Back and Forth Between Guarded and Modal Logics

Erich Grädel* Colin Hirsch* Martin Otto†

Abstract

Guarded fixed point logic μGF *extends the guarded fragment by means of least and greatest fixed points, and thus plays the same role within the domain of guarded logics as the modal* μ*-calculus plays within the modal domain. We provide a semantic characterisation of* μGF *within an appropriate fragment of second-order logic, in terms of invariance under guarded bisimulation. The corresponding characterisation of the modal* μ*-calculus, due to Janin and Walukiewicz, is lifted from the modal to the guarded domain by means of model theoretic translations. At the methodological level, these translations make the intuitive analogy between modal and guarded logics available as a tool in the analysis of the guarded domain.*

1. Introduction

Guarded logics generalise certain features and desirable model theoretic properties of modal logics to a much wider context. The concept of guarded quantification was introduced by Andréka, van Benthem, and Németi [1], who proposed and analysed the *guarded fragment of first-order logic*, GF. This fragment GF is seen to provide a very satisfactory basis for the explanation of quite some of the good behaviour of modal logics at the level of a rich fragment of classical first-order logic. Moreover, some of the robustness of modal logics with respect to natural extension mechanisms, concerning for instance the satisfiability problem, is also reflected in GF. Most notably, not only is GF decidable itself [1], but so is its canonical fixed point extension, μGF, [6]. μGF extends GF so as to render definable least (and greatest) fixed points of guardedly definable positive operations in arbitrary arities. In particular it extends the modal μ-calculus to the guarded domain. Unlike its modal companion, however, it no longer shares the finite model property, though it remains decidable. Another interesting feature, which again highlights the role of μGF as a high-level analogue of the μ-calculus, concerns its potential for model checking applications. The alternation-free fragment of μGF admits linear time model checking algorithms [4]. These and other results of recent research into guarded logics indicate that GF and its relatives provide a very interesting domain of logics, combining rich levels of expressiveness with a very good balance towards algorithmical issues.

Recall how *modal logics*, in the broad model theoretic sense including extensions like CTL and the modal μ-calculus, are characterised by their *invariance under bisimulation*: they cannot distinguish between bisimilar structures. Moreover, invariance under bisimulation is at the root of many of the successful model theoretic tools available for modal logics, like e.g. the tree model property which paves the way towards the use of automata theory. The eminent role that bisimulation plays in the domain of modal logics, is in the guarded domain taken by a similar but much more wide-ranging and finer notion of equivalence induced by *guarded bisimulation*.

The characteristic feature of modal quantification, which is also at the heart of bisimulation, is that one can only directly access nodes along basic edge relations. In the guarded scenario, this is generalised to simultaneous direct accessibility of all tuples that are covered (guarded) by some ground atom. These are what are called *guarded* tuples, and guarded quantification is quantification over guarded tuples. Unlike the modal case, the notion of guardedness does not impose any arity restrictions, so that guarded logics are just as meaningful over structures with higher-arity relations and can address properties of tuples.

One of the underlying technical themes of this paper, also concerning a methodological point of wider interest, is the potential of having certain reductions from the richer scenario of guarded logics to the simpler and well understood scenario of modal logics, or from guarded bisimulation to ordinary bisimulation. Corresponding ideas have been initially explored in [5] in the context of the satisfiability problem for guarded fixed point logic. This methodology is carried much further here and applied to a characterisation issue, concerning the semantic characterisation of guarded fixed point logic within a suitable second-order framework. Characterisation theorems of this type have a

*Mathematische Grundlagen der Informatik, RWTH Aachen, D-52065 Aachen, {graedel,hirsch}@informatik.rwth-aachen.de

†Department of Computer Science, University of Wales, Swansea, SA2 8PP, United Kingdom, m.otto@swan.ac.uk

strong tradition in the field. They are of particular interest since they tighten the close connection between logics in a certain family and characteristic equivalences. At the level of basic modal logic and the guarded fragment of first-order logic, the corresponding characterisation theorems are the following, see [2] and [1].

Theorem 1.1 (van Benthem). *A property of transition systems is definable in propositional modal logic if and only if it is first-order definable and invariant under bisimulation.*

Theorem 1.2 (Andréka, van Benthem, Németi). *The guarded fragment* GF *can express precisely the model classes that are first-order definable and invariant under guarded bisimulation.*

For the modal scenario, a similar and highly non-trivial analogous characterisation was given for the associated fixed point logic in [7].

Theorem 1.3 (Janin, Walukiewicz). *A property of transition systems is definable in the modal μ-calculus if and only if it is definable in monadic second-order logic and invariant under bisimulation.*

Johan van Benthem has raised the question whether μGF admits a similar characterisation, in terms of guarded bisimulation invariance, within some suitable framework of second-order logic. It should be noted that MSO is clearly not the right framework, since monadic second-order quantification does not suffice to simulate the fixed point definitions in μGF. Full second-order logic, on the other hand, is obviously too strong; there are simple examples showing that even for bisimulation invariant properties, full second-order logic goes far beyond the expressive power of μGF. The resulting fragment is, for instance, no longer decidable. It turns out, however, that there is a natural analogue for MSO, which we call *guarded second-order logic* GSO. GSO is best characterised in semantic terms, as full second-order logic with a semantics that restricts second-order quantifiers to range over sets of guarded tuples, rather than over arbitrary relations. The precise definition will be given in section 3, where we also discuss several syntactic variants which turn out to have exactly the expressive power of GSO. And indeed we find the following.

Main Theorem. *Guarded fixed point logic μGF can define precisely the model classes that are definable in guarded second-order logic and are invariant under guarded bisimulation.*

2. Preliminaries

Transition systems and bisimulation. Transition systems (or Kripke structures) are structures whose universe is a set of states (worlds), labelled by unary predicates (basic propositions), and carrying binary transition relations labelled by actions (accessibility relations). We typically write $\mathcal{K} = (V,(E_a)_{a\in A},(P_b)_{b\in B})$ for a transition system with state set V based on a set B of atomic propositions and a set A of actions.

Definition 2.1. A *bisimulation* between two transition systems $\mathcal{K} = (V,(E_a)_{a\in A},(P_b)_{b\in B})$ and $\mathcal{K}' = (V',(E'_a)_{a\in A},(P'_b)_{b\in B})$ is a non-empty relation $Z \subseteq V \times V'$, respecting the P_b in the sense that $v \in P_b$ iff $v' \in P'_b$, for all $b \in B$ and $(v,v') \in Z$, and satisfying the following back and forth conditions.

Forth: for all $(v,v') \in Z$, $a \in A$ and every w such that $(v,w) \in E_a$, there exists a w' such that $(v',w') \in E'_a$ and $(w,w') \in Z$.

Back: for all $(v,v') \in Z$, $a \in A$ and every w' such that $(v',w') \in E'_a$, there exists a w such that $(v,w) \in E_a$ and $(w,w') \in Z$.

Two transition systems with distinguished nodes are *bisimilar*, $\mathcal{K},u \sim \mathcal{K}',u'$, if there is a bisimulation Z between them with $(u,u') \in Z$. We say that two *trees are bisimilar* if they are bisimilar at their roots λ and λ', and just write $\mathcal{T} \sim \mathcal{T}'$ for $\mathcal{T},\lambda \sim \mathcal{T}',\lambda'$.

Every $(\mathcal{K},u)$ has a canonical bisimilar companion which is a tree. This *unravelling* of $\mathcal{K}$ from node u, $\mathcal{T}(\mathcal{K},u)$, is naturally induced on the set of all edge-labelled paths from u in $\mathcal{K}$. E.g. such a path is in P_b over $\mathcal{T}(\mathcal{K},u)$ if its terminal node is in P_b over $\mathcal{K}$. It is easy to see that each $\mathcal{K},u$ is bisimilar to its unravelling: $\mathcal{K},u \sim \mathcal{T}(\mathcal{K},u),u$. Hence, as far as bisimulation invariant properties are concerned, we can restrict our attention to trees.

Definition 2.2. The *unravelling* $\mathcal{T}(\mathcal{K},u)$ of a transition system $\mathcal{K}$ from node u is the tree of all paths through $\mathcal{K}$ that start at u. More formally, given $\mathcal{K} = (V,(E_a)_{a\in A},(P_b)_{b\in B})$ with $u \in V$, the unravelling is $\mathcal{T}(\mathcal{K},u) = (V^{\mathcal{T}},(E^{\mathcal{T}}_a)_{a\in A},(P^{\mathcal{T}}_b)_{b\in B})$, where $V^{\mathcal{T}}$ is the set of all sequences $\overline{v} = v_0a_0v_1a_1\cdots v_{r-1}a_{r-1}v_r$ where $v_i \in V$, $a_i \in A$ such that $v_0 = u$ and $(v_{i-1},v_i) \in E_{a_i}$; $P^{\mathcal{T}}_b$ contains the sequences $v_0a_0v_1a_1\cdots v_{r-1}a_{r-1}v_r$ with $v_r \in P_b$; $E^{\mathcal{T}}_a$ contains the pairs $(\overline{v},\overline{v}av)$ in $V^{\mathcal{T}} \times V^{\mathcal{T}}$.

It is easy to see that each pair $(\mathcal{K},u)$ is bisimilar to its unravelling: $\mathcal{K},u \sim \mathcal{T}(\mathcal{K},u),u$. Hence, as far as bisimulation invariant properties are concerned, we can restrict our attention to trees.

Bisimilar trees admit special *minimal* bisimulations, which may be constructed inductively, level by level starting from (λ,λ') in such a way that at each new level we add a tuple (w,w') to Z only where this is required by a back or forth requirement from the previous level. It is clear that the resulting bisimulation is minimal in the sense that no proper subset would still be a bisimulation. Note that minimal bisimulations are in general not unique. They have, however, the following useful properties.

Lemma 2.3. *Let Z be a minimal bisimulation between $\mathcal{T}$ and $\mathcal{T}'$. If $(v,v') \in Z$ and if u and u' are the parent nodes of v and v', respectively, then also $(u,u') \in Z$ and $(u,v) \in E_a$ iff $(u',v') \in E_a$. Therefore, a minimal bisimulation satisfies the back and forth requirements also with respect to the converses of the E_a. The converse forth property w.r.t. E_a, for instance, is the following. For every $(v,v') \in Z$ and every u such that $(u,v) \in E_a$, there exists a u' such that $(u',v') \in E'_a$ and $(u,u') \in Z$.*

Proof. It is straightforward to show that otherwise $Z - \{(v,v')\}$ would still be a bisimulation. □

It follows that w.r.t. a minimal bisimulation Z between trees $\mathcal{T}$ and $\mathcal{T}'$ and for $(v,v') \in Z$, any path from v in $\mathcal{T}$ can be lifted to a bisimilar path from v' in $\mathcal{T}'$. In particular consider for any w in $\mathcal{T}$ the unique minimal connecting path $(z_0, z_1, \dots, z_r)$ from $v = z_0$ to $w = z_r$ (from v up to the first common ancestor of v and w, and from there down to w). Then there is a path $(z'_0, \dots, z'_r)$ in $\mathcal{T}'$, labelled exactly the same as the original path, and such that $(z_i, z_i) \in Z$ for all i.

2.1. Modal logics

We recall the definitions of propositional modal logic and the μ-calculus. The formulae of these logics are evaluated on Kripke structures at a particular state. Given a formula ψ and a transition system $\mathcal{K}$ with state v, we write $\mathcal{K}, v \models \psi$ to denote that the formula ψ holds in $\mathcal{K}$ at state v.

Propositional modal logic. We describe propositional (multi-)modal logic ML for several transition relations, i.e., for reasoning about transition systems $\mathcal{K} = (V, (E_a)_{a \in A}, (P_b)_{b \in B})$ where in general A may have more than one element. In the literature on modal logic, this system is sometimes called K_n (where $n = |A|$ is the number of actions or 'modalities').

Syntax of ML. The formulae of ML are defined by the following rules.

- Each atomic proposition P_b is a formula.
- If ψ and φ are formulae of ML, then so are $(\psi \wedge \varphi)$, $(\psi \vee \varphi)$ and $\neg\psi$.
- If ψ is a formula of ML and $a \in A$ is an action, then $\langle a \rangle \psi$ and $[a]\psi$ are formulae of ML.

If there is only one transition relation, $A = \{a\}$, one simply writes $\Box$ and $\Diamond$ for $[a]$ and $\langle a \rangle$, respectively.

Semantics of ML. Let ψ be a formula of ML, $\mathcal{K} = (V, (E_a)_{a \in A}, (P_b)_{b \in B})$ a transition system, and $v \in V$ a state. In the case of atomic propositions, $\psi = P_b$, we have $\mathcal{K}, v \models \varphi$ iff $v \in P_b$. Boolean connectives are treated in the natural way. Finally for the semantics of the modal operators we put

$$\mathcal{K}, v \models \langle a \rangle \psi \text{ iff } \mathcal{K}, w \models \psi \text{ for } \textit{some } w \text{ with } (v,w) \in E_a$$
$$\mathcal{K}, v \models [a] \psi \text{ iff } \mathcal{K}, w \models \psi \text{ for } \textit{all } w \text{ with } (v,w) \in E_a$$

The μ-calculus L_μ. The propositional μ-calculus L_μ is propositional modal logic augmented with least and greatest fixed points. It subsumes almost all of the commonly used modal logics, in particular LTL, CTL, CTL*, PDL and also many logics used in other areas of computer science, for instance description logics.

Syntax of L_μ. The μ-calculus extends propositional modal logic ML (including propositional variables $X, Y, \dots$, also viewed as monadic second-order variables) by the following rule for building fixed point formulae.

- If ψ is a formula in L_μ, and X is a propositional variable that does not occur negatively in ψ, then $\mu X.\psi$ and $\nu X.\psi$ are L_μ formulae.

Semantics of L_μ. The semantics of the μ calculus is given as follows. A formula $\psi(X)$ with propositional variable X defines on every transition system $\mathcal{K}$ (with state set V, and with interpretations for other free second-order variables that ψ may have besides X) an operator $\psi^{\mathcal{K}} : \mathrm{Pot}(V) \to \mathrm{Pot}(V)$ assigning to every set $X \subseteq V$ the set

$$\psi^{\mathcal{K}}(X) := \{v \in V : (\mathcal{K}, X), v \models \psi\}.$$

As X occurs only positively in ψ, $\psi^{\mathcal{K}}$ is *monotone* for every $\mathcal{K}$, and therefore has a least and a greatest fixed point. Now we put

> $\mathcal{K}, v \models \mu X.\psi$ iff v is an element of the least fixed point of the operator $\psi^{\mathcal{K}}$. Similarly $\mathcal{K}, v \models \nu X.\psi$ iff s is contained in the greatest fixed point of $\psi^{\mathcal{K}}$.

Remark. By the well known duality between least and greatest fixed points, $\nu X.\psi$ is equivalent to $\neg \mu X.\neg\psi[X/\neg X]$. Hence we could eliminate greatest fixed points. However, it will be more convenient to keep least and greatest fixed points and to work with formulae in *negation normal form*, where negations are applied only to atomic propositions.

There is a variant of L_μ which admits systems of *simultaneous fixed points*. These do not increase the expressive power but sometimes allow for more straightforward formalisations. Here one associates with any tuple $\overline{\psi} = (\psi_1, \dots, \psi_k)$ of formulae $\psi_i(\overline{X}) = \psi_i(X_1, \dots, X_k)$, all positive in the X_i, a new formula $\varphi = \mu \overline{X}.\overline{\psi}$. The semantics of φ

is induced by the least fixed point of the monotone operator $\psi^{\mathcal{K}}$ mapping $\overline{X}$ to $\overline{X}'$ where $X_i' = \{v \in V : (\mathcal{K}, \overline{X}), v \models \psi\}$. More precisely, $\mathcal{K}, v \models \varphi$ iff v is an element of the first component of the least fixed point of the above operator. Similar conventions apply w.r.t. simultaneous greatest fixed point. It is well known that simultaneous fixed points can be uniformly eliminated in favour of individual nested fixed points.

It is easy to see that all formulae of ML and L_μ are invariant under bisimulation.

3. Guarded first and second-order logics

Definition 3.1. Let $\mathfrak{B}$ be a structure with universe B and vocabulary τ.

(i) A set $X \subseteq B$ is *guarded* in $\mathfrak{B}$ if there exists a ground atom $\alpha(b_1, \ldots, b_k)$ such that $\mathfrak{B} \models \alpha(b_1, \ldots, b_k)$ and $X = \{b_1, \ldots, b_k\}$.

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

(iii) A tuple $(b_1, \ldots, b_k) \in B^k$ is a *guarded list* in $\mathfrak{B}$ if its components are pairwise distinct and $\{b_1, \ldots, b_k\}$ is a guarded set. We admit the empty list $\emptyset$ as a guarded list.

(iv) A relation $X \subseteq B^n$ is *guarded* if it only consists of guarded tuples.

Note that a singleton set $X = \{b\}$ is always guarded by the atom $b = b$. The cardinality of guarded sets in $\mathfrak{B}$ is bounded by the maximal arity of the relations in τ, the *width of* τ. Guarded tuples, however, can have any length. Guarded lists will be of technical interest later as succinct tuple representations of guarded subsets.

The guarded fragment GF. The guarded fragment extends modal logic to a richer fragment of first-order logic. Its characteristic feature is a relativised pattern of quantification, which generalises modal quantification.

Syntax of GF. The formulae of GF (in vocabulary τ) are defined by the following rules.

(i) All quantifier free first-order formulae are formulae of GF.

(ii) If ψ and φ are formulae of GF, then so are $(\psi \wedge \varphi)$, $(\psi \vee \varphi)$ and $\neg\psi$.

(iii) If $\psi(\overline{x}, \overline{y})$, with free variables among those listed, is a formula of GF and $\alpha(\overline{x}, \overline{y})$ is a τ-atom in which all displayed variables actually occur, then $\exists\overline{y}\big(\alpha(\overline{x}, \overline{y}) \wedge \psi(\overline{x}, \overline{y})\big)$ and $\forall\overline{y}\big(\alpha(\overline{x}, \overline{y}) \rightarrow \psi(\overline{x}, \overline{y})\big)$ are formulae of GF.

The atoms α relativising GF quantifications are called *guards*. We shall often also use the more intuitive notation $(\forall\overline{y}.\,\alpha)\psi$ and $(\exists\overline{y}.\,\alpha)\psi$ as shorthand for correspondingly relativised first-order quantification.

Semantics of GF. The semantics of GF is the usual one for first-order formulae.

As one simple example of a formula in GF, which will be useful later, consider the formula guarded($\overline{x}$) in variables $\overline{x} = (x_1, \ldots, x_k)$ defining over τ-structures the set of all guarded k-tuples, cf. Definition 3.1. For any complete equality type on $\{1, \ldots, k\}$ specified by a quantifier-free formula $\eta(\overline{x})$ in the language of just $=$, let $\overline{x}^\eta$ be the subtuple of $\overline{x}$ comprising precisely one variable from each $=$-class specified by η. Let $\alpha(\overline{y}, \overline{x}^\eta)$ be a τ-atom in which all variables in $\overline{x}^\eta$ actually occur, the $\overline{y}$ new, i.e. disjoint form $\overline{x}$. Put $\xi_{\eta\alpha}(\overline{x}) := \eta(\overline{x}) \wedge \exists\overline{y}\alpha(\overline{y}, \overline{x}^\eta)$. For the degenerate case of η_0, specifying the equality type of a singleton tuple $x_1 = x_2 = \ldots = x_k$, we just put $\xi_0 = \eta_0(\overline{x})$. It is easily checked that the disjunction over all these formulae $\xi_{\eta\alpha}(\overline{x})$ is as desired.

Note that first-order quantification over an individual single free variable is always admissible in GF, since singletons are guarded (by an $=$-atom): $\exists x\varphi(x) \equiv \exists x(x = x \wedge \varphi(x))$ is in GF if free$(\varphi) \subseteq \{x\}$.

Guarded fixed point logic μGF. Guarded fixed point logic μGF as introduced in [6] is the natural extension of GF by means of least and greatest fixed points (or corresponding systems of simultaneous fixed points).

Syntax of μGF. Starting from GF, with second-order variables $X, Y, Z, \ldots$ that are treated like predicates in τ but may *not* be used in guards, we augment the syntax of GF by the following rule for least and greatest fixed points.

- If $\psi = \psi(X, \overline{x})$ is a formula of μGF, which is positive in X, X k-ary and $\overline{x} = (x_1, \ldots, x_k)$, and all free (first-order) variables of ψ are among these x_i, then $\mu X.\psi$ and $\nu X.\psi$ are also formulae of μGF.

Semantics of μGF. The semantics of $\mu X.\psi$ is the natural one associated with the least fixed point of the monotone operator $\psi^{\mathfrak{B}}$ sending $X \subseteq B^k$ to $\{\overline{b} \in B^k : \mathfrak{B} \models \psi(X, \overline{b})\}$. More precisely,

$\mathfrak{B} \models \mu X.\psi(\overline{b})$ iff $\overline{b}$ is an element of the least fixed point of the operator $\psi^{\mathfrak{B}}$.
Similarly for $\nu X.\psi$ and the greatest fixed point of $\psi^{\mathfrak{B}}$.

One may also admit simultaneous least (and greatest) fixed points w.r.t. tuples of formulae $\overline{\psi} = (\psi_1(X_1, \ldots, X_k, \overline{x}^1), \ldots, \psi_k(X_1, \ldots, X_k, \overline{x}^k))$, where the X_i have only positive occurrences, and the $\overline{x}^i$ contain all free variables in ψ_i and match the arity of the X_i. Then we obtain new formulae $\varphi = \mu\overline{X}.\overline{\psi}$ and $\varphi = \nu\overline{X}.\overline{\psi}$, whose semantics is the natural one associated with the first component X_1 of the least or greatest fixed point of the corresponding monotone operator. As with the μ-calculus, one finds that simultaneous fixed points can be eliminated in μGF, too.

Guarded second-order logic. We introduce the natural second-order extension of the guarded fragment, or the guarded fragment of second-order logic, which relative to GF and μGF occupies a role analogous to that of MSO relative to ML and L_μ. The naturalness of this logic is further demonstrated below, where we show that the corresponding level of expressiveness is surprisingly robust under a number of changes in the actual formalisation and syntax. Indeed, we shall show that three natural candidates for a second-order guarded logic all have the same expressive power. It should be stressed that for all considerations, guardedness (of sets, tuples, or relations) always refers to guardedness w.r.t. the underlying vocabulary τ; at no point will second-order variables be admitted as guards.

In our preferred definition of guarded second-order logic we simply use the syntax of ordinary second-order logic, but restrict it semantically by the stipulation that all second-order quantifiers range just over *guarded relations*, rather than over arbitrary relations. We refer to this semantic restriction as *guarded semantics* for the second-order quantifiers. It is clear, however, that this stipulation may alternatively be captured purely syntactically, by only allowing occurrences of atoms $X\bar{x}$ in conjunction with the GF-formula guarded$(\bar{x})$, which says that $\bar{x}$ is a guarded tuple, thus effectively restricting X to its guarded part.

Definition 3.2. *Guarded second-order logic* GSO is second-order logic with guarded semantics for the second-order quantifiers.

Note that GSO includes full first-order logic. Hence GSO is undecidable and, unlike GF and μGF, not invariant under guarded bisimulation (cf. Definition 4.2). Also note that, as singletons are always guarded, the monadic version of guarded second-order logic coincides with full MSO. Consequently, since MSO is strictly more expressive than FO, the same is true for GSO. Furthermore, we shall see in Lemma 3.5 below that GSO collapses to MSO over words. The robustness of GSO, and its place properly in between MSO and full second-order SO, is underlined by the following.

Lemma 3.3. *The following fragments of second-order logic are equally expressive (with respect to sentences):*

(1) The extension of GF *by full second-order quantification.*

(2) The extension of GF *by second-order quantification with guarded semantics.*

(3) Guarded second-order logic GSO.

Proof. It suffices to argue for translations from (1) and (3) into (2).

For (1) $\mapsto$ (2) consider a second-order variable X in a formula according to (1), which is meant to range over arbitrary rather than guarded relations. Any atom $X\bar{x}$ occurring in a GF sentence necessarily is in the scope of a guarded quantification $(Q\bar{y}\,.\,\alpha(\bar{y},\bar{z}))\varphi$ where the occurrence of $\bar{x}$ in $X\bar{x}$ is free in φ, whence the $\bar{x}$ all occur in α. It follows that the truth value of $X\bar{x}$ for non-guarded tuples has no impact on the truth value of φ.

For (3) $\mapsto$ (2) it suffices to show that unrestricted first-order quantification can be simulated by guarded (in fact: monadic) second-order quantification over GF. To this end, each element variable x is replaced by a set variable X, and we use the following rules for translating formulae.

$$\begin{aligned} x = y &\quad\mapsto\quad \forall x(Xx \leftrightarrow Yx) \\ R\bar{x} &\quad\mapsto\quad (\exists\bar{x}\,.\,R\bar{x}) \textstyle\bigwedge_i X_i x_i) \\ \exists x\varphi(x,\bar{y}) &\quad\mapsto\quad \exists X\big(\varphi(X,\bar{y}) \wedge \text{singleton}(X)\big), \end{aligned}$$

where singleton(X) is a formula stating that X contains exactly one element:
$\exists x Xx \wedge \forall Y\big((\forall x(Yx \to Xx) \to (\forall x \neg Yx \vee (\forall x\, Yx \leftrightarrow Xx))\big)$.

Note that these translations and in particular singleton(X) are in GF, since first-order quantification over a single free first-order variable is always guarded. □

The following two lemmas show that GSO lies strictly between MSO and SO.

Lemma 3.4. GSO *is strictly more expressive than* MSO.

Proof. We show that the existence of a Hamiltonian cycle in an undirected graph can be expressed in GSO. It is known, however, that Hamiltonicity is not expressible in MSO, see e.g. [3]. Here is a GSO-sentence expressing Hamiltonicity:

$$\begin{aligned} \exists H\big((&\forall x\forall y Hxy \to Exy) \wedge \\ &(\forall x\exists y Hxy \wedge \forall z Hxz \to z = y) \wedge \\ &(\forall x\exists y Hyx \wedge \forall z Hzx \to z = y) \wedge \\ &(\forall x\forall y\exists P(Px \wedge Py \wedge \forall z Pz \to \exists z' Pz' \wedge Hzz')). \end{aligned}$$

□

Lemma 3.5. SO *is strictly more expressive than* GSO. *In particular* GSO *collapses to* MSO *over words.*

Proof. We can show that every guarded set over words can be encoded into a series of monadic predicates. The predicate of maximal arity used in words is the edge relation, so a guarded set contains at most two incident nodes. Because of the linear order we can choose e.g. the element a to represent a set $\{a,b\}$ iff Eab holds.

The encoding then uses one monadic predicate for each second-order predicate and each possibility of forming a tuple of appropriate arity out of two variables. Hence GSO is not more expressive than MSO over words, i.e. able to define exactly the regular languages. On the other hand full second-order logic is known to capture the polynomial-time hierarchy. $\square$

To summarise, we have the following hierarchy of logics

$$\text{ML} \subsetneq \text{GF} \subsetneq \text{FO} \subsetneq \text{MSO} \subsetneq \text{GSO} \subsetneq \text{SO}.$$

3.1. A normal form for guarded logics.

We present a normal form for GF and GSO that will be useful in the following. Let $X = \{x_1, x_2, \dots\}$ and $Y = \{y_1, y_2, \dots\}$ be two disjoint sets of variables. Let Z stand for either X or Y. GF_X and GF_Y are defined inductively as follows.

(1) Every relational atomic formula α with $\text{free}(\alpha) \subseteq Z$ belongs to GF_Z.

(2) A boolean combination of formulae in GF_Z also belongs to GF_Z.

(3) Let $m, n \in \mathbb{N}$, $\overline{y} = (y_1, \dots, y_n)$, and let ρ be any partial bijection between $\{1, \dots, n\}$ and $\{1, \dots, m\}$. Then, for every guard $\alpha(y_1, \dots, y_n)$ and $\varphi(y_1, \dots, y_n) \in \text{GF}_Y$ the formula $\big(\exists \overline{y}.\ \bigwedge_{\rho(i)=j} y_i = x_j \wedge \bigwedge_{i \neq j} y_i \neq y_j \wedge \alpha(\overline{y})\big)\varphi(\overline{y})$ is in GF_X. By interchanging x- and y-variables we obtain an analogous rule for GF_Y.

It should be noted that the formulae in GF_X and GF_Y are syntactically *not* in GF. It is clear, however, that these formulae are logically equivalent to guarded ones. Let $\text{GF}_0 := \text{GF}_X \cup \text{GF}_Y$. These syntactic stipulations extend from GF to GSO in the obvious way. We let GSO_0 be the extension of GF_0 by second-order quantification over guarded relations.

In the sequel, relativised quantifications of the type $(\exists \overline{y}.\ \bigwedge_{\rho(i)=j} y_i = x_j \wedge \bigwedge_{i \neq j} y_i \neq y_j \wedge \alpha(\overline{y}))$ as used in GF_0 will be abbreviated as $(\exists^{\neq} \overline{y}.\, \rho(\overline{x}, \overline{y}) \wedge \alpha(\overline{y}))$.

Proposition 3.6. *Every sentence in* GF *is equivalent to a sentence in* GF_0.

The proof is not difficult but a bit technical. We just explain the idea. First, it is well known, that first order sentences can be reformulated so that in all subformulae distinct variables are always to be interpreted by distinct elements. One way to make this precise is to use the quantifier $\exists^{\neq}$ rather than $\exists$ where $\exists^{\neq} x\, \varphi(x, z_1, \dots, z_n)$ means that there exists an x that is distinct from $z_1, \dots, z_n$ such that $\varphi(x, z_1, \dots, z_n)$ holds. Obviously, this does not change the expressive power of first-order sentences, since $\exists x\, \varphi(x, z_1, \dots, z_n)$ is equivalent to $\bigvee_{i=1}^{n} \varphi(z_i, z_1, \dots, z_n) \vee \exists^{\neq} x\, \varphi(x, z_1, \dots, z_n)$. We combine this with the idea that any quantification over a part of the free variables in a formula should come with a complete renaming of all variables, so that we move, say, from x-variables to y-variables. Consider a formula of the form $\exists^{\neq} x_1 x_3 \varphi(x_1, x_2, x_3, x_4)$. We then replace this by the equivalent formula $(\exists^{\neq} y_1 \cdots y_4 \,.\, y_2 = x_2 \wedge y_4 = x_4)\varphi(y_1, y_2, y_3, y_4)$. (Here the notation $\exists^{\neq}\overline{y}$ means that the quantified y-variables must assume distinct values, but the values of a y-variable can be the same as for an x-variable.)

To see why this might be useful (beyond the applications in this paper actually) let us consider another simple example. In the evaluation game for the sentence

$$\begin{aligned}\psi := &\exists^{\neq} x_1 \cdots x_4 (R x_1 \cdots x_4 \wedge \\ &\quad \exists^{\neq} x_1 x_5\, \varphi(x_1, x_5, x_2) \wedge \exists^{\neq} x_3\, \varphi(x_3, x_4, x_3))\end{aligned}$$

the verifier first has to pick distinct elements $a_1, \dots, a_4$ and is then challenged by the falsifier to justify one of the three conjuncts. Suppose that she has to justify the subformula $\exists^{\neq} x_1 x_5\, \varphi(x_1, x_5, x_2)$. This means that she has to keep a_2 and extend it by some new a_1 and a_5 so that she can win the evaluation game for $\varphi(a_1, a_5, a_2)$ holds. Similarly, if she is challenged to verify the third conjunct she has to produce a new a_3 so that she wins the game for $\varphi(a_3, a_4, a_3)$. This becomes more transparent if we reformulate ψ in GF_0 as

$$\begin{aligned}&\exists^{\neq} x_1 \cdots x_4 (R x_1 \cdots x_4 \wedge (\exists^{\neq} y_1 y_2 y_3 \,.\, y_3 = x_2) \\ &\varphi(y_1, y_2, y_3) \wedge \exists^{\neq} y_1 y_2 \,.\, y_2 = x_4)\varphi(y_1, y_2, y_1)).\end{aligned}$$

Indeed this formulation makes it apparent that the verifier moves from a tuple $(x_1, \dots, x_4)$ to a new tuple (y_1, y_2, y_3) or (y_1, y_2) subject to explicitly given equality constraints.

Clearly this normal form extends to other guarded logics. In particular, let GSO_0 be the extension of GF_0 by second-order quantification over guarded relations.

Corollary 3.7. *Every sentence in* GSO *is equivalent to a sentence in* GSO_0.

4. Guarded bisimulation and tree representations

Guarded bisimulation is for GF what bisimulation is for ML.

Definition 4.1. A *guarded bisimulation* between two τ-structures $\mathfrak{A}$ and $\mathfrak{B}$ is a non-empty set I of finite partial isomorphisms $f : X \to Y$ from $\mathfrak{A}$ to $\mathfrak{B}$, where $X \subseteq A$ and $Y \subseteq B$ are guarded sets, such that the following back and forth conditions are satisfied. For every $f : X \to Y$ in I:

Forth: for every guarded set $X' \subseteq A$ there exists a partial isomorphism $g : X' \to Y'$ in I such that f and g agree on $X \cap X'$.

Back: for every guarded set $Y' \subseteq B$ there exists a partial isomorphism $g : X' \to Y'$ in I such that f^{-1} and g^{-1} agree on $Y \cap Y'$.

Two τ-structures $\mathfrak{A}$ and $\mathfrak{B}$ are *guarded bisimilar* (in symbols: $\mathfrak{A} \sim_g \mathfrak{B}$) if there exists a guarded bisimulation between them. We refer to the relation $\sim_g$, which obviously is an equivalence relation between structures, as *guarded bisimulation equivalence*.

Definition 4.2. We say that a sentence ψ is *invariant under guarded bisimulation* if it does not distinguish between guarded bisimilar structures, i.e. if $\mathfrak{A} \sim_g \mathfrak{B}$ and $\mathfrak{A} \models \psi$ then also $\mathfrak{B} \models \psi$. A logic L is invariant under guarded bisimulation if all its sentences are.

Proposition 4.3. GF *and* μGF *are invariant under guarded bisimulation.*

The guarded Ehrenfeucht-Fraïssé game. It is well understood how guarded bisimulation equivalence may be described by means of an associated Ehrenfeucht-Fraïssé game. We indicate the characteristic features of the *guarded game*, in a version directly relating to our GF_0 normal form. Two players, player I and player II, take turns to place and relocate two groups of corresponding, labelled pebbles on elements of the underlying structures $\mathfrak{A}$ and $\mathfrak{B}$, respectively. The rules of the game are such that after each round the groups of pebbles positioned in $\mathfrak{A}$ and $\mathfrak{B}$, respectively, label guarded lists $\overline{a}$ and $\overline{b}$ in such a way that the correspondence $\overline{a} \mapsto \overline{b}$ is a partial isomorphism. In each round, player I has the choice in which of the two structures to play. In the chosen structure, player I then may leave some pebbles fixed, remove some from the board, and re-locate some others. The only restriction is that the new pebble positions again have to label a guarded list. Player II then has to reposition the corresponding pebbles in the other structure so as to produce a locally isomorphic configuration. Player II loses if no such response is available. Now $\mathfrak{A}$ and $\mathfrak{B}$ are guarded bisimilar iff player II can always respond indefinitely, i.e. iff there is a winning strategy for player II in the infinite guarded game. It is apparent that a guarded bisimulation in the sense of Definition 4.1 *is* a formalisation of a (non-deterministic) strategy for player II.

We now use the intuition behind the game to introduce the following structural transformations. One which abstracts from a given τ-structure $\mathfrak{B}$ a *tree representation* $\mathcal{T}(\mathfrak{B})$, which fully describes as a transition system the behaviour of $\mathfrak{B}$ in the guarded game, and thus characterises $\mathfrak{B}$ up to guarded bisimulation equivalence. Another one which, conversely, associates with a tree $\mathcal{T}$ a τ-structure $\mathfrak{D}(\mathcal{T})$, such that the game behaviour specified by $\mathcal{T}$ is realised in the guarded game on $\mathfrak{D}(\mathcal{T})$.

The guiding idea behind these transformations is that guarded bisimulations at the level of the τ-structures lift to ordinary bisimulations at the level of the abstracted transition systems. In particular, $\mathfrak{B}^* := \mathfrak{D}(\mathcal{T}(\mathfrak{B})) \sim_g \mathfrak{B}$ will be a tree-like variant of $\mathfrak{B}$, intuitively corresponding to a *guarded unravelling* of $\mathfrak{B}$, as considered e.g. in [1] and [5].

From structures to trees. Recall from Definition 3.1 that a guarded list in $\mathfrak{B}$ is a tuple $(b_1, \ldots, b_k)$ of *distinct* elements such that $\{b_1, \ldots, b_k\}$ is a guarded set. We regard such guarded lists as descriptions of positions over $\mathfrak{B}$ in the restricted guarded game. The nodes of the induced tree are associated with guarded lists over $\mathfrak{B}$, and we symmetrise the description of the game so as to allow re-labellings (permutations) of the guarded list in conjunction with every move in the game. The information to be recorded in each node $v = (b_1, \ldots, b_k)$ is precisely the isomorphism type of the induced substructure on $\{b_1, \ldots, b_k\}$ in $\mathfrak{B}$. The information to be recorded concerning possible moves describes the constraints imposed on a move from $u = (c_1, \ldots, c_l)$ to $w = (d_1, \ldots, d_k)$ by some choice of elements that remain fixed (according to player I's choice).

For a τ-structure $\mathfrak{B}$ we work with the following vocabulary $\tilde{\tau}$ for the associated tree. If m is the width (maximal arity of predicates) of τ, let S be the set of all τ-structures $\mathfrak{A}$ with universe $\{1, \ldots, k\}$, $0 \le k \le m$, where $\mathfrak{A} = \emptyset$ is admitted for $k = 0$. Let F be the set of all pairs (k, ρ), where $0 \le k \le m$ and ρ is a partial bijection from $\{1, \ldots, m\}$ to $\{1, \ldots, k\}$. Then $\tilde{\tau}$ has monadic predicates $P_{\mathfrak{A}}$ for all $\mathfrak{A} \in S$, and binary predicates E^k_ρ for all $(k, \rho) \in F$. Let G be the set of all guarded lists over $\mathfrak{B}$. Define V as the set of all sequences $g_0(k_1, \rho_1)g_1(k_2, \rho_2)g_2 \ldots g_n$ where all $g_i \in G$, $g_0 = \emptyset$ and for all $g_i(k_{i+1}, \rho_{i+1})g_{i+1}$ we have $k_{i+1} = |g_{i+1}|$ and for all $j \in \mathrm{dom}(\rho)$ the j-th element at g_{i+1} is the same as the $\rho(j)$-th element at g_i. A node $v = g_0 \ldots g_n$ of $\mathcal{T}(\mathfrak{B})$ is naturally associated with the guarded list $g_n = (b_1, \ldots, b_k)$ in $\mathfrak{B}$, in which the sequence terminates. In particular we set $|v| := |g_n| = k$ and let $\mathfrak{A}_v$ be the unique $\mathfrak{A} \in S$ that is isomorphic with $\mathfrak{B} {\restriction} \{b_1, \ldots, b_k\}$ via $i \mapsto b_i$. Then the tree associated with $\mathfrak{B}$ is

$$\mathcal{T}(\mathfrak{B}) = \big(V, (P_{\mathfrak{A}})_{\mathfrak{A} \in S}, (E^k_\rho)_{(k,\rho) \in F}\big),$$

$$\text{where} \quad \begin{array}{rcl} P_{\mathfrak{A}} & = & \{v \in V : \mathfrak{A}_v = \mathfrak{A}\}, \\ E^k_\rho & = & \{(v, v(k,\rho)g) \in V^2\}. \end{array}$$

The following is a direct consequence of the fact that the transition system $\mathcal{T}(\mathfrak{B})$ is precisely set up to capture the behaviour of $\mathfrak{B}$ w.r.t. the guarded game.

Lemma 4.4. *For all τ-structures $\mathfrak{B}$ and $\mathfrak{B}'$: $\mathfrak{B} \sim_g \mathfrak{B}'$ iff $\mathcal{T}(\mathfrak{B}) \sim \mathcal{T}(\mathfrak{B}')$.*

From trees to structures. Conversely, we would like to associate with a tree $\mathcal{T}$ of type $\tilde{\tau}$ a τ-structure $\mathfrak{D}$ for which $\mathcal{T}(\mathfrak{D}) \sim \mathcal{T}$. This is clearly not possible for arbitrary $\mathcal{T}$.

Definition 4.5. Let $\mathcal{T}$ be any $\tilde{\tau}$-tree. We call $\mathcal{T}$ *consistent* if the following are satisfied.

(a) For each node v there is a unique $\mathfrak{A} \in S$ such that $\mathcal{T} \models P_{\mathfrak{A}}(v)$, denoted $\mathfrak{A}_v$. $\mathfrak{A}_\lambda = \emptyset$.

(b) If $(u,v) \in E^k_\rho$ then the partial bijection ρ is an isomorphism between $\mathfrak{A}_v \restriction \mathrm{dom}(\rho)$ and $\mathfrak{A}_u \restriction \mathrm{im}(\rho)$, and for each isomorphism $\pi : \mathfrak{A} \to \mathfrak{A}_v$ from some structure $\mathfrak{A} \in S$ to $\mathfrak{A}_v$ there exists a node w such that $\mathfrak{A}_w = \mathfrak{A}$ and $(u,w) \in E^k_{\rho\circ\pi}$.

Lemma 4.6. *Within the class of $\tilde{\tau}$-trees, the class of consistent trees is bisimulation invariant and first-order definable.*

Every tree $\mathcal{T}(\mathfrak{B})$ is consistent. Conversely, if $\mathcal{T} = (V, (P_{\mathfrak{A}})_{\mathfrak{A}\in S}, (E^k_\rho)_{(k,\rho)\in F})$ is a consistent $\tilde{\tau}$-tree, then we may define an associated τ-structure $\mathfrak{D}(\mathcal{T})$ as follows. Let $U := \{(v,i) : v \in V, i \in \mathfrak{A}_v\}$. Let $\sim$ be the reflexive and symmetric transitive closure of the following relation on U:

$$\big\{((v,i),(w,j)) \in U^2 : (v,w) \in E^k_\rho, (k,\rho) \in F, \rho(j) = i\big\}$$

The universe of $\mathfrak{D}(\mathcal{T})$ is $D := U/_\sim$, the set of $\sim$-equivalence classes $[v,i]$ in U. We say that an equivalence class $d = [v,i]$ lives at node u if $[v,i] = [u,j]$ for some j. We observe that if an equivalence class d lives at nodes u and v in $\mathcal{T}$ then it must also live at every node of the unique shortest path connecting u to v in $\mathcal{T}$. It follows from consistency condition (b), that we may consistently interpret every k-ary predicate in τ over $\mathfrak{D}(\mathcal{T})$ by putting

$$\begin{aligned} &\overline{d} = ([v,i_1],\dots,[v,i_k]) \in R^{\mathfrak{D}(\mathcal{T})} \\ \text{iff}\quad &\mathfrak{A}_v \models R(i_1,\dots,i_k). \end{aligned}$$

Note that $\overline{d} = (d_1,\dots,d_k)$ is a guarded list in $\mathfrak{D}(\mathcal{T})$ if and only if $\overline{d} = ([w,1],\dots,[w,k])$ for some node $w \in \mathcal{T}$ of size $|w| = k$. We say that v *represents* the guarded list $\overline{d}$.

Lemma 4.7. *For all τ-structures $\mathfrak{B}$: $\mathfrak{D}(\mathcal{T}(\mathfrak{B})) \sim_g \mathfrak{B}$.*

Proof. Let $\mathcal{T} = \mathcal{T}(\mathfrak{B})$, $\mathcal{K} = \mathcal{K}(\mathfrak{B})$ and $\mathfrak{D} = \mathfrak{D}(\mathcal{T})$. Recall that $\mathcal{T}$ is the unravelling of $\mathcal{K}$, whence any node v of $\mathcal{T}$ may be associated with a unique node v_0 in $\mathcal{K}$ (the last node in the unravelled path that gives rise to v). Further recall that v_0, being a node in $\mathcal{K}$, is a guarded list in $\mathfrak{B}$.

For any guarded list $\overline{d}$ in $\mathfrak{D}$ and any node v of $\mathcal{T}$ representing this guarded list, let $f_{\overline{d},v}$ be the bijection which maps $\overline{d}$ to the guarded list v_0 in $\mathfrak{B}$. Note that as a map $f_{\overline{d},v}$ only depends on the guarded set $\{d_1,\dots,d_k\}$ and not on the order of components or on the chosen representative v. If v and v' represent the same guarded set, they are linked in $\mathcal{T}$ by a path along which all components always live together. The corresponding path in $\mathcal{K}$ similarly links v_0 to v'_0, with matching permutations along that path.

We claim that the set of all these $f_{\overline{d},v}$ is a guarded bisimulation between $\mathfrak{D}$ and $\mathfrak{B}$. It is obvious from the construction that the $f_{\overline{d},v}$ are partial isomorphisms whose domains are guarded sets. The forth property is immediate from the construction. We indicate the argument for the back property. Let $\overline{d}$ and $\overline{d}'$ be guarded lists in $\mathfrak{D}$, $f_{\overline{d},v} : \overline{d} \mapsto v_0$. Let $\overline{c}$ be the tuple of common components in $\overline{d}$ and $\overline{d}'$. Let v' be any representative of the guarded list $\overline{d}'$. The tuple $\overline{c}$ lives along the unique shortest path from v to v' in $\mathcal{T}$. Projecting this path down to $\mathcal{K}$ we see that $f_{\overline{d}',v'}$ and $f_{\overline{d},v}$ agree on their common domain. $\square$

Remark. $\mathfrak{B}^* := \mathfrak{D}(\mathcal{T}(\mathfrak{B}))$ may be regarded as a *guarded unravelling* of $\mathfrak{B}$, analogous to the standard unravelling of transition systems. Indeed, the resulting guarded bisimilar structure $\mathfrak{B}^*$ is also tree-like in that it has a tree decomposition of width $m-1$, where m is the width of τ. The naked tree $\mathcal{T}(\mathfrak{B})$, stripped of its labels and regarded as a directed graph, together with the natural association of a node v of $\mathcal{T}(\mathfrak{B})$ with the guarded set it represents in $\mathfrak{B}^*$, induces a tree decomposition of $\mathfrak{B}^*$ in the usual sense. Tree unravellings, and the corresponding generalised tree model property for GF and some of its relatives have been put to use e.g. in [1] and [5].

The following proposition extends the intuition that the bisimulation type of $\mathcal{T}(\mathfrak{B})$ captures the guarded bisimulation type of $\mathfrak{B}$ to the setting of all consistent trees. The proof is via a canonical lift of tree bisimulations.

Proposition 4.8. *For any consistent $\tilde{\tau}$-trees $\mathcal{T}$ and $\mathcal{T}'$: if $\mathcal{T} \sim \mathcal{T}'$, then $\mathfrak{D}(\mathcal{T}) \sim_g \mathfrak{D}(\mathcal{T}')$.*

Proof. Let $Z \subseteq V \times V'$ be a *minimal* bisimulation between $\mathcal{T}$ and $\mathcal{T}'$, cf. Lemma 2.3.

For each pair $(v,v') \in Z$ let $f_{vv'}$ be the function that maps the guarded list represented by v to that represented by v'. Clearly, $f_{vv'}$ is a partial isomorphism, since $\mathfrak{A}_v = \mathfrak{A}_{v'}$. Note also that for $(u,v) \in E^k_\rho$ and $(u',v') \in E^{k'}_\rho$ the maps $f_{uu'}$ and $f_{vv'}$ agree on their common domain: this follows from the construction of $\mathfrak{D}(\mathcal{T})$ and $\mathfrak{D}(\mathcal{T}')$, because elements represented at u and v are identified in $\mathfrak{D}(\mathcal{T})$ via ρ in exactly the same way as the corresponding elements at u' and v' are identified in $\mathfrak{D}(\mathcal{T}')$.

We claim that the set of all $f_{vv'}$, for $(v,v') \in Z$, is a guarded bisimulation between $\mathfrak{D}(\mathcal{T})$ and $\mathfrak{D}(\mathcal{T}')$.

To verify, for instance, the forth condition, let $\overline{c}$ and $\overline{d}$ be guarded lists in $\mathfrak{D}(\mathcal{T})$, represented by u and v in $\mathcal{T}$, respectively. Let $(u,u') \in Z$, $f_{uu'} : \overline{c} \mapsto \overline{c}'$. We need to find $v' \in \mathcal{T}'$ such that $(v,v') \in Z$ and such that $f_{vv'}$ agrees with $f_{uu'}$ on their common domain. Let X be the set of common elements in $\overline{c}$ and $\overline{d}$. Consider the unique shortest path from u to v in $\mathcal{T}$. As Z is minimal, this path gives rise to a bisimilar path from u' to some v' in $\mathcal{T}$. Now $f_{vv'}$ is as desired: the elements of X live at all nodes on the path from u to v, whence all the intermediate mappings $f_{ww'}$ along the path, and in particular $f_{vv'}$ respect $f_{uu'} \restriction X$. $\square$

5. Back and forth

Recall the characterisation of the modal μ-calculus from Theorem 1.3. We want to apply this characterisation in restriction to trees, and therefore refer to the following variant, which is proved en route to Theorem 1.3 in [7].

Theorem 5.1 (Janin, Walukiewicz). *A class of trees is definable in the modal μ-calculus if and only if it is definable in monadic second-order logic and closed under bisimulation within the class of all trees.*

Towards a reduction of our main theorem to Theorem 5.1, we define a "forth" translation that maps every sentence $\psi \in \mathrm{GSO}[\tau]$ to a formula $\psi^{\rightarrow}(x) \in \mathrm{MSO}[\tilde{\tau}]$ with one free variable, and a "back" translation that maps every formula $\varphi \in \mathrm{L}_\mu[\tilde{\tau}]$ to a sentence $\varphi^{\leftarrow} \in \mu\mathrm{GF}[\tau]$. These translation will be such that

(1) If $\mathcal{T}$ is a consistent tree with root λ, then $\mathcal{T} \models \psi^{\rightarrow}(\lambda)$ iff $\mathfrak{D}(\mathcal{T}) \models \psi$.

(2) If $\mathfrak{B}$ is a τ-structure and λ is the root of $\mathcal{T}(\mathfrak{B})$, then $\mathcal{T}(\mathfrak{B}), \lambda \models \varphi$ iff $\mathfrak{B} \models \varphi^{\leftarrow}$.

It follows from (1) that GSO sentences that are invariant under guarded bisimulation are mapped to MSO formulae that are bisimulation invariant on consistent trees.

Before giving the formal definitions, we informally discuss the main problems arising from the differences between the guarded and the modal viewpoint.

We wish to translate GSO sentences to MSO formulae and L_μ formulae back to μGF sentences. We want a modal formula to hold at some node v if and only if the corresponding guarded formula holds of the guarded list represented by v. For second-order variables we need to map sets of guarded tuples to sets of nodes and vice versa. For each node v of a consistent tree $\mathcal{T}$, the associated structure $\mathfrak{A}_v$ has in general many different guarded tuples that can occur in any single guarded set. Therefore a second-order variable Z in a GSO sentence will be translated into a sequence $Z^{\rightarrow}$ of set variables $Z_{i_1,\dots,i_r}$, one for each (local) choice of elements. In the other direction we have to deal with monadic second-order variables, which in general range over sets of nodes of arbitrary size. Consequently a monadic second-order variable X is translated into a sequence $X^{\leftarrow}$ of second-order variables $X_i, 0 \le i \le m$, such that each X_i ranges over guarded tuples of length i.

5.1. From guarded to monadic second-order logic.

Without loss of generality we restrict attention to GSO sentences in GSO_0, cf. Proposition 3.7. Let m be the width of τ, i.e. the maximal arity of relations in τ.

Definition 5.2. Let $\mathcal{T}$ be a consistent $\tilde{\tau}$-tree, and let $\mathfrak{D}(\mathcal{T})$ be the associated τ-structure. If Z is an r-ary second-order variable, then $Z^{\rightarrow} := (Z_{i_1,\dots,i_r})_{\{i_1,\dots,i_r \le m\}}$ is the corresponding sequence of monadic predicates. A tuple $J^{\rightarrow}$ of monadic predicates $J_{i_1,\dots,i_r}$ on $\mathcal{T}$ *encodes* an r-ary guarded relation J on $\mathfrak{D}(\mathcal{T})$ iff $J_{i_1,\dots,i_r} = \{v : ([v,i_1],\dots,[v,i_r]) \in J\}$ for all $i_1,\dots,i_r \le m$.

Not all sequences $\overline{J}$ of monadic predicates over $\mathcal{T}$ do indeed encode a guarded relation over $\mathfrak{D}(\mathcal{T})$. To do so, they have to satisfy the following correctness conditions.

(a) $J_{i_1,\dots,i_\ell}$ only contains nodes v where all i_j are in $\mathfrak{A}_v$.
(b) $\overline{J}$ is consistent on tuples living at different nodes, i.e. if in $\mathfrak{D}(\mathcal{T})$ a tuple $(d_1,\dots,d_r)$ is represented by $(i_1,\dots,i_r)$ at node u and by $(j_1,\dots,j_r)$ at node v, then $u \in J_{i_1\cdots i_r}$ iff $v \in J_{j_1\cdots j_r}$.

Lemma 5.3. *For each $r \le m$ there exists a first-order formula* $\mathrm{correct}(Z^{\rightarrow})$ *that expresses the correctness conditions.*

These conditions are necessary and sufficient in the sense that a tuple $\overline{J}$ encodes a guarded relation on $\mathfrak{D}(\mathcal{T})$, if and only if $\mathcal{T} \models$ $\mathrm{correct}(\overline{J})$.

Proof. Note that it suffices to express condition (b) for adjacent nodes to enforce it globally. Thus the consistency requirement for $Z^{\rightarrow}$ can be expressed by a first order formula $c(Z^{\rightarrow})$ that states condition (a) in an obvious way and contains, for each pair $(k,\rho) \in F$ and each tuple $(i_1,\dots,i_r)$ a clause

$$\forall x \forall y (E^k_{\rho(x,y)} \rightarrow (Z_{i_1,\dots,i_r}(y) \leftrightarrow Z_{\rho(i_1),\dots,\rho(i_r)}(x))).$$

The proof of the adequacy claim is straightforward. □

First-order quantifiers also require a special treatment. Let us first consider the case where $\mathcal{T}$ is a tree representation $\mathcal{T}(\mathfrak{B})$. By construction such trees satisfy a strong homogeneity condition, namely for all nodes u,u' and all successors v of u there is a successor v' of u' such that the subtrees with roots v and v' are isomorphic. To put it differently, an indistinguishable copy of any guarded list anywhere in $\mathfrak{D}(\mathcal{T})$ is available locally, at some child of the current node. Therefore guarded first-order quantifications over $\mathfrak{D}(\mathcal{T})$ can be simulated over $\mathcal{T}$ by moving to an immediate successor of the current node (i.e. by a modal quantifier $\Diamond$ or $\Box$). However, if $\mathcal{T}$ is an arbitrary consistent $\tilde{\tau}$-tree, this is no longer the case. To verify a formula of the form $(\exists^{\neq}\overline{y}\,.\,\rho(\overline{x},\overline{y}) \wedge \alpha(\overline{y}))\varphi(\overline{y})$ we want to move from the current tuple $\overline{x}$ to a new tuple $\overline{y}$, guarded by α, such that $\varphi(\overline{y})$ is true and the overlap conditions for $\overline{x}$ and $\overline{y}$ as stated by $\rho(\overline{x},\overline{y})$ are satisfied. In arbitrary consistent trees, such a witness need not exist locally, but may only occur as a remote node, which is linked to the current node by a path along which the common components according to ρ are kept.

To capture this situation in MSO we use a sequence $\overline{W}$ of monadic predicates W^k_ρ, one for each $(k,\rho) \in F$, that partition the set of nodes according to their size and their overlap with the current node. The proof of the following is then straightforward.

Lemma 5.4. *There is a first-order formula* F-part$(z,\overline{W})$ *expressing the following correctness conditions on partitions with respect to node z. For every consistent tree $\mathcal{T}$, $u \in \mathcal{T}$, and every sequence $\overline{W} = (W^k_\rho)_{(k,\rho)\in F}$ of monadic predicates on $\mathcal{T}$, we have $\mathcal{T} \models$* F-part$(u,\overline{W})$ *iff $W^k_\rho = \{w : |\mathfrak{A}_w| = k$ and, for all $i,j,[w,i] = [u,j] \iff \rho(i) = j\}$.*

Proof. As in the previous lemma, it suffices to express the required condition at the current node and for every edge. The formula F-part$(z,\overline{W})$ states that

(a) The sets $\overline{W}$ form a partition of the universe.

(b) The node z itself belongs to $W^{|\mathfrak{A}_z|}_{\mathrm{id}}$.

(c) If $y \in W^k_\rho$, then $|\mathfrak{A}_y| = k$.

(d) For all (k,ρ) and $(\ell,\sigma) \in F$, if $(x,y) \in E^k_\rho$, $|\mathfrak{A}_x| = \ell$, then $x \in W^\ell_\sigma$ iff $y \in W^k_{\sigma\circ\rho}$.

By induction on the distance from z one easily shows that F-part$(z,\overline{W})$ expresses the right property. ☐

The translation. Recall that formulae in GSO_0 either belong to GSO_X and have all their free first-order variables in X, or they belong to GSO_Y and have all their free first-order variables in Y, where X and Y are two disjoint sets of variables. Further, formulae in GSO_X that start with a quantifier are of the form $\psi(\overline{x}) := \exists^{\neq}\overline{y}\,.\,\rho(\overline{x},\overline{y}) \wedge \alpha(\overline{y}))\varphi(\overline{y})$ with $\alpha(\overline{y}), \varphi(\overline{y}) \in \mathrm{GSO}_Y$.

We inductively translate every formula $\psi(x_1,\dots,x_k,Z_1,\dots,Z_r) \in \mathrm{GSO}_X[\tau]$ into a $\mathrm{MSO}[\tilde{\tau}]$ formula $\psi^{\rightarrow}(x,Z_1^{\rightarrow},\dots,Z_r^{\rightarrow})$, with a single free first-order variable x and sequences of monadic second-order variables $Z_i^{\rightarrow}$ that correspond to the second-order variables Z_i. Similarly, formulae in GSO_Y are translated into formulae $\psi^{\rightarrow}$ with free first-order variable y. We just present the translation for formulae in GSO_X:

(1) If $\psi = R(x_{i_1},\dots,x_{i_r})$ for $R \in \tau$, set $\psi^{\rightarrow}(x) := \bigvee\{P_{\mathfrak{A}}(x) : \mathfrak{A} \in S, \mathfrak{A} \models R(i_1,\dots,i_r)\}$.

(2) If $\psi = Z(x_{i_1},\dots,x_{i_r})$, for some r-ary relation variable Z, set $\psi^{\rightarrow}(x) := Z_{i_1,\dots,i_r}(x)$.

(3) Let $(\neg\psi)^{\rightarrow}(x) := \neg(\psi^{\rightarrow}(x))$, $(\psi \vee \varphi)^{\rightarrow}(x) := \psi^{\rightarrow}(x) \vee \varphi^{\rightarrow}(x)$, $(\psi \wedge \varphi)^{\rightarrow}(x) := \psi^{\rightarrow}(x) \wedge \varphi^{\rightarrow}(x)$.

(4) If $\psi = (\exists^{\neq}\overline{y}\,.\,\rho(\overline{x},\overline{y}) \wedge \alpha(\overline{y}))\varphi(\overline{y})$, let $\psi^{\rightarrow}(x) := (\exists\overline{W}\,.\,\text{F-part}(x,\overline{W}))(\exists y\,.\,W^{|\overline{x}|}_\rho(y))(\alpha \wedge \varphi)^{\rightarrow}(y)$.

(5) For $\psi = \exists Z\varphi$ with r-ary relation variable Z, let $\psi^{\rightarrow}(x) := (\exists Z^{\rightarrow}\,.\,\mathrm{correct}(Z^{\rightarrow}))\varphi^{\rightarrow}(x)$.

Theorem 5.5. *Let $\mathcal{T}$ be a consistent $\tilde{\tau}$-tree with root λ, let $\mathfrak{D}(\mathcal{T})$ be the associated τ-structure, and let ψ be a sentence in* $\mathrm{GSO}[\tau]$. *Then $\mathfrak{D}(\mathcal{T}) \models \psi$ iff $\mathcal{T} \models \psi^{\rightarrow}(\lambda)$.*

Proof. This theorem is a consequence of the following more general statement. Consider any formula $\psi(x_1,\dots,x_k,Z_1,\dots,Z_s)$ in $\mathrm{GSO}_X[\tau]$ with free first- and second-order variables as displayed. Its translation into $\mathrm{MSO}[\tilde{\tau}]$ is $\psi^{\rightarrow}(x,Z_1^{\rightarrow},\dots,Z_s^{\rightarrow})$. Let $(d_1,\dots,d_k)$ be a guarded list in $\mathfrak{D}(\mathcal{T})$, and let v be a node of $\mathcal{T}$ such that $|\mathfrak{A}_v| = k$ and $d_i = [v,i]$. Let $J_1,\dots,J_s$ be sets of guarded tuples in $\mathfrak{D}(\mathcal{T})$, and let $J_1^{\rightarrow},\dots,J_s^{\rightarrow}$ be their representations according to Definition 5.2.

Claim. $\mathfrak{D}(\mathcal{T}) \models \psi(d_1,\dots,d_k,J_1,\dots,J_r) \iff \mathcal{T} \models \psi^{\rightarrow}(v,J_1^{\rightarrow},\dots,J_r^{\rightarrow})$.

Note that for sentences (i.e. $k = r = 0$) and for $v = \lambda$, the claim implies the theorem. The claim itself is established inductively. The cases corresponding to (1) — (3) are immediate.

(4) Let $\psi := (\exists^{\neq}\overline{y}\,.\,\rho(\overline{x},\overline{y}) \wedge \alpha(\overline{y}))\varphi(\overline{y})$, $\psi^{\rightarrow}(x) := (\exists\overline{W}\,.\,\text{F-part}(x,\overline{W}))(\exists y\,.\,W^{|\overline{x}|}_\rho(y))(\alpha \wedge \varphi)^{\rightarrow}(y)$.

As second-order variables play no role for this case, we suppress them. Suppose that $\mathfrak{D}(\mathcal{T}) \models \psi(\overline{d})$. Then there exists a guarded list $\overline{e} = (e_1,\dots,e_\ell)$ of length $\ell = |\overline{y}|$ such that $\mathfrak{D}(\mathcal{T}) \models \alpha(\overline{e}) \wedge \varphi(\overline{e})$ and $e_i = d_j$ iff $\rho(i) = j$. Since $\overline{e}$ is guarded, there exists a node w such that all e_i live together at w. Actually, due to the last of the conditions for consistent trees, we can assume that $\overline{e} = ([w,1],\dots,[w,\ell])$ and $|\mathfrak{A}_w| = \ell$. We know that there exists a $\overline{W}$ satisfying F-part$(v,\overline{W})$ and that for all $(i,j) \in \rho$ we have $[w,i] = [v,j]$. It follows that $\mathcal{T} \models W^\ell_\rho(w)$ and, by induction hypothesis, $\mathcal{T} \models (\alpha \wedge \varphi)^{\rightarrow}(w)$. Therefore $\mathcal{T} \models \psi^{\rightarrow}(v)$.

Conversely, suppose that $\mathcal{T} \models \psi^{\rightarrow}(v)$. Hence for the (unique) tuple $\overline{W}$ satisfying F-part$(v,\overline{W})$, there exists a node $w \in W^\ell_\rho$ such that $\mathcal{T} \models (\alpha \wedge \varphi)^{\rightarrow}(w)$. For $e_i := [w,i]$ we find that $e_i = d_j$ for all $(i,j) \in \rho$ and, by induction hypothesis, $\mathfrak{D}(\mathcal{T}) \models (\alpha \wedge \varphi)(\overline{e})$. Therefore $\mathfrak{D}(\mathcal{T}) \models \psi(\overline{d})$.

For (5), the claim is immediate from the induction hypothesis and from Lemma 5.3. ☐

Corollary 5.6. *If $\psi \in \mathrm{GSO}[\tau]$ is a sentence that is invariant under guarded bisimulation, then $\psi^{\rightarrow}(x)$ is bisimulation invariant on consistent $\tilde{\tau}$-trees.*

Proof. Let $\mathcal{T},\mathcal{T}'$ be two bisimilar, consistent $\tilde{\tau}$-trees. Then $\mathfrak{D}(\mathcal{T}) \sim_g \mathfrak{D}(\mathcal{T})$ by Proposition 4.8. It follows that $\mathcal{T} \models \psi^{\rightarrow}(\lambda) \Leftrightarrow \mathfrak{D}(\mathcal{T}) \models \psi \Leftrightarrow \mathfrak{D}(\mathcal{T}') \models \psi \Leftrightarrow \mathcal{T}' \models \psi^{\rightarrow}(\lambda')$. ☐

5.2. From the μ-calculus to guarded fixed point logic.

The translation back from the modal into the guarded world again requires some preliminary discussion. Every formula φ of the μ-calculus, evaluated on a tree $\mathcal{T} = \mathcal{T}(\mathfrak{B})$, defines the set $\varphi^{\mathcal{T}}$ of all nodes v such that $\mathcal{T}, v \models \varphi$. Recall how each node v of $\mathcal{T}$ represents a guarded list g_v in $\mathfrak{B}$ So the idea is to translate φ into a guarded formula $\psi(\overline{x})$ defining in $\mathfrak{B}$ the set $\psi^{\mathfrak{B}} = \{\overline{b} \in \mathfrak{B} : \mathfrak{B} \models \psi(\overline{b})\}$, which should be equal to $\{g_v \in \mathfrak{B} : \mathcal{T}, v \models \varphi\}$. The main problem is that guarded lists do not have a fixed length, so that φ must actually be translated into a tuple $(\varphi_0^{\leftarrow}, \ldots, \varphi_m^{\leftarrow})$ of formulae over $\mathfrak{D}$.

Definition 5.7. Let X be a monadic second-order variable. Then $X^{\leftarrow} := (X^{(0)}, \ldots, X^{(m)})$, where each X_i is an i-ary second-order variable. Let N be a set of nodes in a tree $\mathcal{T}(\mathfrak{B})$. The *representation* of N in $\mathfrak{B}$ is $N^{\leftarrow} := (N^{(0)}, \ldots, N^{(m)})$ where $N^{(i)} := \{g_v \in \mathfrak{B} : v \in N, |\mathfrak{A}_v| = i\} \subseteq B^i$.

Note that, since the root λ is the only node of size 0, either $N^{(0)} = \emptyset$ or $N^{(0)} = \{\lambda\}$ — we say that $N^{(0)} = \mathit{false}$ or $N^{(0)} = \mathit{true}$.

In the sequel we assume w.l.o.g. that L_μ formulae are written without ν-operators, that in any formula $\mu X.\xi$ the fixed point variable X occurs in ξ only inside the scope of modal operators, and even that in each $\varphi \in L_\mu$, the fixed point formulae $\mu X.\xi(X)$ themselves only occur inside the scope of modal operators, cf. [8].

The translation. For every formula $\varphi \in L_\mu[\tilde{\tau}]$ we now define formulae $\varphi_k^{\leftarrow}(x_1, \ldots, x_k) \in \mu\mathrm{GF}[\tau]$, one for each $k \le m$, in which each monadic second-order variable X of φ is represented by a tuple of second-order variables $X^{\leftarrow}$ (of arities $0, 1, \ldots, m$).

(1) If $\varphi = P_{\mathfrak{A}}$, then set $\varphi_k^{\leftarrow} := \mathit{false}$ if $|\mathfrak{A}| \neq k$, and otherwise let $\varphi_k^{\leftarrow}(x_1, \ldots, x_k)$ be the conjuction over all atomic and negated atomic τ-formulae α such that $\mathrm{free}(\alpha) \subseteq \{x_1, \ldots, x_k\}$ and $\mathfrak{A} \models \alpha(1, \ldots, k)$. Note that for $k = 0$, this gives $\varphi_k^{\leftarrow} = \mathit{true}$ if $\mathfrak{A} = \emptyset$ and $\varphi_k^{\leftarrow} = \mathit{false}$ otherwise.

(2) If $\varphi = X$, and $1 \le k \le m$, then $\varphi_k^{\leftarrow}(x_1, \ldots, x_k) := X^{(k)}(x_1, \ldots, x_k)$. We set $\varphi_0^{\leftarrow} := \mathit{false}$.

(3) Set $(\neg\varphi)_k^{\leftarrow} := \neg\varphi_k^{\leftarrow}$, $(\varphi \vee \vartheta)_k^{\leftarrow} := \varphi_k^{\leftarrow} \vee \vartheta_k^{\leftarrow}$ and $(\varphi \wedge \vartheta)_k^{\leftarrow} := \varphi_k^{\leftarrow} \wedge \vartheta_k^{\leftarrow}$, for all φ, ϑ and k.

(4) If $\varphi = \langle \ell, \rho \rangle \vartheta$, then $\varphi_k^{\leftarrow}(x_1, \ldots, x_k) := (\exists^{\neq} y_1, \ldots, y_\ell \,.\, \rho(\overline{x}, \overline{y}) \wedge \mathrm{guarded}(\overline{y}))\vartheta_\ell^{\leftarrow}(\overline{y})$ for all $k \le m$.

(5) If $\varphi = \mu X.\vartheta$, then $(\varphi_1^{\leftarrow}, \ldots, \varphi_m^{\leftarrow})$ is the following simultaneous least fixed point, which can be simulated by nested individual fixed points in the standard way:
$$\mu_{X^{\leftarrow}}[\vartheta_1^{\leftarrow}(y_1, X^{\leftarrow}), \ldots, \vartheta_m^{\leftarrow}(\overline{y}_m, X^{\leftarrow})],$$
where $\overline{y}_i = y_1, \ldots, y_i$. We set $\varphi_0 := \mathit{false}$.

Theorem 5.8. *Let $\mathfrak{B}$ be a τ-structure, let λ be the root of $\mathcal{T}(\mathfrak{B})$ and let ψ be a formula in $L_\mu[\tilde{\tau}]$. Then $\mathfrak{B} \models \psi_0^{\leftarrow}$ iff $\mathcal{T}(\mathfrak{B}), \lambda \models \psi$.*

Proof. Again we prove a more general statement involving free variables. W.l.o.g., consider the case of just one free monadic second-order variable Y. Let $\varphi(Y)$ be a formula in $L_\mu[\tilde{\tau}]$, N a set of nodes in $\mathcal{T} = \mathcal{T}(\mathfrak{B})$ with representation $N^{\leftarrow}$.

Claim. *$(\varphi_0(N^{\leftarrow})^{\mathfrak{B}}, \ldots, \varphi_m(N^{\leftarrow})^{\mathfrak{B}})$ is the representation of $\varphi(N)^{\mathcal{T}}$ in $\mathfrak{B}$.*

The claim is proved inductively. The cases corresponding to (1) — (3) are trivial.

Consider (4) and let $\varphi = \langle \ell, \rho \rangle \vartheta$. Suppose that $\mathcal{T}, v \models \varphi$. Then there is a node w such that $(v, w) \in E_\rho^\ell$ and $\mathcal{T}, w \models \vartheta$. Let $g_w = (c_1, \ldots, c_\ell)$; then $c_i = b_j$ for all $(i, j) \in \rho$ and, by induction hypothesis, $\mathfrak{B} \models \vartheta_k^{\leftarrow}(c_1, \ldots, c_\ell)$. Hence $\mathfrak{B} \models \varphi_k^{\leftarrow}(b_1, \ldots, b_k)$.

Conversely, suppose that $\mathfrak{B} \models \varphi_k^{\leftarrow}(b_1, \ldots, b_k)$. This means that there exists a guarded list $(c_1, \ldots, c_\ell)$ with $c_i = b_j$ for all $(i, j) \in \rho$ such that $\mathfrak{B} \models \vartheta_\ell^{\leftarrow}(c_1, \ldots, c_\ell)$. By the construction of $\mathcal{T}(\mathfrak{B})$, there exists a node w such that $(v, w) \in E_\rho^k$ and $g_w = (c_1, \ldots, c_\ell)$. The induction hypothesis implies that $\mathcal{T}, w \models \vartheta$. It follows that $\mathcal{T}, v \models \varphi$.

For (5) finally let $\varphi = \mu.X\vartheta(X)$. Consider the stages of the fixed point induction on φ, $\varphi^0 := \mathit{false}$, $\varphi^{\alpha+1} := \vartheta(\varphi^\alpha)$, and $\varphi^\beta := \bigvee_{\alpha<\beta} \varphi^\alpha$ for limit ordinals β. Let similarly, for the simultaneous fixed point induction on $\overline{\varphi} = (\varphi_1^{\leftarrow}, \ldots, \varphi_m^{\leftarrow})$, $\overline{\varphi}^0 := (\mathit{false}, \ldots, \mathit{false})$, $\overline{\varphi}^{\alpha+1} := (\vartheta_1(\overline{\varphi}^\alpha), \ldots, \vartheta_m(\overline{\varphi}^\alpha))$ and $\overline{\varphi}^\beta := \bigvee_{\alpha<\beta} \overline{\varphi}^\alpha$ (componentwise) in limits.

By induction hypothesis, if $M^{\leftarrow}$ is the representation of M in $\mathfrak{B}$, then $(\vartheta_0^{\leftarrow}(M^{\leftarrow}), \ldots, \vartheta_m^{\leftarrow}(M^{\leftarrow}))$ defines in $\mathfrak{B}$ the representation of $\vartheta(M)^{\mathcal{T}}$. By induction on α, $\overline{\varphi}^\alpha$ (together with $(\varphi_0^{\leftarrow})^\alpha := \mathit{false}$) defines in $\mathfrak{B}$ the representation of the set defined by φ^α in $\mathcal{T}$. Hence the same is true for the least fixed points. ☐

We are now in a position to prove our main theorem.

Theorem 5.9. *Every sentence in* GSO *that is invariant under guarded bisimulation is equivalent to a sentence in* μGF.

Proof. Let $\psi \in \mathrm{GSO}[\tau]$ be invariant under guarded bisimulation and let $\psi^{\to(z)}$ be its translation into $\mathrm{MSO}[\tilde{\tau}]$. By Lemma 5.6 $\psi^{\to(z)}$ is bisimulation-invariant on consistent trees. Recall that the consistency condition for trees can be formulated by a monadic second-order sentence γ, which is bisimulation invariant with respect to all trees. As a consequence, the formula $(\gamma \to \psi^{\to})(x)$ is bisimulation invariant on arbitrary trees. By the Janin-Walukiewicz Theorem, Theorem 5.1 above, there exists an equivalent formula φ in the μ-calculus. Let $\varphi_0^{\leftarrow}$ be its translation into $\mu\mathrm{GF}[\tau]$. Putting everything together, we have
$$\begin{aligned} \mathfrak{B} \models \psi \;&\Leftrightarrow\; \mathfrak{D}(\mathcal{T}(\mathfrak{B})) \models \psi \;\Leftrightarrow\; \mathcal{T}(\mathfrak{B}) \models \psi^{\to}(\lambda) \\ &\Leftrightarrow\; \mathcal{T}(\mathfrak{B}), \lambda \models \varphi \;\Leftrightarrow\; \mathfrak{B} \models \varphi_0^{\leftarrow}. \end{aligned}$$

The first equivalence uses guarded bisimulation invariance of ψ and Lemma 4.7; the second one is an application of Theorem 5.5; the third equivalence reflects the input from the Janin Walukiewicz Theorem; the fourth is an application of Theorem 5.8. $\square$

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] J. van Benthem. *Modal Logic and Classical Logic*. Bibliopolis, Napoli, 1983.

[3] H.-D. Ebbinghaus and J. Flum. *Finite Model Theory*. Springer, 1995.

[4] G. Gottlob, E. Grädel, and H. Veith. Datalog lite. Technical Report DBAI-TR-98-22, Vienna University of Technology, Institute for Information Systems, 1998.

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

[6] E. Grädel and I. Walukiewicz. Guarded fixed point logic. In *Proc. 14th IEEE Symp. on Logic in Computer Science*, 1999.

[7] D. Janin and I. Walukiewicz. On the expressive completeness of the propositional mu-calculus with respect to monadic second order logic. In *Proceedings of 7th International Conference on Concurrency Theory CONCUR '96*, number 1119 in Lecture Notes in Computer Science, pages 263–277. Springer-Verlag, 1996.

[8] D. Niwinski and I. Walukiewicz. Games for the mu-calculus. *Theoretical Computer Science*, 163:99–116, 1997.