the following semantics for second-order quantification. $\mathfrak{B} \models \exists R\varphi$ iff there exists a *guarded* relation R on $\mathfrak{B}$ such that $(\mathfrak{B}, R) \models \varphi$.

It is clear that the semantic restriction for the second-order quantifiers in GSO could also be captured purely syntactically, by admitting second order quantification only of the form $\exists X(\forall \boldsymbol{y}(X\boldsymbol{y} \rightarrow \mathit{guarded}(\boldsymbol{y})) \wedge \varphi(X))$ where $\mathit{guarded}(\boldsymbol{y})$ is a first-order formula expressing that $\boldsymbol{y}$ is a guarded tuple. More explicitly, on structures with relations $R_1, \ldots, R_t$

$$\mathit{guarded}(\boldsymbol{y}) := \bigvee_{i=1}^{t} \exists \boldsymbol{x}(R_i\boldsymbol{x} \wedge \bigwedge_j \bigvee_\ell y_j = x_\ell).$$

Note that GSO includes monadic second-order logic MSO. The expressive power of GSO is strictly between MSO and full second-order logic [12].

Tree width. The tree width of a structure measures how closely it ressembles a tree. Informally a structure has tree width $\leq k$, if it can be covered by (possibly overlapping) substructures of size at most $k+1$ which are arranged in a tree-like manner. Tree width is an important notion in graph theory. For instance, it is known that many difficult or undecidable computational problems on graphs become easy on graphs of bounded tree width. For instance, trees and forests have tree width 1, cycles have tree width 2, and the $n \times n$-grid has tree width n. Here we need the notion of tree width for arbitrary relational structures.

Definition 2.2. A structure $\mathfrak{B}$ (with universe B) has *tree width* k if k is the minimal natural number satisfying the following condition. There exists a directed tree $T = (V, E)$ and a function $F : V \rightarrow \{X \subseteq B : |X| \leq k+1\}$, assigning to every node v of T a set $F(v)$ of at most $k+1$ elements of $\mathfrak{B}$, such that the following two conditions hold.

(i) For every guarded set X in $\mathfrak{B}$ there exists a node v of T with $X \subseteq F(v)$.

(ii) For every element b of $\mathfrak{B}$, the set of nodes $\{v \in V : b \in F(v)\}$ is connected (and hence induces a subtree of T).

For each node v of T, $F(v)$ induces a substructure $\mathfrak{F}(v) \subseteq \mathfrak{B}$ of cardinality at most $k+1$. $\langle T, (\mathfrak{F}(v)_{v\in T})\rangle$ is called a *tree decomposition* of width k of $\mathfrak{B}$.

Definition 2.3. (Generalized tree model property). Let $\mathcal{L}$ be a logic, and $\mathcal{C}$ a class of structures. We say that $\mathcal{L}$ *has the generalized tree model property on* $\mathcal{C}$ if there exists a computable function t, assigning to every sentence $\psi \in \mathcal{L}$ a natural number $t(\psi)$ such that, whenever ψ is satisfiable on $\mathcal{C}$, then there also exists a model $\mathfrak{A} \models \psi$ such that $\mathfrak{A} \in \mathcal{C}$ and $\mathfrak{A}$ has tree width at most $t(\psi)$. In the case where $\mathcal{C}$ is the class of all structures, we simply say that $\mathcal{L}$ has the generalized tree model property.

Remark. The definition only requires that $t(\psi)$ must be computable from ψ. In all applications that have been considered so far, $t(\psi)$ can be immediately read off from simple syntactic parameters of ψ such as the maximal arity of the relation symbols in ψ, or the maximal number of free variables in its subformulae.

Theorem 2.4 (Decidability criterion). *Let $\mathcal{L}$ be a logic, $\mathcal{C}$ a class of structures such that the following two conditions are satisfied.*

(1) $\mathcal{L}$ has the generalized tree model property on $\mathcal{C}$.

(2) There is an algorithm translating every sentence $\psi \in \mathcal{L}$ into a sentence of GSO *that is equivalent to ψ on $\mathcal{C}$.*

Then $\mathrm{Sat}_{\mathcal{C}}(\mathcal{L})$, the satisfiability problem for $\mathcal{L}$ on $\mathcal{C}$, is decidable.

The proof is based on an interpretation argument from the GSO-theory of structures of bounded tree width into the monadic second-order theory of all trees (proved to be decidable in [14]).

In order to apply the decidability criterion in a particular situation, the main problem usually is to establish the generalized tree model property. There are two important methods for this purpose: (1) bisimulations and unravelling, and (2) tableaux. Both have been successfully applied. In the following we will use the bisimulation method to establish decidability results for action guarded logics.

3 Action guarded logics

Guarded fragments of first-order (and fixed point) logic are defined by restricting quantification such that, semantically speaking, each subformula can refer only to tuples of elements that are 'very close together' or 'guarded'. Syntactically this means that 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 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 GF, the guarded fragment of first-order logic [1], and also in the logics considered below, only atoms are allowed as guards (for more liberal variants, see e.g. [9]). It has been shown that GF and other guarded logics are decidable, and indeed have good model-theoretic and algorithmic properties (finite model property, generalized tree model property, efficient model checking, automata-based decision procedures etc.) which make them interesting for a number of applications of logic in computer science. See [1, 3, 5, 7, 10, 9, 11, 8, 12, 13]

The guards in GF can be viewed as generalizations of actions in modal and process logics and of the roles in description logics. However, contrary to modal or description logics, there is no strict separation in GF between state formula (or concept definitions) and actions (or roles). While this is desirable in many contexts (it increases expressiveness) it does on the other side lead to problems with some extensions of GF. In particular, it has been shown in [10] that the naive extensions of GF by counting quantifiers, functionality constraints or transitivity axioms (which are important features for description logics) are undecidable.

This motivates the study of *action guarded logics* that are built with relation symbols from two disjoint vocabularies τ_{state} and τ_{action}. Accordingly predicates are called *state predicates* and *action predicates*. Each state predicate $S \in \tau_{\text{state}}$ has a designated arity $m \geq 1$ which means that S describes properties of m-tuples. Each action predicate $A \in \tau_{\text{action}}$ comes with an arity (i, j), with $i \geq 0$, $j \geq 1$, which means that A describes an action from i-tuples to j-tuples. The separation of state predicates and action predicates makes it possible to use different formula building rules for them and thus to recover some decidability properties that fail for more naive extensions of the guarded fragment. In some cases it may even be desirable to go one step further and use several action vocabularies whose predicates are used in different ways (e.g. with or without counting quantifiers, with or without transitivity constraints etc.).

Definition 3.1. The *action-guarded fragment* AGF of first-order logic has two kind of formulae, *state formulae* and *action guards* (or simply guards) which are defined inductively as follows.

(A1) Every atomic formulae $A(\boldsymbol{x}; \boldsymbol{y})$, where A is an action predicate of arity (i, j) and $\boldsymbol{x} = x_1, \ldots, x_i$, $\boldsymbol{y} = y_1, \ldots, y_j$ are distinct variables, is an action guard from $\boldsymbol{x}$ to $\boldsymbol{y}$.

(S1) Every equality $x = y$ and every atom $S(\boldsymbol{x})$, where S is a state predicate, is a state formula.

(S2) State formulae are closed under Boolean operations.

(S3) If $\varphi(\boldsymbol{x}, \boldsymbol{y})$ is a state formula with free variables only among those displayed, $\mathit{action}(\boldsymbol{x}; \boldsymbol{y})$ is an action guard from $\boldsymbol{x}$ to $\boldsymbol{y}$, then $(\exists \boldsymbol{y} \,.\, \mathit{action}(\boldsymbol{x}; \boldsymbol{y}))\varphi(\boldsymbol{x}, \boldsymbol{y})$ and $(\forall \boldsymbol{y} \,.\, \mathit{action}(\boldsymbol{x}; \boldsymbol{y}))\varphi(\boldsymbol{x}, \boldsymbol{y})$ are state formulae.

Remark. In [3] Johan van Benthem describes two basic action guarded fragments, denoted GSAL_1 and GSAL_2. It is not difficult to see that AGF extends both of van Benthem's variants.

The action guarded fragment with counting, AGFC, extends AGF by counting countifiers. The quantifier rule in Definition 3.1 is replaced by

(S3)' If $\varphi(\boldsymbol{x}, \boldsymbol{y})$ is a state formula with free variables only among those dispayed, $action(\boldsymbol{x}; \boldsymbol{y})$ is an action guard from $\boldsymbol{x}$ to $\boldsymbol{y}$ and $m \in \mathbb{N}$, then $(\exists^{\geq m}\boldsymbol{y} \,.\, action(\boldsymbol{x}; \boldsymbol{y}))\varphi(\boldsymbol{x}, \boldsymbol{y})$ and $(\exists^{\leq m}\boldsymbol{y} \,.\, action(\boldsymbol{x}; \boldsymbol{y}))\varphi(\boldsymbol{x}, \boldsymbol{y})$ are state formulae.

The semantics is the obvious one. A special case are *functionality constraints on actions* which are expressed by formulae $(\exists^{\leq 1}y \,.\, Fxy)y = y$.

Inverse actions. A popular feature in description logics are inverse actions, which can of course also be added to AGF and AGFC.

Definition 3.2. AGFI and AGFCI are obtained by adding to AGF and AGFC, respectively, the following rule

(A2) If $action(\boldsymbol{x}; \boldsymbol{y})$ is an action guard from $\boldsymbol{x}$ to $\boldsymbol{y}$, then it is also an action guard from $\boldsymbol{y}$ to $\boldsymbol{x}$.

Counting-bisimulations. Note that AGF and AGFI are subfragments of GF so decidability for these logics follows easily from the decidability of GF. The cases of AGFC and AGFCI are more complicated. We prove, via an appropriate notion of bisimulation, that the generalized tree model property holds for these logics. Since AGFC and AGFCI are fragments of first-order logic, decidability then follows via the criterion explained above.

Definition 3.3. Let $\mathfrak{A}$ and $\mathfrak{A}'$ be two structures of vocabulary $\tau = \tau_{state} \dot{\cup} \tau_{action}$ with universes A and A', respectively. A *partial state-isomorphism* from $\mathfrak{A}$ to $\mathfrak{A}'$ is a bijection $f : X \to X'$ for $X \subseteq A$ and $X' \subseteq A'$ such that for every tuple $\boldsymbol{a} \in X^n$ and every state atom $\varphi(x_1, \dots, x_n)$, $\mathfrak{A} \models \varphi(\boldsymbol{a})$ iff $\mathfrak{A}' \models \varphi(f\boldsymbol{a})$. To put it differently, f is an isomorphism between the state-reducts of the substructures induced by X in $\mathfrak{A}$ and X' in $\mathfrak{A}'$.

Counting bisimulations are dependent on the definition of action guards that is used (e.g. on the presence or absence of inverse actions). For simplicity we first work with action guards without inverses.

Definition 3.4. A set $X \subseteq A$ is *action-guarded* in $\mathfrak{A}$ if there exists an action guard $action(\boldsymbol{x}; \boldsymbol{y})$ such that $\mathfrak{A} \models action(\boldsymbol{a}; \boldsymbol{b})$ and $X = \boldsymbol{a} \cup \boldsymbol{b}$. A *counting bisimulation* between $\mathfrak{A}$ and $\mathfrak{A}'$ is a non-empty set I of partial state-isomorphisms $f : X \to X'$ from $\mathfrak{A}$ to $\mathfrak{A}'$ such that X and X' are action-guarded sets, and for each $f : X \to X'$ in I the following condition is satisfied:

For every action guard $action(\boldsymbol{x}; \boldsymbol{y})$ from $\boldsymbol{x}$ to $\boldsymbol{y}$ and all tuples $\boldsymbol{a} \in X^i$, $\boldsymbol{a}' = f\boldsymbol{a}$ there exists a bijection $\pi : \{\boldsymbol{b} : \mathfrak{A} \models action(\boldsymbol{a}; \boldsymbol{b})\} \to \{\boldsymbol{b}' : \mathfrak{A}' \models action(\boldsymbol{a}'; \boldsymbol{b}')\}$ such that whenever $\boldsymbol{b}' = \pi\boldsymbol{b}$, the function $g : (\boldsymbol{a}, \boldsymbol{b}) \mapsto (\boldsymbol{a}', \boldsymbol{b}')$ also belongs to I.

We say that $\mathfrak{A}$ and $\mathfrak{A}'$ are *counting-bisimilar*, in symbols $\mathfrak{A} \sim_c \mathfrak{A}'$ if there exists a counting bisimulation between them.

Proposition 3.5. *If $\mathfrak{A} \sim_c \mathfrak{A}'$, then $\mathfrak{A} \equiv_{AGFC} \mathfrak{A}'$.*

Counting-bisimilar structures can in fact not even be distinguished by the logic AGFC$^\infty$ that extends AGFC by (1) conjunctions and disjunctions over any (finite or infinite) set of formulae; (2) counting quantifiers of the form $\exists^{\geq\kappa}$ and $\exists^{\leq\kappa}$ where κ can be any (finite or infinite) cardinal. Proposition 3.5 follows from the following stronger result.

Proposition 3.6. *Let I be a counting bisimulation from $\mathfrak{A}$ to $\mathfrak{A}'$, and let $\boldsymbol{a}$ and $\boldsymbol{a}'$ be tuples in $\mathfrak{A}$ and $\mathfrak{A}'$ such that $f(\boldsymbol{a}) = \boldsymbol{a}'$ for some $f \in I$. Then, for every state formula $\psi(\boldsymbol{x})$ in* AGFC$^\infty$ *we have $\mathfrak{A} \models \psi(\boldsymbol{a})$ if and only if $\mathfrak{A}' \models \psi(\boldsymbol{a}')$.*

Proof. If ψ is a state-atom then the claim is immediate from the definitions. For $\psi(\boldsymbol{x}) = \neg\varphi_i(\boldsymbol{x})$ or $\psi(\boldsymbol{x}) = \bigwedge_i \varphi_i(\boldsymbol{x})$, the induction step is trivial. Let $\psi(\boldsymbol{x}) = (\exists^{\bowtie\kappa}\boldsymbol{y} \,.\, \mathit{action}(\boldsymbol{a};\boldsymbol{y}))\varphi(\boldsymbol{a},\boldsymbol{y})$, where $\bowtie$ is $\leq$ or $\geq$. By the definition of a counting bisimulation, there exists a bijection $\pi : S = \{\boldsymbol{b} : \mathfrak{A} \models \mathit{action}(\boldsymbol{a};\boldsymbol{b})\} \to S' = \{\boldsymbol{b}' : \mathfrak{A}' \models \mathit{action}(\boldsymbol{a}';\boldsymbol{b}')\}$ and for each $\boldsymbol{b} \in S$, a partial isomorphism $g \in I$, such that $g(\boldsymbol{a},\boldsymbol{b}) = (\boldsymbol{a}',\pi\boldsymbol{b})$. By induction hypothesis we have, for each $\boldsymbol{b} \in S$ that $\mathfrak{A} \models \varphi(\boldsymbol{a},\boldsymbol{b}) \iff \mathfrak{A}' \models \varphi(\boldsymbol{a}',\pi\boldsymbol{b})$. Since π is bijective, $|\{\boldsymbol{b} \in S : \mathfrak{A} \models \varphi(\boldsymbol{a},\boldsymbol{b})\}| = |\{\boldsymbol{b}' \in S' : \mathfrak{A} \models \varphi(\boldsymbol{a}',\boldsymbol{b}')\}|$. Hence $\mathfrak{A} \models \psi(\boldsymbol{a})$ iff $\mathfrak{A}' \models \psi(f\boldsymbol{a})$. □

Theorem 3.7. *Let $\mathfrak{A}$ be an infinite structure of vocabulary $\tau = \tau_{\mathit{state}} \cup \tau_{\mathit{action}}$ such that for every action predicate of arity (i,j) we have $i+j \leq k$. Then there exists a τ-structure $\mathfrak{A}'$ of tree width $\leq k-1$ such that $\mathfrak{A} \sim_c \mathfrak{A}'$.*

Proof. We inductively construct a tree T and assign to each node v of T the following objects:

(a) an action guarded set $F(v)$ of $\mathfrak{A}$ with at most k elements.

(b) a set $F'(v)$ of the same cardinality of $F(v)$.

(c) a bijection $f_v : F(v) \to F'(v)$. This induces a unique τ_{state}-structure $\mathfrak{F}'(v)$ with universe $F'(v)$ such that f_v is a partial state isomorphism from $\mathfrak{A}$ to $\mathfrak{F}'(v)$.

(d) (except for the root) an action guard $\beta_v = \mathit{action}(\boldsymbol{a}';\boldsymbol{b}')$, where $\boldsymbol{a}' \cup \boldsymbol{b}' \subseteq F'(v)$.

The root of T is λ, with $F(\lambda) = F'(\lambda) = \varnothing$.

Suppose now that the tree has already been constructed up to node v. A tuple $\boldsymbol{a}$ of $\mathfrak{A}$ is called *new at node* v if either v is the root and $\boldsymbol{a}$ is the empty tuple, or $\boldsymbol{a} \subseteq F(v)$ but $\boldsymbol{a} \not\subseteq F(u)$ for the parent node u of v. For each tuple $\boldsymbol{a}$ that is new at v and every action guard $\mathit{action}(\boldsymbol{x};\boldsymbol{y})$ where $\boldsymbol{x}$ has the same length as $\boldsymbol{a}$, let $S := \{\boldsymbol{b} : \mathfrak{A} \models \mathit{action}(\boldsymbol{a};\boldsymbol{b})\}$. For each $\boldsymbol{b} \in S$, create a successor node $w = w[\mathit{action}(\boldsymbol{a};\boldsymbol{b})]$ of v and set

- $F(w) := \boldsymbol{a} \cup \boldsymbol{b}$.
- $F'(w) := \boldsymbol{a}' \cup \boldsymbol{b}'$ where $\boldsymbol{a}' = f_v(\boldsymbol{a})$ is defined in such a way that the function $f_w : F(w) \to F'(w)$ that maps $(\boldsymbol{a}, \boldsymbol{b})$ to $(\boldsymbol{a}', \boldsymbol{b}')$ is a bijection, and the elements of $F'(w) - F'(v)$ are fresh elements (i.e. they do not occur at any ancestor of v nor at any other child of v).
- $\beta_w := \mathit{action}(\boldsymbol{a}'; \boldsymbol{b}')$.

Finally, let $\mathfrak{A}'$ be the structure whose state reduct is $\mathfrak{A}' \restriction \tau_{state} := \bigcup_{v \in T} \mathfrak{F}'(v)$ and such that for every action-guard β, we have $\mathfrak{A}' \models \beta$ iff $\beta = \beta_v$ for some node $v \in T$. We claim that $I = \{f_v : v \in T\}$ is a counting bisimulation between $\mathfrak{A}$ and $\mathfrak{A}'$.

To see this, let $f_v \in I$, take any action guard $\mathit{action}(\boldsymbol{x}; \boldsymbol{y})$ and tuples $\boldsymbol{a}$ and $\boldsymbol{a}' = f_v(\boldsymbol{a})$. From v, move to the unique ancestor of v such that $\boldsymbol{a}$ is new at node u. Clearly we also have $f_u(\boldsymbol{a}) = \boldsymbol{a}'$. Let $S = \{\boldsymbol{b} : \mathfrak{A} \models \mathit{action}(\boldsymbol{a}; \boldsymbol{b})$ and $S' := \{\boldsymbol{b}' : \mathfrak{A} \models \mathit{action}(\boldsymbol{a}'; \boldsymbol{b}')\}$. For each $\boldsymbol{b} \in S$ we have a successor node $w[\mathit{action}(\boldsymbol{a}; \boldsymbol{b})]$ of v. We claim that the function $\pi : S \to S'$ that maps $\boldsymbol{b}$ to $f_{w[\mathit{action}(\boldsymbol{a};\boldsymbol{b})]}(\boldsymbol{b})$ is the desired bijection.

It is clear that π is injective. By the construction of $\mathfrak{A}'$, the only possibility that $\mathit{action}(\boldsymbol{a}'; \boldsymbol{b}')$ can hold in $\mathfrak{A}'$ is that $\mathit{action}(\boldsymbol{a}'; \boldsymbol{b}') = \beta_w$ for some successor node w of v. But then it must be the case that $w = w[\mathit{action}(\boldsymbol{a}; \boldsymbol{b})]$ for some $\boldsymbol{b} \in S$. Hence π is also surjective. Finally if $\pi(\boldsymbol{b}) = \boldsymbol{b}'$, then the corresponding function f_w maps $(\boldsymbol{a}, \boldsymbol{b})$ to $(\boldsymbol{a}', \boldsymbol{b}')$. Since $f_w \in I$ this proves that I is a counting bisimulation. ☐

Corollary 3.8. *The satisfiability problem for AGFC is decidable.*

In fact, a stronger result follows. Let μAGFC be the fixed point extension of AGFC, which relates to AGFC in the same way as the μ-calculus relates to propositional modal logic and μGF relates to the guarded fragment (see [13]). Fixed point formulae in μAGFC are of the form $\mu R.\psi(R, \boldsymbol{x})$ or $\nu R.\psi(R, \boldsymbol{x})$ where R is a relation symbol of arity k and ψ is a formula whose free first-order variables are among $x_1, \ldots, x_k$. The semantics of $\mu R.\psi$ is the least fixed point of the monotone operator $\psi^{\mathfrak{B}}$ sending $R \subseteq B^k$ to $\{\boldsymbol{b} \in B^k : \mathfrak{B} \models \psi(R, \boldsymbol{b})\}$. That is, $\mathfrak{B} \models \mu R.\psi(\boldsymbol{b})$ iff $\boldsymbol{b}$ is an element of the least fixed point of the operator $\psi^{\mathfrak{B}}$. Similarly for $\nu R.\psi$ and the greatest fixed point of $\psi^{\mathfrak{B}}$.

Corollary 3.9. *The satisfiability problem for* μAGFC *is decidable.*

Proof. Our reasoning follows the arguments in [8] for a similar decidability result. We combine three facts

(1) LFP, the classical fixed-point extension of first-order logic (see e.g. [4]) has the Löwenheim-Skolem property, i.e. every satisfiable formula in LFP has a finite or countable model. The same is true for μAGFC since it is a fragment of LFP.

(2) On any class of structures of bounded cardinality, fixed points can be simulated by infinitary Boolean operations (take the infinitary disjunction over the formulae that define the stages of the fixed point induction). Hence, on structures of bounded cardinality, μAGFC can be embedded into AGFC$^\infty$. But this implies that μAGFC is invariant under counting bisimulation. Therefore it satisfies the generalized tree model property.

(3) Finally, μAGFC can be embedded into GSO.

Hence the decidability criterion applies also to μAGFC. □

Adding inverse actions. With minor modifications, the same proof goes through also in the presence of inverse actions.

Theorem 3.10. *The satisfiability problems for* AGFI, AGFCI *and* μAGFCI *are decidable.*

To determine the complexity of these logics and to get practical algorithms other methods must be used (since already the the first-order theory of trees has non-elementary complexity). But the generalized tree model property is very useful also for this purpose, since it permits the use of automata-based decision procedures. For instance, alternating two-way tree automata, that have been used in [15] for deciding the μ-calculus with inverse modalities and in [13] for deciding the guarded fixed point logic μGF, can be tailored for the reasoning problems in action guarded logics. Under reasonable restrictions (that the relations have bounded arities, and that numbers in counting quantifiers are coded in unary) one can obtain in this way an EXPTIME decision procedure for μAGFCI.

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] F. BAADER, R. MOLITOR, AND S. TOBIES, *The guarded fragment of conceptual graphs*, Tech. Rep. LTCS-98-10, RWTH Aachen, 1998. See http://www-lti.informatik.rwth-aachen.de/Forschung/Papers.html.

[3] J. VAN BENTHEM, *Modal logic in two gestalts*, in Advances in Modal Logic. Proceedings of AiML 98, Uppsala 1998, M. Z. et al., ed., CSLI Publications, 2000.

[4] H.-D. EBBINGHAUS AND J. FLUM, *Finite Model Theory*, Springer, 2nd edition ed., 1999.

[5] H. Ganzinger and H. de Nivelle, *A superposition decision procedure for the guarded fragment with equality*, in Proc. 14th IEEE Symp. on Logic in Computer Science, 1999, pp. 295–305.

[6] H. Ganzinger, C. Meyer, and M. Veanes, *The two-variable guarded fragment with transitive relations*, in Proc. 14th IEEE Symp. on Logic in Computer Science, 1999, pp. 24–34.

[7] G. Gottlob, E. Grädel, and H. Veith, *Datalog LITE: A deductive query language for efficient temporal reasoning*, to appear in ACM Transaction on Computational Logic. See also: Tech. Rep. DBAI-TR-98-22, Vienna University of Technology, Institute for Information Systems, 1998.

[8] E. Grädel, *The decidability of guarded fixed point logic*, in JFAK. Essays Dedicated to Johan van Benthem on the Occasion of his 50th Birthday, J. Gerbrandy, M. Marx, M. de Rijke, and Y. Venema, eds., CD-ROM, Amsterdam University Press, 1999. See http://turing.wins.uva.nl/ j50/cdrom/.

[9] ——, *Decision procedures for guarded logics*, in Automated Deduction - CADE16. Proceedings of 16th International Conference on Automated Deduction, Trento, 1999, vol. 1632 of Lecture Notes in Artificial Intelligence, Springer-Verlag, 1999.

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

[11] ——, *Why are modal logics so robustly decidable?*, Bulletin of the European Association for Theoretical Computer Science, 68 (1999), pp. 90–103.

[12] E. Grädel, C. Hirsch, and M. Otto, *Back and forth between guarded and modal logics*, in Proc. 15th IEEE Symp. on Logic in Computer Science, 2000.

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

[14] S. Shelah, *The monadic theory of order*, Annals of Mathematics, 102 (1975), pp. 379–419.

[15] M. Vardi, *Reasoning about the past with two-way automata*, in Automata, Languages and Programming ICALP 98, vol. 1443 of Springer Lecture Notes in Computer Science, 1998, pp. 628–641.