Back and Forth Between Guarded and Modal Logics[1]

ERICH GRÄDEL and COLIN HIRSCH
University of Technology Aachen
and
MARTIN OTTO
University of Wales, Swansea

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. Guarded second-order logic, the fragment of second-order logic which is introduced in the context of our characterisation theorem, captures a natural and robust level of expressiveness with several equivalent characterisations. For a wide range of issues in guarded logics it may take up a role similar to that of monadic second-order in relation to modal logics. At the more general methodological level, the translations between the guarded and modal domains make the intuitive analogy between guarded and modal logics available as a tool in the further analysis of the model theory of guarded logics.

Categories and Subject Descriptors: F.4.1 [**Mathematical Logic**]: Mathematical Logic

General Terms: Theory

Additional Key Words and Phrases: Bisimulation, guarded logic, modal logic, model theory

1. INTRODUCTION

Guarded logics generalise certain features and desirable model theoretic properties of modal logics to a more general setting. The concept of guarded quantification was introduced by Andréka, van Benthem, and Németi [Andréka et al. 1998], who proposed and analysed the *guarded fragment of first-order logic*, GF. This fragment GF provides a very satisfactory basis for the explanation of the good behaviour of modal logics at the level of a rich fragment of classical first-order logic. Moreover the robust decidability properties [Vardi 1997; Grädel 2001b] of modal logics (in particular the good algorithmic properties of logics like CTL or the modal μ-calculus that

[1]An extended abstract of this paper previously appeared as [Grädel et al. 2000]

extend propositional modal logic by second-order features and/or natural recursion mechanisms) are also reflected in GF. Most notably, not only is GF decidable itself [Andréka et al. 1998], but so is its canonical fixed point extension, μGF [Grädel and Walukiewicz 1999]. μGF extends GF so as to render definable least and greatest fixed points of GF definable positive operators of arbitrary arities. In particular it extends the modal μ-calculus to the guarded domain. Unlike its modal companion, however, it no longer has 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, are efficient algorithmic techniques for model checking and query evaluation. The alternation-free fragment of μGF admits linear time model checking [Gottlob et al. pear] and game based techniques provide efficient evaluation algorithms also for other interesting fragments [Berwanger and Grädel 2001].

These and other results of recent research into guarded logics indicate that GF and its relatives provide a very interesting domain of logics, with a good balance between expressive power and algorithmic manageability.

From a model theoretic perspective, *modal logics* — in a broad sense, including extensions like CTL and the modal μ-calculus — are characterised by their *invariance under bisimulation*, i.e. they cannot distinguish between bisimilar structures. Moreover, invariance under bisimulation is at the root of many algorithmically important model theoretic properties 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 taken up by the similar, but much more wide-ranging and finer notion of equivalence induced by *guarded bisimulation* in the guarded domain.

We consider both modal and guarded logics – as well as classical first-order and second-order logic – in the setting of relational structures. For modal logics we therefore appeal to their semantics over transition systems, regarded as relational structures with binary transition relations and unary predicates for basic propositions, while guarded logics deal with structures of arbitrary relational type.

The characteristic feature of modal quantification is that one can only access nodes along basic transition relations. In the guarded scenario, this is generalised to 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 meaningful over structures with higher-arity relations and can address properties of tuples. Also, guarded quantification has no preferred sense of direction (in modal parlance: inverse modalities are automatically included); guarded quantification is not local, in the sense that it allows jumps from one guarded tuple to any other guarded tuple, overlapping or not (in modal parlance: universal modalities are automatically included); and guarded logics naturally have equality.

The main technical theme of this paper is a reduction from the richer scenario of guarded logics to the simpler and better understood scenario of modal logics, or from guarded bisimulation to ordinary bisimulation. Corresponding ideas have been explored in [Grädel 2001a] in the context of the satisfiability problem for guarded fixed point logic. This methodology is carried much further here and applied to a semantic characterisation of guarded fixed point logic within a suitable second-order framework.

Characterisation theorems of this type have a strong tradition in the field. At the level of basic modal logic and the guarded fragment of first-order logic, the corresponding characterisation theorems are the following, see [van Benthem 1983] and [Andréka et al. 1998].

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. *A property of relational structures is definable in the guarded fragment* GF *if and only if it is 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 [Janin and Walukiewicz 1996].

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* MSO *and invariant under bisimulation.*

Johan van Benthem has raised the question whether μGF admits a similar characterisation, in terms of guarded bisimulation, 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. Moreover, the resulting fragment is no longer decidable (see Proposition 3.11).

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.

It should be noted that GSO properly extends monadic second-order logic, and in particular includes all of first-order logic. Our main result is the following theorem.

Theorem 1.4. *A property of relational structures is definable in guarded fixed point logic* μGF *if and only if it is definable in guarded second-order logic* GSO *and invariant under guarded bisimulation.*

We note that, of the above mentioned characterisation theorems, only the characterisation of basic modal logic is known to survive as a theorem of finite model theory; the corresponding analogue of Theorem 1.1 is due to Eric Rosen [Rosen 1997], also compare [Otto]. The status in finite model theory of the other ones, including our main theorem, is open.

The precise definition of GSO with some further discussion is given in Section 3.4. There are several equivalent syntactic characterisations. In particular, it turns out that the same level of expressiveness is attained by extending GF by full second-order quantification. Another characterisation of GSO highlights the relation with MSO. Over ordinary relational structures GSO can express exactly what can be expressed in MSO over the many-sorted presentation of relational structures by incidence graphs (see Section 7.1).

THEOREM 1.5. *A property of relational structures is definable in* GSO *if and only if the corresponding property of their incidence graphs is definable in* MSO.

The relationship between the expressive power of MSO on ordinary relational encodings versus incidence encodings has long been studied for (many important classes of) graphs, see in particular the work of Courcelle [Courcelle 1997]. In view of our result that MSO over incidence graphs can be pulled back to GSO over ordinary relational structures, this question precisely concerns the relationship between MSO and GSO.

The structure of the paper is as follows. Section 2 fixes basic terminology and reviews some relevant facts from the world of modal logics. Section 3 gives an introduction to the world of guarded logics; here we also define guarded second-order logic and discuss its expressive power. Sections 4 and 5 form the methodological core of the paper: we establish two-way links between the two worlds, associating relational structures with transition systems or trees, guarded bisimulation with bisimulation, and guarded second-order logic with monadic second-order logic; this leads up to the proof of our characterisation theorem, Theorem 1.4 above, for sentences. The ramification of our technique to formulae with free variables and structures with parameters is discussed in Section 6; these considerations support the stronger reading of Theorem 1.4 comprising formulae with free variables. Section 7.1 compares guarded second-order logic to monadic second-order logic and proves Theorem 1.5 and Section 7.2 finally considers guarded logics on graphs and transition systems, the natural playground of modal logics.

2. PRELIMINARIES

2.1 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. We regard trees as special transition systems.

Definition 2.1. A transition system $\mathcal{T} = (V, (E_a)_{a\in A}, (P_b)_{b\in B})$ is a *tree* if the E_a are mutually disjoint and their union $E = \bigcup_{a\in A} E_a$ forms a directed tree in the graph theoretic sense.

Bisimulation equivalence between transition systems not only captures the natural notion of behavioural equivalence between processes but also corresponds to the natural notion of Ehrenfeucht-Fraïssé equivalence associated with modal quantification.

Definition 2.2. 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 $\mathcal{K}, \mathcal{K}'$ with distinguished nodes $u \in \mathcal{K}$ and $u' \in \mathcal{K}'$ are *bisimilar*, denoted $\mathcal{K}, u \sim \mathcal{K}', u'$, 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 $\mathcal{T}, \lambda \sim \mathcal{T}', \lambda'$.

A crucial model theoretic feature of modal logics is the *tree model property*, the fact that every satisfiable formula is satisfiable in a tree. This is a well known and straightforward consequence of bisimulation invariance, since every transition system $(\mathcal{K}, u)$ 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 $\mathcal{K}$ with obvious interpretations of atomic propositions and accessibility relations. Unravellings are important as tree-like bisimilar companions, $\mathcal{T}(\mathcal{K}, u), u \sim \mathcal{K}, u$. As far as bisimulation invariant properties are concerned, they allow us to restrict attention to trees.

Definition 2.3. 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
$\mathcal{T}(\mathcal{K}, u) = (V^{\mathcal{T}}, (E_a^{\mathcal{T}})_{a\in A}, (P_b^{\mathcal{T}})_{b\in B})$, where

(i) $V^{\mathcal{T}}$ is the set of all sequences $\overline{v} = v_0 a_1 v_1 a_2 \cdots v_{r-1} a_r v_r$ where $v_i \in V$, $a_i \in A$ such that $v_0 = u$ and $(v_{i-1}, v_i) \in E_{a_i}$;

(ii) $P_b^{\mathcal{T}}$ contains the sequences $v_0 a_1 v_1 a_2 \cdots v_{r-1} a_r v_r$ with $v_r \in P_b$;

(iii) $E_a^{\mathcal{T}}$ contains the pairs $(\overline{v}, \overline{v}av)$ in $V^{\mathcal{T}} \times V^{\mathcal{T}}$.

The *natural projection* $\pi \colon \mathcal{T}(\mathcal{K}, u) \to \mathcal{K}, u$ sends $\overline{v} = v_0 a_1 v_1 a_2 \cdots v_{r-1} a_r v_r \in V^{\mathcal{T}}$ to its last node $\pi(\overline{v}) = v_r \in V$.

Note that a canonical bisimulation between $\mathcal{T}(\mathcal{K}, u)$ and $\mathcal{K}, u$ is induced by the graph of the natural projection.

Trees are special as bisimilar companions to arbitrary transition systems also because bisimulation equivalence between trees comes with an additional feature. A *two-way bisimulation* is one that not only satisfies the back and forth conditions for the transition relations E_a themselves, but also for their inverses $E_a^{-1} = \{(u, v) : (v, u) \in E_a\}$. In general, two-way bisimulation is a much stronger notion than ordinary bisimulation; not so, however, for trees.

LEMMA 2.4. *Bisimilar trees are two-way bisimilar.*

PROOF. Let $\mathcal{T}$ and $\mathcal{T}'$ be bisimilar trees with roots λ and λ', respectively. We inductively construct a two-way bisimulation $Z = \bigcup_{n<\omega} Z_n$ between $\mathcal{T}$ and $\mathcal{T}'$. The *depth* of a node is its distance from the root. Each Z_n will associate nodes of depth n in $\mathcal{T}$ with nodes of depth n in $\mathcal{T}'$. Let $Z_0 = \{(\lambda, \lambda')\}$. For $n > 0$, let $(v, v') \in Z_n$ if v and v' have depth n and

(1) $\mathcal{T}, v \sim \mathcal{T}', v'$ (with respect to usual (forward) bisimulation), and
(2) the edges to v and v' from their parent nodes u and u' have the same label, and $(u, u') \in Z_{n-1}$.

Obviously Z is a two-way bisimulation. □

2.2 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.
(i) Each atomic proposition P_b is a formula.
(ii) If ψ and φ are formulae of ML, then so are $(\psi \wedge \varphi)$, $(\psi \vee \varphi)$ and $\neg\psi$.
(iii) 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

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

The $\boldsymbol{\mu}$-calculus $\mathbf{L}_{\boldsymbol{\mu}}$. The propositional μ-calculus L_μ is propositional modal logic augmented with least and greatest fixed points. It was introduced in [Kozen 1983], and has in the meanwhile been established as *the* fundamental modal fixed point logic, subsuming almost all of the commonly used modal logics used in verification and reasoning about programs, in particular the temporal logics LTL, CTL, CTL*, dynamic logic PDL [Harel et al. 2000], and also many logics used in other areas of computer science, for instance description logics [Donnini et al. 1996]. For more information on the μ-calculus, we refer to [Arnold and Niwiński 2001; Bradfield and Stirling 2001] and the references there.

Syntax of L_μ. The μ-calculus extends propositional modal logic ML (including propositional variables $X, Y, \ldots$, which are 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 $\psi^{\mathcal{K}} : \mathcal{P}(V) \to \mathcal{P}(V)$ assigning to every set $X \subseteq V$ the set

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

If 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 $\psi^{\mathcal{K}}$,

$\mathcal{K}, v \models \nu X.\psi$ iff v is an element of 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 sometimes be more convenient to keep least and greatest fixed points and 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 $\overline{\psi} = (\psi_1, \ldots, \psi_k)$ of formulae $\psi_i(\overline{X}) = \psi_i(X_1, \ldots, X_k)$, 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 $\varphi^{\mathcal{K}}$ mapping $\overline{X}$ to $\overline{X}'$ where $X'_i = \{v \in V : (\mathcal{K}, \overline{X}), v \models \psi_i\}$. By convention $\mathcal{K}, v \models \varphi$ 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 nested individual fixed points. This is based on a general fact concerning fixed points of monotone operators which is sometimes called the Bekic principle (see e.g. [Arnold and Niwiński 2001]) and which applies to least fixed point logics in general. In its simplest form for the μ-calculus, it says that

$$\mu XY.(\psi(X,Y), \varphi(X,Y)) \equiv \mu X.\psi(X, \mu Y.\varphi(X,Y)).$$

It is obvious that all formulae of ML and L_μ are invariant under bisimulation (see e.g. [Blackburn et al. 2001; Janin and Walukiewicz 1996]).

3. GUARDED FIRST AND SECOND-ORDER LOGICS

Definition 3.1. Let $\mathfrak{B}$ be a relational 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 τ-structure $\mathfrak{A}$ (in particular a substructure $\mathfrak{A} \subseteq \mathfrak{B}$) is *guarded* if its universe is a guarded set in $\mathfrak{A}$ (in $\mathfrak{B}$).

(iii) A tuple $(b_1, \dots, b_n) \in B^n$ is *guarded* in $\mathfrak{B}$ if $\{b_1, \dots, b_n\} \subseteq X$ for some guarded set $X \subseteq B$.
(iv) A tuple $(b_1, \dots, b_k) \in B^k$ is a *guarded list* in $\mathfrak{B}$ if its components are pairwise distinct and $\{b_1, \dots, b_k\}$ is a guarded set. We admit the empty list $\emptyset$ as a guarded list.
(v) 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 $b = b$. Correspondingly, any monadic relation is a guarded relation according to (v). 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 sets; note that unlike guarded tuples they list a full guarded set, not just a subset of a guarded set.

Example 3.2. Let $\mathfrak{A}$ be the structure with universe $A = \{1, 2, 3, 4\}$ and the ternary relation $R = \{(1, 2, 3), (4, 2, 3)\}$. The substructure of $\mathfrak{A}$ induced by $\{2, 3, 4\}$ is a guarded structure, but $\mathfrak{A}$ itself is not. The tuples $(3, 2, 1)$ and $(2, 3, 4)$ are guarded tuples and guarded lists, $(1, 1)$ and $(4, 4, 3, 2)$ are guarded tuples but not guarded lists and $(4, 3, 2, 1)$ is neither.

3.1 The guarded fragment GF of first-order logic

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. 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 $\psi(\overline{x}, \overline{y})$ is a formula of GF and $\alpha(\overline{x}, \overline{y})$ is a τ-atom such that all free variables of ψ do actually occur in α 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. When writing $\alpha(\overline{x}, \overline{y})$ for the guard and $\psi(\overline{x}, \overline{y})$ for the kernel that is being quantified, we implicitly appeal to the convention that the free variables of ψ are *among* those displayed and that *all* variables displayed in the atom α actually occur in α.

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

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

It is obvious that GF generalises ML. A typical modal formula $\varphi = [1]\langle 2\rangle P$, rendered in first-order terms as $\varphi(x) = \forall y\big(E_1xy \rightarrow \exists z(E_2yz \wedge Pz)\big)$, is clearly in GF. The following are some simple examples of GF formulae that are not in ML (even though in a language of transition systems): $\forall y(E_1yx \rightarrow (E_2xy \wedge y \neq x))$, $\forall x\big(x = x \rightarrow \exists y(Exy \wedge \neg Eyx)\big)$.

The following example will also be of technical use later.

Example 3.3. The set of all guarded k-tuples in structures of vocabulary $\tau = \{R_1, \dots R_t\}$ is defined by the formula

$$\text{guarded}(x_1, \dots, x_k) := \bigvee_{i=1}^{t} \exists \overline{y} \Big(R_i \overline{y} \wedge \bigwedge_{\ell=1}^{k} \bigvee_{j} x_\ell = y_j \Big).$$

Note that this formula is *not* in GF since the variables $x_1, \dots, x_k$ do not occur in the atoms $R_i \overline{y}$. However it is not difficult to construct an equivalent guarded formula. For any complete equality type on $\{x_1, \dots, x_k\}$ specified by a quantifier-free formula $\eta(\overline{x})$ in the language of just $=$, let $\overline{x}^\eta$ be a 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 and the $\overline{y}$ are new, i.e. disjoint from $\overline{x}$. 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 $x_1 = x_2 = \dots = x_k$, 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.

Likewise, we note that quantifications of the form $\exists \overline{x}(\text{guarded}(\overline{x}) \wedge \varphi(\overline{x}))$ and $\forall \overline{x}(\text{guarded}(\overline{x}) \rightarrow \varphi(\overline{x}))$, though not strictly in GF, may equivalently be expressed in GF syntax. We shall sometimes use such liberalised formalisations for the sake of clarity and convenience.

Definition 3.4. We say that a formula $\varphi(\overline{x})$ is *variable-guarded* if it implies guardedness of the tuple $\overline{x}$ of free variables, i.e., if $\varphi(\overline{x})$ is logically equivalent to $\varphi(\overline{x}) \wedge \text{guarded}(\overline{x})$.

As GF imposes no restrictions at the quantifier free level or on Boolean combinations, formulae of GF are not in general variable guarded. However, any formula of GF is equivalent to a Boolean combination of variable-guarded GF formulae. To see this, note that any GF formula is a Boolean combination of atomic statements, which are trivially variable-guarded, and formula whose top-level syntactic element is a guarded quantification – thereby providing guards for their respective free variables.

3.2 Guarded fixed point logic μGF

Guarded fixed point logic μGF as introduced in [Grädel and Walukiewicz 1999] 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 $X, Y, Z, \dots$ that are treated like predicates in τ but may *not* be used in guards, we augment the syntax by the following rule for building fixed point formulae.

> Let X be a k-ary relation variable and let $\overline{x} = x_1, \dots, x_k$ be a k-tuple of distinct variables. Further, let $\psi(X, \overline{x})$ be a formula with only positive occurrences of X, where X is not used in guards and where all free variables of ψ are among these x_i. Then also $[\mathbf{lfp}\ X\overline{x} \,.\, \psi](\overline{x})$ and $[\mathbf{gfp}\ X\overline{x} \,.\, \psi](\overline{x})$ are formulae of μGF.

Semantics of μGF. The semantics of $[\mathbf{lfp}\ X\overline{x} \,.\, \psi](\overline{x})$ 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,

$\mathfrak{B} \models [\mathbf{lfp}\ X\overline{x}\,.\,\psi](\overline{b})$ iff $\overline{b}$ is an element of the least fixed point of the operator $\psi^{\mathfrak{B}}$. Similarly for $[\mathbf{gfp}\ X\overline{x}\,.\,\psi](\overline{x})$ and the greatest fixed point of $\psi^{\mathfrak{B}}$.

One may also admit *simultaneous* least and greatest fixed points in μGF. Given formulae $\psi_1(X_1,\dots,X_k,\overline{x}_1),\dots,\psi_k(X_1,\dots,X_k,\overline{x}_k)$, with only positive occurrences of the relation variables $X_1,\dots,X_k$ and where the $\overline{x}_i$ contain all free first-order variables in ψ_i and match the arity of the X_i,

$$S := \begin{cases} X_1\overline{x}_1 & := \ \psi_1(X_1,\dots,X_k,\overline{x}_1) \\ & \ \ \vdots \\ X_k\overline{x}_k & := \ \psi_k(X_1,\dots,X_k,\overline{x}_k) \end{cases}$$

is a *system of fixed point equations*, and $[\mathbf{lfp}\ X_i\,.\,S](\overline{x}_i)$ and $[\mathbf{gfp}\ X_i\,.\,S](\overline{x}_i)$ are formulae of μGF.

On every structure $\mathfrak{A}$, the system S defines a monotone operator $S^{\mathfrak{A}}$ on k-tuples of relations, with least fixed point $\mathbf{lfp}(S^{\mathfrak{A}}) = (X_1^{\mu},\dots,X_k^{\mu})$ and greatest fixed point $\mathbf{gfp}(S^{\mathfrak{A}}) = (X_1^{\nu},\dots,X_k^{\nu})$. Now $\mathfrak{A} \models [\mathbf{lfp}\ X_i\,.\,S](\overline{a})$ iff $\overline{a} \in X_i^{\mu}$ and $\mathfrak{A} \models [\mathbf{gfp}\ X_i\,.\,S](\overline{a})$ iff $\overline{a} \in X_i^{\nu}$.
Again, the Bekic principle implies that simultaneous fixed points can be eliminated in μGF.

It should be stressed that the presence of extra first-order parameters in fixed point operations as well as the use of second-order variables and fixed points as guards is disallowed in μGF. These restrictions are essential for keeping the semantics invariant under guarded bisimulation. For instance, with the use of a first-order parameter (here x), one can define the transitive closure of a binary relation E by the formula $\varphi(x,y) := [\mathbf{lfp}\ Xy\,.\,Exy \vee (\exists z.Ezy)Xz](y)$. However, the transitive closure query is *not* invariant under guarded bisimulation and it is known that adding transitive closure to GF produces an undecidable logic [Grädel 1999]. Allowing fixed-point variables in guards would make full first-order quantification available. For instance, $\exists y\varphi(x,y) \equiv [\mathbf{gfp}\ Xxy\,.\,(\exists y.Xxy)\varphi(x,y)](x,x)$.

A seemingly more restrictive variant of guarded fixed point logic would only allow least and greatest fixed points over variable-guarded formulae in the sense of Definition 3.4 above, or fixed points like $[\mathbf{lfp}\ X\overline{x}\,.\,\varphi(X,\overline{x}) \wedge \mathrm{guarded}(\overline{x})](\overline{x})$. We refer to these as *strictly guarded fixed points*. Strictly guarded least and greatest fixed points can obviously only define guarded relations. A restriction of μGF to strictly guarded fixed points, however, does not diminish its expressive power. This claim is easily verified for sentences and variable-guarded formulae, see [Grädel and Walukiewicz 1999]. Formulae with arbitrary tuples of free first-order variables, however, require some extra attention. See Section 6.3 for the proof of the following proposition.

PROPOSITION 3.5. *Any formula of* μGF *is logically equivalent to one in which all fixed points are strictly guarded.*

3.3 Guarded bisimulation

A fundamental tool for analysing guarded logics are guarded bisimulations, introduced in [Andréka et al. 1998].

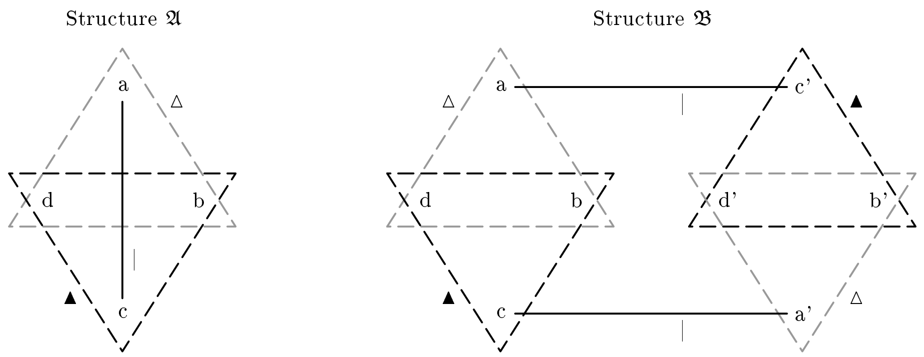

Fig. 1. Sketches of $\mathfrak{A}$ and $\mathfrak{B}$

Definition 3.6. 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 $\mathfrak{B}$, such that the following back and forth conditions are satisfied.

Forth: For every $f : X \to Y$ in I and 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 $f : X \to Y$ in I and 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. Two τ-structures with parameter tuples $\mathfrak{A}, \overline{a}$ and $\mathfrak{B}, \overline{b}$ are *guarded bisimilar* ($\mathfrak{A}, \overline{a} \sim_g \mathfrak{B}, \overline{b}$), if there is a guarded bisimulation I between $\mathfrak{A}$ and $\mathfrak{B}$ containing the partial isomorphism $f : \overline{a} \mapsto \overline{b}$.

Note that, with the possible exception of the one partial isomorphism dealing with a given pair of parameter tuples, one may restrict attention to systems of partial isomorphisms whose domain and range are guarded sets in $\mathfrak{A}$ and $\mathfrak{B}$, respectively.

We refer to the relation $\sim_g$ as *guarded bisimulation equivalence*.

Example 3.7. Let $\mathfrak{A}$ and $\mathfrak{B}$ be two structures as depicted in Figure 1, with two ternary relations $\vartriangle$ and $\blacktriangle$ and one binary relation $|$ (all assumed to be symmetric as in hypergraph theory).

A guarded bisimulation $I : \mathfrak{A} \sim_g \mathfrak{B}$ is given by $I = \{f_\vartriangle : abd \mapsto abd,\ f'_\vartriangle : abd \mapsto a'b'd'$ $f_\blacktriangle : cbd \mapsto cbd,\ f'_\blacktriangle : cbd \mapsto c'b'd',\ f_| : ac \mapsto ac',\ f'_| : ac \mapsto a'c\}$, plus all subfunctions of the ones listed.

The reader is encouraged to check the details to get more familiar with the definition.

Definition 3.8. 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 $\mathfrak{B} \models \psi$. Similarly, a formula $\varphi(\overline{x})$ is invariant under guarded bisimulation if whenever $\mathfrak{A}, \overline{a} \models \varphi(\overline{x})$ and $\mathfrak{A}, \overline{a} \sim_g \mathfrak{B}, \overline{b}$ we also have $\mathfrak{B}, \overline{b} \models \varphi(\overline{x})$. A logic L is invariant under guarded bisimulation if all L-formulae are invariant under guarded bisimulation.

It suffices to look at variable-guarded formulae when checking a logic like GF or μGF for invariance under guarded bisimulation, as any formula is equivalent to a Boolean combination of variable-guarded formulae. For these the following are easily proved by syntactic induction, cf. also [Andréka et al. 1998; Grädel 2001a].

Proposition 3.9. *GF and μGF are invariant under guarded bisimulation.*

3.4 Guarded second-order logic GSO

We now introduce a guarded variant of second-order logic, denoted GSO. There are several semantically equivalent definitions. In our preferred definition of GSO 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* in the sense of Definition 3.1 (v), rather than over arbitrary relations.

Definition 3.10. *Guarded second-order logic* GSO is second-order logic with *guarded semantics* for the second-order quantifiers: all second order quantifiers range over guarded relations (i.e., collections of guarded tuples).

The definition is motivated by the idea that GSO should relate to GF and μGF in the same way as MSO relates to ML and L_μ. Since μGF has fixed-point definitions (i.e. second-order constructs) for relations of any arity, MSO itself is clearly not the right logic. On the other side, as already mentioned in the introduction, full second-order logic is far too powerful (no matter whether we consider invariance under bisimulation or under guarded bisimulation).

Proposition 3.11. *There exist second-order sentences that are invariant under guarded bisimulation and not equivalent to any sentence in μGF (or GSO). Further, the satisfiability problem for (guarded) bisimulation-invariant second-order logic is undecidable.*

Proof. We first observe that bisimilarity is a second-order property. For each finite vocabulary, there exists a second-order formula $\text{g-sim}(x, y)$ such that $\mathfrak{A} \models \text{g-sim}(a, a')$ iff $\mathfrak{A}, a \sim_g \mathfrak{A}, a'$. Now, on the vocabulary $\{E, \text{red}\}$ of coloured graphs, let ψ be the second-order sentence

$$\forall x \forall y \big(\text{red}(x) \wedge \text{red}(y) \rightarrow \text{g-sim}(x, y)\big).$$

Note that ψ is invariant under guarded bisimulation. Indeed suppose that $\mathfrak{A} \models \psi$ and $\mathfrak{A} \sim_g \mathfrak{B}$. Let b, b' be red nodes in $\mathfrak{B}$. By the back-condition of guarded bisimulations, there exist red nodes a, a' in $\mathfrak{A}$ such that $\mathfrak{A}, a \sim_g \mathfrak{B}, b$ and $\mathfrak{A}, a' \sim_g \mathfrak{B}, b'$. As $\mathfrak{A}, a \sim \mathfrak{A}, a'$ it follows that $\mathfrak{B}, b \sim_g \mathfrak{B}, b'$. Hence $\mathfrak{B} \models \psi$. (On trees, ψ is even invariant under bisimulation.)

To show that ψ is not equivalent to any GSO-sentence it suffices to consider ψ on $\mathcal{T} = (V, E, \text{red})$ where (V, E) is the complete binary tree. Clearly, $\mathcal{T} \models \psi$ if and only if all red nodes in $\mathcal{T}$ are on the same level. But on trees GSO and MSO coincide (see Section 7.1 below for a more general result), and it is well-known that the equal level predicate on the binary tree is not MSO-definable (see e.g. [Thomas 1990, p. 183]).

For the second claim we assume that the reader is familiar with the method of proving undecidability via domino problems (see e.g. [Börger et al. 1997]). With each domino system $\mathcal{D}$ one associates a formula which is satisfiable if and only if $\mathcal{D}$ admits a tiling of the infinite grid $(\mathbb{N} \times \mathbb{N}, H, V)$. Here we have to do this in second-order logic in a way that respects bisimilarity. Given a domino system $\mathcal{D}$ we construct a second-order sentence $\psi_{\mathcal{D}}$ with binary relation H and V and unary relations $\overline{D}$ (one for each domino) that expresses the following:

—Each point has an H-successor and a V-successor.

—If $x \sim_g y$ then all H-successors of x are bisimilar to all H-successors of y. Similarly for V.

—For each point x, its H-V-successors are bisimilar to its V-H-successors.

—Each point carries precisely one tile, and the horizontal and vertical adjacency conditions imposed by $\mathcal{D}$ are respected.

The models of $\psi_{\mathcal{D}}$ are precisely the transition systems that are bisimilar to an admissible tiling $(\mathbb{N} \times \mathbb{N}, H, V, \overline{D})$ of the infinite grid by $\mathcal{D}$. In particular $\psi_{\mathcal{D}}$ is invariant under bisimulation, and the satisfiability problem for the class of these sentences is undecidable. $\square$

We will show that the expressive power of GSO is between MSO and SO, and that GSO has the desired properties. Further, GSO defines a very robust level of expressiveness which is invariant 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; at no point will second-order variables be admitted as guards.

Note that the semantic restriction of second-order quantification in GSO may alternatively be captured purely syntactically, for instance by only allowing occurrences of atoms $X\overline{x}$ in conjunction with the GF-formula guarded$(\overline{x})$, which says that $\overline{x}$ is a guarded tuple, thus effectively restricting X to its guarded part.

GSO includes full first-order logic. Hence GSO is undecidable and, unlike GF and μGF, not invariant under guarded bisimulation. 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 3.14 below that GSO collapses to MSO over words. The robustness of GSO is underlined by the following.

Proposition 3.12. *The following fragments of second-order logic are equally expressive:*

(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).

From (1) to (2). We show how to replace any second-order quantifier that ranges over arbitrary relations by second-order quantifiers that range over guarded relations. Let $\psi(X, \overline{x})$ be a formula according to (1), in which X is a k-ary second-order variable that does not occur as a guard, and consider the formula $\exists X \psi(X, \overline{x})$

in the sense of ordinary unconstrained semantics for the second-order quantification. Let the free first-order variables of ψ be $\overline{x} = (x_1, \ldots, x_n)$. We will modify ψ in a way that each atomic statement $X(\overline{y})$ is either eliminated or replaced by $X(\overline{y}) \land \mathrm{guarded}(\overline{y})$.

For any atom $X(\overline{y})$ in ψ within the scope of a first-order quantifier the definition of GF ensures that the variables $\overline{y}$ are included in the guard statement of that quantification[2]. These occurrences can therefore be replaced by $X(\overline{y}) \land \mathrm{guarded}(\overline{y})$ without changing the semantics of ψ.

On the other hand the quantifier-free (w.r.t. first-order quantification) part of ψ may depend on truth values of X for unguarded tuples, but with the restriction that the elements of these tuples are among the valuation of $\overline{x}$. This gives us a fixed finite number of unguarded tuples relevant to ψ so that the truth value of X for these tuples can be explicitly enumerated. We formalise this as follows. Let $H = H(X, \overline{x})$ be the set of all $\{X\}$-structures with universe $\overline{x}$. Replace $\exists X \psi(X, \overline{x})$ by $\exists X \bigvee_{\mathfrak{H} \in H} \psi^{\mathfrak{H}}$, where $\psi^{\mathfrak{H}}$ is obtained from ψ by replacing all atoms $X(\overline{y})$ where $\overline{y} \subseteq \overline{x}$ with $(\mathrm{guarded}(\overline{y}) \land X(\overline{y})) \lor \neg\,\mathrm{guarded}(\overline{y})$, if $\mathfrak{H} \models X(\overline{y})$, and $(\mathrm{guarded}(\overline{y}) \land X(\overline{y}))$, otherwise.

The modified formula only contains occurrences of X that depend on the guarded part of X. They are guarded either explicitly through conjunction with $\mathrm{guarded}(\cdot)$, or implicitly, because they are in the scope of a guarded quantification. Any unrestricted valuation for X that satisfies $\psi(X, \overline{x})$ gives rise to an X' which is the guarded part of X, and an $\mathfrak{H}$ that coincides with X on all atomic statements built with elements from $\overline{x}$, in a way that X' satisfies $\psi^{\mathfrak{H}}(X', \overline{x})$, and vice versa.

From (3) to (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.

$$\begin{aligned}
x = y &\mapsto \forall x(Xx \leftrightarrow Yx)\\
R\overline{x} &\mapsto (\exists \overline{x} \,.\, R\overline{x}) \textstyle\bigwedge_i X_i x_i\\
Z\overline{x} &\mapsto (\exists \overline{x} \,.\, \mathrm{guarded}(\overline{x})) \textstyle\bigwedge_i X_i x_i \land Z\overline{x}\\
\exists x \varphi(x, \overline{y}) &\mapsto \exists X \big(\varphi(X, \overline{y}) \land \mathrm{singleton}(X)\big),
\end{aligned}$$

where $\mathrm{singleton}(X)$ is a formula stating that X contains exactly one element:

$$\mathrm{singleton}(X) := \exists x Xx \land \forall Y \big((\forall x(Yx \to Xx) \to (\forall x\, \neg Yx \lor (\forall x\, Yx \leftrightarrow Xx))\big).$$

Note that these translations and in particular $\mathrm{singleton}(X)$ are in GF, since first-order quantification over a single free first-order variable x is always guarded (by $x = x$). □

[2]More precisely, at least the guard statement of the *innermost* quantification above $X(\overline{y})$ includes all $\overline{y}$.

Example 3.13. Hamiltonicity of graphs is GSO-definable as follows.

$$\exists H \ \big(\forall x \forall y (Hxy \rightarrow Exy) \ \wedge \ \forall x \, (\exists^{=1} y \, Hxy \ \wedge \ \exists^{=1} y \, Hyx) \wedge \\ \forall X \, \big[\big(\exists x \, Xx \wedge \forall x \forall y (Hxy \wedge Xx \rightarrow Xy)\big) \rightarrow \forall x \, Xx \big] \big)$$

Evaluated on a graph $G = (V, E)$ the formula says that there exists an $H \subseteq E$ with unique successors and predecessors such that (V, H) is connected. This means that G has a Hamilton cycle.

As Hamiltonicity is known not to be expressible in MSO (see [Ebbinghaus and Flum 1999]), the example shows that GSO is more expressive than MSO. In fact, GSO lies strictly between MSO and SO.

LEMMA 3.14. GSO *collapses to* MSO *over words.*

PROOF. As usual we represent words $w = w_0 \cdots w_{n-1} \in B^*$ by word structures $\big(\{0, \ldots, n-1\}, S, (P_b)_{b \in B}\big)$ where $S = \{(i, i+1) : i < n-1\}$ and P_b is the set of positions in the word carrying the letter b, i.e., $P_b = \{i < n : w_i = b\}$. The predicate of maximal arity in a word structure is the successor relation, so a guarded set is either a singleton or a set $\{i, i+1\}$. A guarded n-ary relation therefore contains only n-tuples $(a_1, \ldots, a_n)$ such that $\{a_1, \ldots, a_n\} \subseteq \{i, i+1\}$ for some i and can therefore be encoded by a sequence of monadic relations. For instance a guarded k-ary relation X can be represented by $\{X_u : u \in \{0,1\}^k\}$, where for each $u = u_0 \ldots u_{k-1}$, $X_u := \{i < k-1 : (i + u_0, \ldots, i + u_{k-1}) \in X\}$. Hence GSO is no 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 (which is of course much larger than the class of regular languages). $\Box$

COROLLARY 3.15. SO *is strictly more expressive than* GSO.

To summarise, we have the following hierarchy of logics.

PROPOSITION 3.16. ML $\subsetneq$ GF $\subsetneq$ FO $\subsetneq$ MSO $\subsetneq$ GSO $\subsetneq$ SO.

Relationships between GSO and MSO are further explored in Section 7.1 below. In particular we shall see that a collapse, as exhibited over words in the proof of the previous Lemma, occurs over some important classes of graphs.

Unlike MSO, GSO is sufficiently expressive to capture μGF. By Proposition 3.5, we may w.l.o.g. consider μGF formulae whose fixed point applications are strictly guarded so that these fixed points are themselves guarded relations. It is clear that such fixed points are definable within GSO by means of the usual second-order characterisation of least and greatest fixed points:

$$[\mathbf{lfp}\, X\overline{x} \,.\, \varphi(X, \overline{x})](\overline{y}) \equiv \forall X \big[\big((\forall \overline{x}.\text{guarded}(\overline{x}))(\varphi(X, \overline{x}) \rightarrow X\overline{x})\big) \rightarrow X\overline{y}\big].$$

It is also clear that GSO is strictly more powerful than μGF. For instance, as GSO includes all of MSO, it is neither decidable nor invariant under guarded bisimulation.

PROPOSITION 3.17. μGF $\subsetneq$ GSO.

3.5 A normal form for guarded quantification

It will be convenient for our intended translation from the guarded world into the modal world to adopt a normalisation for guarded quantification resembling the first-order transcription of modal quantification that uses only two first-order variables, x and y. Recall that the crucial feature of that translation is how x and y are used in alternating fashion to capture the semantics of nested modal quantification, as in $[\Box\Diamond\Box P](x) = \forall y\big(Exy \rightarrow \exists x\big(Eyx \wedge \forall y(Exy \rightarrow Py)\big)\big)$. Corresponding to the state variables x or y in ML we essentially want to use variable tuples for guarded lists $\overline{x}$ and $\overline{y}$ in GF. Corresponding to the transition relations Exy between states we record the overlap between components of $\overline{y}$ and components of $\overline{x}$ in the form of identities $y_i = x_j$.

Let $X = \{x_1, x_2, \ldots\}$ and $Y = \{y_1, y_2, \ldots\}$ be two disjoint sets of variables.

Definition 3.18. Let $\mathrm{GF}_0 := \mathrm{GF}_X \cup \mathrm{GF}_Y$, GF_X and GF_Y inductively defined as follows.

(1) Every relational atomic formula α with $\mathrm{free}(\alpha) \subseteq X$ belongs to GF_X; every relational atomic formula α with $\mathrm{free}(\alpha) \subseteq Y$ belongs to GF_Y.
(2) A boolean combination of formulae in GF_X also belongs to GF_X; similarly for GF_Y.
(3) Let $m, n \in \mathbb{N}$, $\overline{y} = (y_1, \ldots, y_n)$, and let ρ be any partial 1-1 map from $\{1, \ldots, n\}$ into $\{1, \ldots, m\}$. Then, for every guard $\alpha(y_1, \ldots, y_n)$ and $\varphi(y_1, \ldots, y_n) \in \mathrm{GF}_Y$ the following formula is in GF_X:

$$\big(\exists \overline{y}.\ \textstyle\bigwedge_{\rho(i)=j} y_i = x_j \wedge \bigwedge_{i \neq j} y_i \neq y_j \wedge \alpha(\overline{y})\big)\varphi(\overline{y}).$$

Interchanging the roles of X- and Y-variables we obtain formulae in GF_Y.

We let GSO_0 be the extension of GF_0 by second-order quantification over guarded relations.

In the sequel, we let $\exists^{\neq}\overline{y}\ldots$ be an abbreviation for $\exists\overline{y}\big(\bigwedge_{i\neq j} y_i \neq y_j \wedge \ldots\big)$ and denote relativised quantifications of the type $(\exists\overline{y}.\ \bigwedge_{\rho(i)=j} y_i = x_j \wedge \bigwedge_{i\neq j} y_i \neq y_j \wedge \alpha(\overline{y}))\varphi(\overline{y})$ as used in GF_0 as $(\exists^{\neq}\overline{y}\,.\,\rho(\overline{x},\overline{y}) \wedge \alpha(\overline{y}))\varphi(\overline{y})$. Note that $(\exists^{\neq}\overline{y}\,.\,\rho(\overline{x},\overline{y}) \wedge \alpha(\overline{y}))\varphi(\overline{y})$ says that we can pass from a current tuple instantiating $\overline{x}$ to a guarded list $\overline{y} = (y_1, \ldots, y_n)$ which extends the subtuple $(x_{\rho(i)}) = (y_i)$ in such a manner that the $\overline{y}$ satisfy the guard $\alpha(\overline{y})$ and the formula $\varphi(\overline{y})$.

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.

Proposition 3.19. *Every sentence in* GF *is equivalent to a sentence in* GF_0.

Proof. For an inductive argument, we consider formulae of GF and show the following.

Let $\varphi(\overline{z})$ be any formula in GF with free variables among $\overline{z}$. Let $\eta(\overline{z})$ be any complete quantifier free equality type on $\overline{z}$, and let $\overline{z}^{\eta}$ be an n-tuple of variables from $\overline{z}$ such that every $=$-class of η is represented exactly once. Then there is a formula $\varphi_X^{\eta}(\overline{x}) \in GF_X$ such that

$$\eta(\overline{z}) \wedge \overline{x} = \overline{z}^{\eta} \models \varphi(\overline{z}) \leftrightarrow \varphi_X^{\eta}(\overline{x}),$$

and, similarly, a corresponding $\varphi_Y^\eta(\overline{y}) \in \mathrm{GF}_Y$.

The atomic and Boolean cases are trivial and it just remains to deal with guarded quantification. So let $\varphi(\overline{z})$ be of the form $\varphi(\overline{z}) = \big(\exists \overline{w}.\alpha(\overline{z},\overline{w})\big)\psi(\overline{z},\overline{w})$.

For the given equality type $\eta = \eta(\overline{z})$, consider all equality types $\eta'(\overline{z},\overline{w})$ that are compatible with the given η. Clearly, $\varphi \wedge \eta$ is equivalent to a disjunction over all such η' of the form

$$\varphi(\overline{z}) \wedge \eta(\overline{z}) = \bigvee_{\eta' \supseteq \eta} \big(\exists \overline{w}.\alpha(\overline{z},\overline{w})\big)\big(\eta'(\overline{z},\overline{w}) \wedge \psi(\overline{z},\overline{w})\big).$$

We may therefore consider just one such disjunct, for a fixed $\eta'(\overline{z},\overline{w})$. Let $(\overline{z},\overline{w})^{\eta'} = (u_1,\dots,u_m) = \overline{u}$ be a tuple of representatives according to η', and let $\overline{y} = (y_1,\dots,y_m)$.

According to the inductive hypothesis there is some formula $\psi_Y^{\eta'}(y_1,\dots,y_m) \in \mathrm{GF}_Y$ such that $\eta'(\overline{z},\overline{w}) \wedge \overline{y} = \overline{u} \models \psi(\overline{z},\overline{w}) \leftrightarrow \psi_Y^{\eta'}(\overline{y})$.

Let $\alpha'(\overline{y})$ be obtained from $\alpha(\overline{z},\overline{w})$ by replacing by y_i each variable in $\alpha(\overline{z},\overline{w})$ that is identified with u_i by η'. If $\overline{z} = (z_1,\dots,z_k)$, let ρ be the partial 1-1 map defined by $\rho(i) = j$ if η' identifies u_i with z_j. Then

$$\big(\exists \overline{w}.\alpha(\overline{z},\overline{w})\big)\big(\eta'(\overline{z},\overline{w}) \wedge \psi(\overline{z},\overline{w})\big) \equiv \big(\exists \overline{y}. \textstyle\bigwedge_{\rho(i)=j} y_i = z_j \wedge \alpha'(\overline{y})\big)\psi_Y^{\eta'}(\overline{y}).$$

It follows that, after substituting X-variables for $\overline{z}$, the formula $(\exists^{\neq}\overline{y}.\rho(\overline{z},\overline{y}) \wedge \alpha')\psi_Y^{\eta'}(\overline{y})$ is in GF_X and satisfies the inductive claim. $\square$

Recall that GSO_0 is the extension of GF_0 by second-order quantification over guarded relations. The proof of the proposition clearly extends to cover GSO_0.

COROLLARY 3.20. *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. We put this correspondence on a precise footing in a framework of model theoretic translations. We associate relational structures with transition systems and vice versa, in such a way that guarded bisimulation classes of relational structures correspond to bisimulation classes of transition systems or, more specifically, of trees. This will pave the way to go back and forth between the guarded world (relational structures, guarded bisimulation equivalence, guarded logics) and the modal world (transition systems and trees, bisimulation, modal logics).

The relevant back and forth translations from structures to trees and back from trees to structures will first be presented here and in Section 5 for the parameter free case which will support the proof of our main theorem for sentences. This case is conceptually and notationally easier.

As we shall indicate in Section 6, the treatment given here extends easily to cover also the case of variable-guarded formulae. In Section 6 we shall also treat the extension to the most general case of formulae with free variables and structures with arbitrary (i.e. not necessarily guarded) tuples of parameters.

4.1 The guarded Ehrenfeucht-Fraïssé game

Guarded bisimulation equivalence may be described in terms of a natural Ehrenfeucht-Fraïssé game associated with GF. We indicate the characteristic features of the *guarded game*, as outlined in [Andréka et al. 1998], and introduce a restricted or standardised version of the game that is closely related to the GF_0 normal form of Section 3.5.

The standard game for GF and guarded bisimulation equivalence is analogous to the usual pebble games for bounded-variable logics. The purpose of the finite ℓ-round game is to determine whether two structures $\mathfrak{A}$ and $\mathfrak{B}$ are indistinguishable in GF up to guarded quantifications of depth ℓ. The infinite game, which we are mainly interested in, allows us to determine whether $\mathfrak{A}$ and $\mathfrak{B}$ are guarded bisimulation equivalent, or indistinguishable in the infinitary variant of GF.

There are two players, I and II. Player I tries to show that $\mathfrak{A}$ and $\mathfrak{B}$ are inequivalent, while Player II attempts to show that they are equivalent. The task for Player II is to mimic moves that Player I makes in one of the structures with corresponding moves in the opposite structure.

During the game pebbles can be placed on elements of the respective structures. The crucial restriction for the guarded game is that the currently marked elements in either structure must form a guarded set. After each round of the game both structures will carry pebbles with corresponding labels, so that the pebble placement induces a correspondence $\overline{a} \mapsto \overline{b}$, where $a \in A$ is associated to $b \in B$ if a and b are the positions of corresponding pebbles. Initially, i.e., before the first round, neither $\mathfrak{A}$ nor $\mathfrak{B}$ carry pebbles.

In each new round, Player I chooses one of the structures, possibly removes some of the pebbles, and then places some pebbles (from among those just removed or those not on the structure at the start of this round) on elements of this structure. The only constraint for Player I is that after this move all pebbles on this structure again form a guarded set. Player II answers by removing, from the other structure, precisely those pebbles corresponding to those first taken off in I's move, and then places pebbles corresponding to those placed by I. Player II is required to make her choices such that the new correspondence $\overline{a} \mapsto \overline{b}$ between pebbled elements is a partial isomorphism. Note that in particular this also forces her to place her pebbles so that they form a guarded set. If no such move is available, then Player II loses and the game is over.

The whole game can be played with a fixed finite number of rounds, or as an infinite game. In the ℓ-round game, Player II wins a play if she was able to respond through all ℓ rounds; Player I only wins if Player II got stuck before completion of the last round. In the infinite game, an infinite play in which Player II can always respond to the next move of Player I, is won by Player II; Player I only wins if II gets stuck after some finite number of rounds.

The classical results on Ehrenfeucht-Fraïssé games (and their proofs) obviously translate to this guarded setting: Player II has a winning strategy in the ℓ-round game if $\mathfrak{A}$ and $\mathfrak{B}$ cannot be distinguished by any sentence of GF with nesting depth ℓ for guarded quantification; and Player II has a winning strategy in the infinite game if $\mathfrak{A}$ and $\mathfrak{B}$ cannot be distinguished by any sentence of GF_∞, infinitary guarded logic, or the guarded fragment of $\mathrm{L}_{\infty\omega}$. The latter, of course, is also equivalent to $\mathfrak{A} \sim_g \mathfrak{B}$. We shall here only use this characterisation of guarded bisimulation equivalence by means of the infinite game.

Proposition 4.1. *Player II has a winning strategy in the infinite guarded*

Ehrenfeucht-Fraïssé game on $\mathfrak{A}$ *and* $\mathfrak{B}$ *iff* $\mathfrak{A} \sim_g \mathfrak{B}$.

This equivalence is straightforward, but becomes even more immediate if we look at a slight variation of the game. This variation – although equivalent to the standard version described above – has a closer analogy with the quantification pattern of GF_0, whereas the standard game is modelled on the standard quantification pattern in GF.

Let m be the width of the vocabulary τ, i.e., the maximal arity of relations in τ, which is also a bound on the size of guarded sets in τ-structures. It is not hard to see that the number of pebbles in each structure can w.l.o.g. be restricted to m. In fact, the constraint in terms of guarded sets means that whenever more than m pebbles are in the game some elements must be multiply pebbled. Whenever several pebbles are placed on the same elements, the multiplicities and pebbles concerned must be the same in both structures since the pebbles induce a partial isomorphism. It is then clear that we could w.l.o.g. force Player I to avoid multiple pebbling. One verifies inductively that if Player I can force a win in the old game then this is still true under this additional constraint. But together with the constraint that the set of pebbled elements always be a guarded set, this precisely means that we require the pebbled tuples to be guarded lists in the sense of Definition 3.1.

The analogy with quantification in GF_0 is now obvious:

– Quantification is over tuples that are guarded lists (with explicit mention of the ground atom that guards the component set); this corresponds to a new pebble placement.
– Constraints in terms of identities between newly quantified elements (quantified variables) and the previous elements (free variables) are described by a partial 1-1 map; this corresponds to the choice of pebbles that remain fixed in place during Player I's move.

With regard to the second correspondence it should be pointed out that both in the game and in GF_0 quantification nothing prevents Player II to put other pebbles back into positions that were just previously pebbled; in that sense Player I and GF can explicitly force some identifications but not forbid others.

We thus obtain the following restricted version of the guarded game, which, as we have argued, is equivalent to the more liberal standard version.

The restricted guarded game.. Let $\mathfrak{A}$ and $\mathfrak{B}$ be τ-structures, τ finite and relational of width m. The restricted guarded game on $\mathfrak{A}$ and $\mathfrak{B}$ is played by two players, I and II, who take turns to position and relocate pebbles from two sets of m pebbles labelled $1, \ldots, m$, one set for each structure. 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 may remove some of the pebbles currently placed on that structure; Player II removes the corresponding pebbles in the opposite structure. Then Player I may place any number of pebbles back onto elements of the chosen structure in such a way that the overall tuple of pebbled positions again forms a guarded list; Player II must place corresponding pebbles onto elements of the opposite structure in such a way that the induced correspondence between the pebbled elements in $\mathfrak{A}$ and $\mathfrak{B}$ is a partial isomorphism. Player II loses if no such response is available. The game is infinite unless Player II loses in this way. Player II wins a play if she does not get stuck.

It is apparent that a guarded bisimulation in the sense of Definition 3.6 *is* a formalisation of a (non-deterministic) strategy for Player II.

LEMMA 4.2. *Player II has a winning strategy in the restricted guarded game on $\mathfrak{A}$ and $\mathfrak{B}$ iff $\mathfrak{A} \sim_g \mathfrak{B}$.*

We now use the intuition behind the game to introduce the following two 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 [Andréka et al. 1998] and [Grädel 2001a].

4.2 From structures to trees

Recall from Definition 3.1 that a guarded list in $\mathfrak{B}$ is a tuple $(b_1, \dots, b_k)$ of *distinct* elements such that $\{b_1, \dots, b_k\}$ is a guarded set. We regard such guarded lists as descriptions of positions over $\mathfrak{B}$ in the restricted guarded game. The behaviour of $\mathfrak{B}$ w.r.t. the game can firstly be described as a game graph in the form of a transition system $\mathcal{K}(\mathfrak{B})$. The states of the game graph are the game positions formalised as guarded lists over $\mathfrak{B}$. Guarded lists are really used as enumerations of guarded sets and every guarded set of size k can thus be represented by $k!$ many different guarded lists. As there is in general no canonical way to single out any particular choice, we symmetrise the description of the game positions so as to include all these variant descriptions of the same guarded set, which are related by permutations that correspond to re-labellings of the pebbles.

One could introduce a kind of ε-move in the restricted guarded game in which the current pebbles are permuted, according to some choice of a permutation made by Player I, which then has to be copied identically by Player II. Of course such moves would be a vacuous addition to the game, as far as winning or losing the game is concerned. For the resulting game graph, however, they allow for a locally more symmetric description.

The information to be recorded in each node $v = (b_1, \dots, b_k)$ of the game graph is precisely the isomorphism type of the guarded substructure induced on $\{b_1, \dots, b_k\}$ (cf. Definition 3.1 (ii)). The transition relations of the game graph describe possible moves and record the constraints imposed on a move from $\overline{c} = (c_1, \dots, c_l)$ to $\overline{d} = (d_1, \dots, d_k)$ by some choice of elements that have to remain fixed (according to Player I's choice).

The transition system $\mathcal{K}(\mathfrak{B})$ *and tree* $\mathcal{T}(\mathfrak{B})$.. With the finite relational vocabulary τ of width m, associate the following vocabulary $\tilde{\tau}$ for the associated transition systems and trees. Let S be the set of all guarded τ-structures $\mathfrak{A}$ with universes $\{1,\dots,k\}$, $0 \leq k \leq m$, plus the empty structure $\mathfrak{A} = \emptyset$ for $k = 0$. Let F be the set of all partial 1-1 maps ρ from $\{1,\dots,m\}$ to $\{1,\dots,m\}$. The associated vocabulary $\tilde{\tau}$ has predicates

$$\begin{array}{lll} P_{\mathfrak{A}} & \text{(monadic)} & \text{for } \mathfrak{A} \in S \\ E_\rho & \text{(binary)} & \text{for } \rho \in F. \end{array}$$

If $\mathfrak{B}$ is a τ-structure, we write $G = G(\mathfrak{B})$ for the set of guarded lists over $\mathfrak{B}$,

$$G(\mathfrak{B}) = \{\overline{b} \colon \overline{b} \text{ a guarded list in } \mathfrak{B}\},$$

where we recall from Definition 3.1 that this includes the empty list $\emptyset$.

Definition 4.3. The transition system $\mathcal{K}(\mathfrak{B})$ is the following $\tilde{\tau}$-structure

$$\begin{array}{rcl} \mathcal{K}(\mathfrak{B}) & = & \big(G(\mathfrak{B}), (E_\rho)_{\rho\in F}, (P_{\mathfrak{A}})_{\mathfrak{A}\in S}\big) \\ \text{where} \quad E_\rho & = & \big\{\big((c_1,\dots,c_l),(d_1,\dots,d_k)\big) \colon d_j = c_{\rho(j)} \text{ for all } j \in \mathrm{dom}(\rho)\big\} \\ P_{\mathfrak{A}} & = & \big\{(b_1,\dots,b_k) \colon \mathfrak{B}\restriction\{b_1,\dots,b_k\} \simeq \mathfrak{A} \text{ via } b_i \mapsto i\big\}. \end{array}$$

Note that we have multiple transitions between nodes; in particular $E_{\rho'} \subseteq E_\rho$ for all $\rho \subseteq \rho'$.

The transition relations of $\mathcal{K}(\mathfrak{B})$ reflect possible moves over $\mathfrak{B}$ in the restricted guarded game. The choice of the E_ρ and the $P_{\mathfrak{A}}$ is such that bisimulations between two transition systems $\mathcal{K}(\mathfrak{B})$ and $\mathcal{K}(\mathfrak{B}')$ capture winning strategies for Player II in the restricted game on $\mathfrak{B}$ and $\mathfrak{B}'$. To see this, consider a game position with pebbles placed on $\overline{b}$ in $\mathfrak{B}$ and on $\overline{b}'$ in $\mathfrak{B}'$. If $\mathcal{K}(\mathfrak{B}),\overline{b} \sim \mathcal{K}(\mathfrak{B}'),\overline{b}'$, then $\overline{b} \mapsto \overline{b}'$ must be a partial isomorphism since $\overline{b}$ and $\overline{b}'$ are labelled by the same $P_{\mathfrak{A}}$.

Now consider a move in the game in which Player I plays in $\mathfrak{B}$ say, from some position $\overline{c} = (c_1,\dots,c_l)$ to a new position $\overline{d} = (d_1,\dots,d_k)$ over $\mathfrak{B}$. Up to possible permutations in the enumeration of these guarded lists (or a consistent relabelling of pebbles over both structures) this relocation can have been effected by leaving put any subset of pebbles that mark elements in $\{c_1,\dots,c_l\} \cap \{d_1,\dots,d_k\}$.

Suppose Player I left put pebbles on positions $\overline{c}_0 \subseteq \overline{c}$. Then there is some partial 1-1 map ρ from $\{1,\dots,k\}$ to $\{1,\dots,l\}$ such that $\overline{c}_0 = (d_{\rho(j)})_{j\in\mathrm{dom}(\rho)}$. It follows that $(\overline{c},\overline{d}) \in E_\rho$ in $\mathcal{K}(\mathfrak{B})$. If now $\mathcal{K}(\mathfrak{B}),\overline{c} \sim \mathcal{K}(\mathfrak{B}'),\overline{c}'$, then an application of the forth property yields a position $\overline{d}'$ over $\mathfrak{B}'$ for which $(\overline{c}',\overline{d}') \in E_\rho$ in $\mathcal{K}(\mathfrak{B}')$ and $\mathcal{K}(\mathfrak{B}),\overline{d} \sim \mathcal{K}(\mathfrak{B}'),\overline{d}'$. It follows that Player II can make a move to $\overline{d}'$ in $\mathfrak{B}'$, which satisfies both the constraints regarding pebbles that were fixed by I, and the requirement that the resulting correspondence $\overline{d} \mapsto \overline{d}'$ is again a partial isomorphism.

Inductively we thus obtain the following.

Lemma 4.4. *For all* τ*-structures* $\mathfrak{B}$ *and* $\mathfrak{B}'$*:* $\quad \mathfrak{B} \sim_g \mathfrak{B}' \quad$ *iff* $\quad \mathcal{K}(\mathfrak{B}) \sim \mathcal{K}(\mathfrak{B}')$.

The desired tree representation $\mathcal{T}(\mathfrak{B})$ of (the game on) $\mathfrak{B}$ is now obtained as an unravelling of the transition system $\mathcal{K}(\mathfrak{B})$ from the empty list. Note that the empty list $\emptyset \in G(\mathfrak{B})$ corresponds to the start position of the game, with no pebbles placed. Compare Definition 2.3 for unravellings.

The set V of nodes of $\mathcal{T}(\mathfrak{B})$ is the set of all sequences $g_0\rho_1 g_1\rho_2 g_2\dots g_n$ where all $g_i \in G(\mathfrak{B})$, with $g_0 = \emptyset$ (the empty guarded list) and for all $g_i\rho_{i+1}g_{i+1}$ we have

(i) $\rho_{i+1} \subseteq \{1,\dots,|g_{i+1}|\} \times \{1,\dots,|g_i|\}$,

(ii) 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.

According to Definition 2.3, the natural projection $\pi : \mathcal{T}(\mathfrak{B}) \to \mathcal{K}(\mathfrak{B})$ maps a node $v = g_0 \dots g_n$ to the guarded list $g_n = (b_1,\dots,b_k)$ in $\mathfrak{B}$ in which the sequence v terminates.

We associate some relevant information in $\pi(v)$ with v itself. In particular we let $|v| := |\pi(v)| = k$ (the size of v) and, if $\pi(v) = (b_1,\dots,b_k)$, we denote as $\mathfrak{A}_v$ the unique $\mathfrak{A} \in S$ that is isomorphic with $\mathfrak{B}{\restriction}\{b_1,\dots,b_k\}$ via $i \mapsto b_i$.

Definition 4.5. For a τ-structure $\mathfrak{B}$ we let $\mathcal{T}(\mathfrak{B})$ be the unravelling of the transition system $\mathcal{K}(\mathfrak{B})$ from the empty list.

$$\begin{aligned} \mathcal{T}(\mathfrak{B}) &= \big(V, (P_{\mathfrak{A}})_{\mathfrak{A}\in S}, (E_\rho)_{\rho\in F}\big), \\ \text{where}\quad P_{\mathfrak{A}} &= \big\{v \in V : \mathfrak{A}_v = \mathfrak{A}\big\}, \\ E_\rho &= \big\{(u,v) \in V^2 : v = u\rho g\big\}. \end{aligned}$$

Example 4.6. Let $\mathfrak{B}$ be as in Figure 1. We intuitively describe $\mathcal{K}(\mathfrak{B})$ and $\mathcal{T}(\mathfrak{B})$. For illustrative purposes we use the elements of $\mathfrak{B}$ in the universes of the $\mathfrak{A}_v$, instead of initial segments of the natural numbers as required in the definition.

$\mathcal{K}(\mathfrak{B})$ is an (edge and node labelled) graph that contains one node for each guarded list in $\mathfrak{B}$. This includes (a,b,d), (b,d,a) and (a,c'), but not (a,c). Any two (not necessarily distinct) nodes are connected by an $E_\emptyset$-edge, i.e. an E_ρ-edge for the empty ρ. If the intersection of the universes of the two nodes is non-empty, there are further connecting edges for corresponding non-empty ρ. E.g. from (a,c) to (c,a) there is an E_ρ-edge for all ρ that are a subset of $\{2 \mapsto 1, 1 \mapsto 2\}$, and from (a,b,d) to (d,c,b) when ρ is a subset of $\{1 \mapsto 3, 3 \mapsto 2\}$ (the first element at the destination node (d,c,b) is the same as the third element at the source node (a,b,d) and the third element at the destination node is the same as the second one at the source node).

$\mathcal{T}(\mathfrak{B})$ is the unravelling of $\mathcal{K}(\mathfrak{B})$. The structure $\mathfrak{A}_\lambda$ represented at the root is the empty structure. Inductively, set of successors of each node contains (possibly several) copies of each non-empty guarded list of $\mathfrak{B}$, i.e. (possibly several) copies of each node of $\mathcal{K}(\mathfrak{B})$. Several successors obtained from the same guarded list are used to disentangle multiple edges as occur in $\mathcal{K}(\mathfrak{B})$. E.g. a node v in $\mathcal{T}(\mathfrak{B})$ with $A_v = (a,b,d)$ has one successor node with universe (d,c,b) for each ρ that is a subset of $\{1 \mapsto 3, 3 \mapsto 2\}$ (the same set as above) and each of these successors is connected via exactly one of the E_ρ.

A major difference between how $\mathcal{K}(\mathfrak{B})$ and $\mathcal{T}(\mathfrak{B})$ represent $\mathfrak{B}$ is that in $\mathcal{K}(\mathfrak{B})$ each element of $\mathfrak{B}$ only occurs once. That is, for any two guarded lists that share an element, say b, there is an E_ρ-edge between them such that ρ explicitly connects the two b's. $\mathcal{T}(\mathfrak{B})$ behaves very differently. Consider a node v with where the universe of the structure represented at v is (a,c') and its successors v' with (a',c) and v'' with (a,c'). Since the universes of v and v', resp. v' and v'' do not share common elements (of $\mathfrak{B}$), there are only $E_\emptyset$-edges from v to v' and v' to v''. After the normalization step, that makes every $\mathfrak{A}_v$ have universe $\{1,\dots,|\mathfrak{A}_v|\}$, the information that the a at v and the a at v'' were actually the same element, is lost. This means that each element of $\mathfrak{B}$ will have multiple copies in the τ-structure that is in turn obtained from $\mathcal{T}(\mathfrak{B})$, as explained in the following section.

The following is then a direct consequence of Lemma 4.4, and the fact that unravellings are bisimilar companions.

COROLLARY 4.7. *For all τ-structures $\mathfrak{B}$ and $\mathfrak{B}'$:* $\quad \mathfrak{B} \sim_g \mathfrak{B}' \quad$ *iff* $\quad \mathcal{T}(\mathfrak{B}) \sim \mathcal{T}(\mathfrak{B}')$.

4.3 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 $\mathcal{T}$. We collect some suitable conditions, which are obviously satisfied by every tree $\mathcal{T}(\mathfrak{B})$, in the following definition.

Definition 4.8. 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 $\mathfrak{A}_v$; $\mathfrak{A}_\lambda = \emptyset$, and $\mathfrak{A}_u \neq \emptyset$ for all $u \neq \lambda$.
(b) If $(u, v) \in E_\rho$ then ρ is a partial isomorphism between $\mathfrak{A}_v$ and $\mathfrak{A}_u$.
(c) If $(u, v) \in E_\rho$, then for every isomorphic embedding $\pi : \mathfrak{A} \to \mathfrak{A}_v$ from some structure $\mathfrak{A} \in S$ to $\mathfrak{A}_v$ there exists a node w such that $\mathfrak{A}_w = \mathfrak{A}$ and $(u, w) \in E_{\rho\circ\pi}$.

The local richness conditions expressed in (c) is technically useful as it ensures that all enumerations of every guarded (sub)set are locally available as guarded lists. Note that the $\mathcal{T}(\mathfrak{B})$ satisfy much stronger homogeneity properties. For any two nodes v, w in $\mathcal{T}(\mathcal{B})$ there is a canonical isomorphism between the two forests consisting of the trees rooted at the successor nodes of v and w, respectively. This property cannot be captured in MSO. MSO-definability and closure under bisimulation of the underlying class will be technically essential in the sequel. The conditions (a)—(c) are of course first-order expressible.

LEMMA 4.9. *Within the class of $\tilde{\tau}$-trees, the class of consistent trees is bisimulation invariant and first-order definable.*

This is obvious from the definition. The important fact, however, is that these simple first-order expressible consistency conditions are sufficiently restrictive to allow us to (re)construct from every consistent $\tilde{\tau}$-tree an associated τ-structure. And in the special case that the $\tilde{\tau}$-tree comes as $\mathcal{T}(\mathfrak{B})$ for some τ-structure $\mathfrak{B}$, whose guarded behaviour is described in $\mathcal{T}(\mathfrak{B})$, we actually recover this behaviour: the resulting τ-structure will be guarded bisimulation equivalent to the original one.

The idea is to start from the $\mathfrak{A}_v$ that are represented in the nodes v of $\mathcal{T}$, and to patch these local isomorphism types together in such a way that the moves in the guarded bisimulation game recorded in $\mathcal{T}$ are realised in the resulting structure. To this end we need to identify elements in neighbouring $\mathfrak{A}_u$ and $\mathfrak{A}_v$, attached to nodes u and v, that are linked by E_ρ, in accordance with the partial isomorphism ρ. In other words, we let $\mathfrak{A}_u$ and $\mathfrak{A}_v$ overlap according to identifications prescribed in E_ρ. Technically we take the disjoint union of the $\mathfrak{A}_v$ and form the quotient w.r.t. equalities imposed by overlaps according to the E_ρ. We describe this process more formally in the following.

Let $\mathcal{T} = (V, (P_{\mathfrak{A}})_{\mathfrak{A}\in S}, (E_\rho)_{\rho\in F})$ be a consistent $\tilde{\tau}$-tree. We find the associated τ-structure $\mathfrak{D}(\mathcal{T})$ as follows. Let $U := \{(v, i) : v \in V, i \in \mathfrak{A}_v\}$ be the disjoint union of the $\mathfrak{A}_v$.

Let $\approx$ be the reflexive, symmetric transitive closure of the following relation $\approx_0$ on U:

$$(v,i) \approx_0 (w,j) \quad \Leftrightarrow \quad (v,w) \in E_\rho \text{ and } \rho(j) = i \quad \text{for some } \rho \in F.$$

The universe of $\mathfrak{D}(\mathcal{T})$ is $D := U/_\approx$, the set of $\approx$-equivalence classes $[v,i]$ in U. We say that an equivalence class $d = [v,i]$ *lives at node* u, or that d is *represented* in that node u, if $[v,i] = [u,j]$ for some j.

Note that $\approx_0$ describes the local identification pattern imposed by single E_ρ-edges. Intuitively, it is necessary to take its transitive closure because paths along consecutive different E_ρ-edges in $\mathcal{T}$ can lead to identifications involving one and the same element, if this element happens to be passed through the successive overlaps along that path. It is important to note that we are thus dealing with non-local phenomena, insofar as nodes that are far apart in $\mathcal{T}$ may represent the same element. It is clear from the definition of $\approx_0$, however, that the set of nodes of $\mathcal{T}$ that represent some fixed equivalence class d form a connected subgraph, i.e., a subtree. If an equivalence class d lives at nodes u and v in $\mathcal{T}$ then it must also live at every node along 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 R in τ over D by putting $\overline{d} = ([v,i_1],\dots,[v,i_k]) \in R$ if $\mathfrak{A}_v \models R(i_1,\dots,i_k)$.

For further reference we sum up the construction in the following definition.

Definition 4.10. For any consistent $\tilde{\tau}$-tree $\mathcal{T} = (V,(P_\mathfrak{A})_{\mathfrak{A}\in S},(E_\rho)_{\rho\in F})$ let $\mathfrak{D}(\mathcal{T}) = (D,(R)_{R\in\tau})$ be the τ-structure on universe $D = U/\approx$, where $U = \dot{\bigcup}_{v\in V} A_v = \{(v,i) \colon v \in V, i \in \mathfrak{A}_v\}$ and $\approx$ is as defined above, with interpretations of $R \in \tau$ according to

$$R^{\mathfrak{D}(\mathcal{T})} = \{\overline{d} = ([v,i_1],\dots,[v,i_k]) \colon (i_1,\dots,i_k) \in R^{\mathfrak{A}_v}\}.$$

Note that $\overline{d} = (d_1,\dots,d_k)$ is a guarded list in $\mathfrak{D}(\mathcal{T})$ if and only if $\overline{d} = ([v,1],\dots,[v,k])$ for some node $v \in \mathcal{T}$ of size $|v| = k$. In this case we say that v *represents the guarded list* $\overline{d}$. Note that the components of this list can be represented at other nodes too, including all nodes that represent permutations of the same guarded list, and possibly also at nodes representing guarded super-lists.

Lemma 4.11. *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 $\mathcal{K}$, and that the natural projection $\pi \colon \mathcal{T} \to \mathcal{K}$ associates to any node v of $\mathcal{T}$ the node $\pi(v)$ in $\mathcal{K}$, which is the last node in the path that gives rise to v. Further recall that $\pi(v)$, being a node in $\mathcal{K}$, is by definition a guarded list in $\mathfrak{B}$, $\pi(v) \in G(\mathfrak{B})$.

Let $\overline{d} = (d_1,\dots,d_k)$ be a guarded list in $\mathfrak{D}$, v any node of $\mathcal{T}$ representing this guarded list, $\overline{d} = ([v,1],\dots,[v,k])$. Let $\pi(v) = (b_1,\dots,b_k) \in G(\mathfrak{B})$. We let $f_{\overline{d},v}$ be the partial 1-1 map from $\mathfrak{D}$ to $\mathfrak{B}$ that maps $\overline{d}$ to the guarded list $\pi(v)$ in $\mathfrak{B}$:

$$f_{\overline{d},v} \colon d_i \mapsto b_i, \quad i = 1,\dots,k.$$

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 lists that enumerate the same guarded set in $\mathfrak{D}$, they are linked in $\mathcal{T}$ by a path along which all components always live together. The corresponding path in $\mathcal{K}$ similarly links $\pi(v)$ to $\pi(v')$, 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. Consider $f_{\overline{d},v} : \overline{d} \mapsto \pi(v)$. Its domain is enumerated by the guarded list $\overline{d} = (d_1, \dots, d_l)$, represented at some node v in $\mathcal{T}$; its image is enumerated by the guarded list $\pi(v) = (b_1, \dots, b_l)$ of $\mathfrak{B}$. It remains to check the back and forth conditions.

For the forth property consider any other guarded set in $\mathfrak{D}$, enumerated as a guarded list $\overline{d}'$ in $\mathfrak{D}$, which is represented by some node v' in $\mathcal{T}$. Let $\overline{c}$ be the tuple of common components in $\overline{d}$ and $\overline{d}'$. The tuple $\overline{c}$ lives along the unique shortest path from v to v' in $\mathcal{T}$. Projecting this path down to $\mathcal{K}$ and $\mathfrak{B}$ we see that $f_{\overline{d}',v'}$ and $f_{\overline{d},v}$ agree on their common domain.

For the back property consider any other guarded set in $\mathfrak{B}$, enumerated as a guarded list $\overline{b}' = (b'_1, \dots, b'_k)$. Consider both $\overline{b}'$ and $\overline{b} = \pi(v) = (b_1, \dots, b_k)$ as nodes of $\mathcal{K}$ and note that $\mathcal{T}, v \sim \mathcal{K}, \overline{b}$. Let $\rho = \{(i,j) \colon b'_i = b_j\}$. It follows that $(\overline{b}, \overline{b}') \in E_\rho$. As $\mathcal{T}, v \sim \mathcal{K}, \overline{b}$, there is a node v' in $\mathcal{T}$ such that $(v, v') \in E_\rho$ and $\mathcal{T}, v' \sim \mathcal{K}, \overline{b}'$. This node v' represents a guarded list $\overline{d}'$ of $\mathfrak{D}$ such that $f_{\overline{d}',v'} \colon \overline{d}' \mapsto \overline{b}'$. As E_ρ induces an identification of elements between $\overline{b}'$ and $\overline{b}$ according to ρ, it is clear that the inverses of $f_{\overline{d}',v'}$ 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 [Andréka et al. 1998] and [Grädel 2001a].

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 4.12. *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 two-way bisimulation between $\mathcal{T}$ and $\mathcal{T}'$ according to Lemma 2.4. Recall in this context that, according to our convention in Definition 2.1, trees have unique edge labels.

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. This is immediate from $\mathfrak{A}_v = \mathfrak{A}_{v'}$, which in turn is just a consequence of bisimilarity between v and v' at the level of basic propositions: $v' \in P_{\mathfrak{A}}$ iff $v \in P_{\mathfrak{A}}$ for all $\mathfrak{A} \in S$. Note also that for $(u, v) \in E_\rho$ and $(u', v') \in E'_\rho$ 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 $\mathfrak{D}(\mathcal{T}')$.

As the situation is entirely symmetric, it suffices to argue, for instance, for 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 a two-way bisimulation, 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 $f_{uu'}{\restriction}X$. $\square$

Remark. It is straightforward to extend all the notions and observations of this section to the case of structures $\mathfrak{B}$ with a distinguished parameter tuple $\overline{b}$ that is a guarded list in $\mathfrak{B}$. The tree $\mathcal{T}(\mathfrak{B}, \overline{b})$, for instance, would be rooted in the guarded list $\overline{b}$ rather than in the empty list, and could alternatively be described as the unravelling of $\mathcal{K}(\mathfrak{B})$ from $\overline{b}$. In the opposite direction, i.e. for the passage from a consistent $\tilde{\tau}$-tree to a τ-structure, one can clearly extract a guarded list from the root as the distinguished parameter tuple for $\mathfrak{D}(\mathcal{T})$. All results, and in particular Lemma 4.11 and Proposition 4.12, go through trivially with a distinguished guarded list of parameters.

5. BACK AND FORTH

In this section, as in the last, we explicitly work towards the case of sentences, and a proof of the main theorem for sentences.

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 [Janin and Walukiewicz 1996].

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, in the case of sentences, 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 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) with Proposition 4.12 that GSO sentences that are invariant under guarded bisimulation are mapped to MSO formulae that are bisimulation invariant on consistent trees; cf. Corollary 5.8 below.

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 $\mathfrak{A}_v$ (if not empty) represents many different guarded tuples, which may or may not occur in a given guarded set, independently of each other.

In the translation that takes us from relational structures to trees, a second-order variable, which ranges over some guarded relation in the relational structure, has to be coded by several monadic second-order variables over the tree, each one containing the information about membership of one particular guarded tuple of components that live at the corresponding node. We translate an r-ary second-order variable Z in a GSO sentence into a sequence $Z^{\rightarrow}$ of monadic variables $Z_{i_1,\ldots,i_r}$, one for each (local) choice of elements from the guarded list. The idea is that $v \in Z_{i_1,\ldots,i_r}$ for a node v representing a guarded list $([v,1],\ldots,[v,k])$ stands for $([v,i_1],\ldots,[v,i_r]) \in Z$.

In the other direction we have to deal with monadic second-order variables, which range over arbitrary sets of nodes of the tree, and in particular over sets of nodes of different sizes, corresponding to guarded lists of different lengths. We choose a translation that effectively splits up any monadic second-order variable X into disjoint parts, each corresponding to nodes of a fixed size, before translating these parts into guarded relations of appropriate arity. Consequently, a monadic second-order variable X is translated into a sequence $X^{\leftarrow}$ of second-order variables $X_i, 0 \leq i \leq m$, such that each X_i ranges over guarded lists of length i. The idea is that $\overline{d} \in X_i$ stands for $v \in X$ where the guarded list $\overline{d}$ has length i and is represented at node v.

5.1 From guarded to monadic second-order logic.

Without loss of generality we restrict attention to GSO sentences in GSO_0, cf. Corollary 3.20. Let m be the width of τ, i.e. the maximal arity of relations in τ. If Z is an r-ary second-order variable, then $Z^{\rightarrow} := (Z_{i_1,\ldots,i_r})_{i_1,\ldots,i_r \leq m}$ is the corresponding sequence of monadic predicates according to the above discussion.

Definition 5.2. Let $\mathcal{T}$ be a consistent $\tilde{\tau}$-tree, and let $\mathfrak{D}(\mathcal{T})$ be the associated τ-structure. A tuple $J^{\rightarrow}$ of monadic predicates $J_{i_1,\ldots,i_r}$ on $\mathcal{T}$ *encodes* an r-ary guarded relation J on $\mathfrak{D}(\mathcal{T})$ iff $J_{i_1,\ldots,i_r} = \{v : ([v,i_1],\ldots,[v,i_r]) \in J\}$ for all $i_1,\ldots,i_r \leq m$.

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) $J_{i_1,\ldots,i_\ell}$ 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 $(d_1,\ldots,d_r)$ is represented by $(i_1,\ldots,i_r)$ at node u and by $(j_1,\ldots,j_r)$ 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 $r \leq m$ there exists a first-order formula* $\mathrm{correct}(Z^{\rightarrow})$ *that expresses the correctness conditions (a) and (b) above.*

These conditions are necessary and sufficient in the sense that a tuple $\overline{J}$ over a consistent tree $\mathcal{T}$ 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 $\mathrm{correct}(Z^{\rightarrow})$ that states condition (a) in an obvious way and contains, for each $\rho \in F$ and each tuple $(i_1, \ldots, i_r)$ over $\mathrm{dom}(\rho)$ a clause

$$\forall x \forall y \big[E_\rho xy \rightarrow \big(Z_{i_1,\ldots,i_r}(y) \leftrightarrow Z_{\rho(i_1),\ldots,\rho(i_r)}(x) \big) \big].$$

The proof of the adequacy claim is straightforward. □

First-order quantifiers also require special treatment. Let us first consider the case where $\mathcal{T}$ is a tree representation $\mathcal{T}(\mathfrak{B})$. As noted earlier such trees satisfy the strong homogeneity condition that 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 $\mathfrak{B}$ is available locally, at some child of the current node in $\mathcal{T}(\mathfrak{B})$. Therefore guarded first-order quantifications over $\mathfrak{B}$ can be simulated over $\mathcal{T}(\mathfrak{B})$ by moving to an immediate successor of the current node (i.e. by a modal quantifier $\Diamond$ or $\Box$). However, in the relationship between $\mathcal{T}$ and $\mathfrak{D}(\mathcal{T})$, 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.

Example 5.4. Let $\tau = \{R\}$, R ternary, and consider the GF_0 formula $\varphi(y_1) = (\exists^{\neq} x_2 x_3 . R(y_1 x_2 x_3))$. We give a partial description of a series of consistent $\tilde{\tau}$-trees $\mathcal{T}_n$ as follows (explicitly exhibiting only the parts relevant to this example). $\mathcal{T}_n$ has nodes $\lambda, v_0, v_1, \ldots, v_n$ (plus some extra nodes required by consistency condition (c) in Definition 4.8); $\mathfrak{A}_\lambda = \emptyset$, $\mathfrak{A}_{v_i} = (\{1\}, R^{\mathfrak{A}_{v_i}} = \{(1,1,1)\})$ for $i < n$ and $\mathfrak{A}_{v_n} = (\{1,2,3\}, R^{\mathfrak{A}_{v_n}} = \{(1,1,1),(1,2,3)\})$. There is an $E_\emptyset$-edge from λ to v_0, and for each $i = 1, \ldots, n$ there is an $E_\emptyset$ and an E_ρ-edge from v_{i-1} to v_i, where ρ is the function that maps 1 to 1.

Observe that $\mathfrak{D}(\mathcal{T}_n) = \mathfrak{A}_{v_n}$ and consequently $\mathfrak{D}(\mathcal{T}_n) \models \varphi([v_0, 1])$ for all n. Whatever the exact definition of $\varphi^{\rightarrow}(z)$, we expect that $\mathcal{T}_n \models \varphi^{\rightarrow}(v_0)$ due to the fact that if $y_1 = [v_0, 1] = [v_n, 1]$ then $\exists^{\neq} x_2 x_3 . R(y_1 x_2 x_3)$ holds in $\mathfrak{A}_{v_n}$. However, in each $\mathcal{T}_n$ the distance from v_0 to a node that can verify the existence of the additional elements as required by φ is n. This shows that it will not be possible to simply translate our given guarded first-order quantification into their modal counterparts $\Diamond$ and $\Box$, since those are local.

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

LEMMA 5.5. *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* $\overline{W} = (W^k_\rho)_{k \le m, \rho \in F}$ *of monadic predicates on* $\mathcal{T}$*, we have* $\mathcal{T} \models$ F-part$(u, \overline{W})$ *if and only if for all* k, ρ

$$W^k_\rho = \{w : |\mathfrak{A}_w| = k \text{ and } \rho(i) = j \Leftrightarrow [w,i] = [u,j]\}.$$

PROOF. As in the previous lemma, it suffices to impose corresponding conditions locally and along all edges. 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 ρ, k, ℓ and σ: if $(x, y) \in E_\rho$, $|\mathfrak{A}_x| = \ell$, $|\mathfrak{A}_y| = k$, then $x \in W^\ell_\sigma \Leftrightarrow y \in W^k_{\sigma \circ \rho}$ and $y \in W^k_\sigma \Leftrightarrow x \in W^\ell_{\sigma \circ \rho^{-1}}$.

One shows by induction on the distance from u that F-part$(u, \overline{W})$ 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 an $\mathrm{MSO}[\tilde{\tau}]$ 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) The translation commutes with $\neg$, $\wedge$, $\vee$.

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

(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 5.6. *Let* $\mathcal{T}$ *be a consistent* $\tilde{\tau}$*-tree with root* λ*,* $\mathfrak{D}(\mathcal{T})$ *the associated* τ*-structure and* ψ *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 5.7. $\mathfrak{D}(\mathcal{T}) \models \psi(d_1,\dots,d_k,J_1,\dots,J_r) \iff \mathcal{T} \models \psi^{\rightarrow}(v,J_1^{\rightarrow},\dots,J_r^{\rightarrow})$.

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 $\psi := (\exists^{\neq}\overline{y}\,.\,\rho(\overline{x},\overline{y}) \wedge \alpha(\overline{y}))\varphi(\overline{y})$. As second-order variables play no role for this case, we suppress them. $\psi^{\rightarrow}(x) = (\exists\overline{W}\,.\,\text{F-part}(x,\overline{W}))\bigvee_{\sigma\supseteq\rho}(\exists y\,.\,W_\sigma^{|\overline{y}|}(y))(\alpha\wedge\varphi)^{\rightarrow}(y)$.

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 $e_i = d_j$ for $\rho(i) = j$. Thus $\sigma = \{(i,j)\colon e_i = d_j\}$ extends ρ. Since $\overline{e}$ is guarded, there exists a node w of size $|w| = \ell$ such that all e_i live together at w. Actually, due to condition (c) for consistent trees, we can assume that $\overline{e} = ([w,1],\dots,[w,\ell])$. We know that there exists a $\overline{W}$ satisfying F-part$(v,\overline{W})$. Since $\sigma = \{(i,j)\colon [w,i] = [v,j]\}$, it follows that $\mathcal{T} \models W_\sigma^\ell(w)$ and, by induction hypothesis, $\mathcal{T} \models (\alpha\wedge\varphi)^{\rightarrow}(w)$. Therefore $\mathcal{T} \models \psi^{\rightarrow}(v)$.

Conversely, suppose that $\mathcal{T} \models \psi^{\rightarrow}(v)$. For the (unique) tuple $\overline{W}$ satisfying F-part$(v,\overline{W})$, there exist $\sigma \supseteq \rho$ and a node $w \in W_\sigma^\ell$ 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 5.8. *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 4.12. 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 also 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 from the discussion leading up to Definition 4.5 how each node v of $\mathcal{T}$ represents a guarded list $\pi(v)$ 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 $\{\pi(v) \in \mathfrak{B} : \mathcal{T},v \models \varphi\}$. The main problem is that the guarded lists represented in $\varphi^{\mathcal{T}}$ are not in general of the same lengths, whence the full set cannot be described by a single guarded formula $\psi^{\mathfrak{B}}$ at all. We will actually translate φ into a tuple $(\varphi_0^{\leftarrow},\dots,\varphi_m^{\leftarrow})$ of formulae over $\mathfrak{B}$ where $\varphi_k = \varphi(x_1,\dots,x_k)$, with the intention that φ_k defines the set of those guarded lists $\overline{b} = \pi(v)$ of length k, for which $v \in \varphi^{\mathcal{T}}$. Note that the 0-component, $\varphi_0^{\leftarrow}$, will be a sentence whose truth value tells us whether φ is satisfied in λ, the only node of size 0. As pointed out in the introduction to this chapter, the same problem occurs in connection with the translation of monadic second-order variables, whose interpretations, like $\varphi^{\mathcal{T}}$, can consist of arbitrary collections of nodes that correspond to guarded lists of different lengths. The same representation mechanism works in both cases.

With any monadic second-order variable X we associate a tuple $X^{\leftarrow} := (X^{(0)}, \dots, X^{(m)})$, where each $X^{(k)}$ is a k-ary second-order variable. $X^{(0)}$ in particular should be regarded as a Boolean variable with values *true* or *false*. Formally one can of course replace this nullary second-order variable by using a unary one (which is automatically guarded) whose interpretations are restricted to the empty set (for *false*) or the full universe (for *true*).

Definition 5.9. Let N be a set of nodes in a tree $\mathcal{T}(\mathfrak{B})$. The *representation* of N in $\mathfrak{B}$ is $N^{\leftarrow} := (N^{(0)}, \dots, N^{(m)})$ where $N^{(k)} := \{\pi(v) : v \in N \text{ and } |\mathfrak{A}_v| = k\} \subseteq B^k$.

Note again that, since the root λ is the only node of size 0, $N^{(0)} = \mathit{true}$ iff $\lambda \in N$.

It is known (see e.g. [Niwinski and Walukiewicz 1997]) that we can 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 $\varphi \in \mathrm{L}_\mu$, the fixed point formulae $\mu X.\xi(X)$ themselves only occur inside the scope of modal operators.

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

(1) If $\varphi = P_{\mathfrak{A}}$, then set $\varphi_k^{\leftarrow} := \mathit{false}$ if $|\mathfrak{A}| \neq k$, and otherwise let $\varphi_k^{\leftarrow}(x_1, \dots, x_k)$ be the atomic type of $(1, \dots, k)$ in $\mathfrak{A}$, i.e., the conjunction over all atomic and negated atomic τ-formulae α such that $\mathrm{free}(\alpha) \subseteq \{x_1, \dots, x_k\}$ and $\mathfrak{A} \models \alpha(1, \dots, k)$. 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 $1 \leq k \leq m$, then $\varphi_k^{\leftarrow}(x_1, \dots, x_k) := X^{(k)}(x_1, \dots, x_k)$. We set $\varphi_0^{\leftarrow} := \mathit{false}$.
(3) The translation commutes with $\neg$, $\wedge$ and $\vee$ (for each k).
(4) If $\varphi = \langle\rho\rangle\vartheta$, then $\varphi_k^{\leftarrow}(x_1, \dots, x_k) := \bigvee_\alpha (\exists^{\neq} y_1, \dots, y_\ell \,.\, \rho(\overline{x}, \overline{y}) \wedge \alpha(\overline{y}))\vartheta_\ell^{\leftarrow}(\overline{y})$, where α ranges over all guards in variables $y_1, \dots, y_\ell$ for ℓ such that $\mathrm{dom}(\rho) \subseteq \{1, \dots, \ell\}$.
(5) If $\varphi = \mu X.\vartheta$, then $\varphi_0^{\leftarrow}, \varphi_1^{\leftarrow}, \dots, \varphi_m^{\leftarrow}$ are the components $[\mathbf{lfp}\ X^{(i)} \,.\, S]$ of the simultaneous least fixed point defined by the system

$$S := \begin{cases} X^{(0)} & := \ \vartheta_0^{\leftarrow}(X^{\leftarrow}) \\ X^{(1)} y_1 & := \ \vartheta_1^{\leftarrow}(y_1, X^{\leftarrow}) \\ & \ \ \vdots \\ X^{(m)} \overline{y}_m & := \ \vartheta_m^{\leftarrow}(\overline{y}_m, X^{\leftarrow}) \end{cases}$$

of fixed point equations, where $\overline{y}_k = (y_1, \dots, y_k)$.

In connection with $X^{(0)}$ in (5) note that in the light of the assumption that all μ-operators and fixed-point variables occur only in the scope of modal operators, we are not ultimately interested in whether λ enters a fixed point. In this sense the Boolean component is redundant.

THEOREM 5.10. *Let $\mathfrak{B}$ be a τ-structure, let λ be the root of $\mathcal{T}(\mathfrak{B})$ and let ψ be a formula in $\mathrm{L}_\mu[\tilde{\tau}]$. Then $\mathfrak{B} \models \psi_0^{\leftarrow}$ iff $\mathcal{T}(\mathfrak{B}), \lambda \models \psi$.*

PROOF. Again we inductively prove a more general statement, involving free second-order variables and the semantics of ψ on the whole of $\mathcal{T}(\mathfrak{B})$. We consider the case of just one free monadic second-order variable Y. This case is entirely indicative of the general case, but notationally somewhat lighter. Let $\varphi(Y)$ be a formula in $\mathrm{L}_\mu[\tilde{\tau}]$, N a set of nodes in $\mathcal{T} = \mathcal{T}(\mathfrak{B})$ with representation $N^{\leftarrow}$.

CLAIM 5.11. $(\varphi_0(N^{\leftarrow})^{\mathfrak{B}}, \dots, \varphi_m(N^{\leftarrow})^{\mathfrak{B}})$ is the representation of $\varphi(N)^{\mathcal{T}}$ in $\mathfrak{B}$ in the sense of Definition 5.9.

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

Consider (4) and let $\varphi = \langle\rho\rangle\vartheta$. Suppose first that $\mathcal{T}, v \models \varphi$, where $|v| = k$, $\pi(v) = (b_1, \dots, b_k)$. We need to show that $\mathfrak{B} \models \varphi_k^{\leftarrow}(b_1, \dots, b_k)$. As $\mathcal{T}, v \models \varphi$, there is a node w such that $(v, w) \in E_\rho$ and $\mathcal{T}, w \models \vartheta$. Let $\pi(w) = (c_1, \dots, c_\ell)$ and note that $(v, w) \in E_\rho$ ensures that $\mathrm{dom}(\rho) \subseteq \{1, \dots, \ell\}$. Then $b_i = c_j$ for all $(i, j) \in \rho$ and, by induction hypothesis, $\mathfrak{B} \models \vartheta_\ell^{\leftarrow}(c_1, \dots, c_\ell)$. Hence $\mathfrak{B} \models \varphi_k^{\leftarrow}(b_1, \dots, b_k)$.

Conversely, suppose that $\mathfrak{B} \models \varphi_k^{\leftarrow}(b_1, \dots, b_k)$. This means that there exists a guarded list $(c_1, \dots, c_\ell)$ with $b_i = c_j$ for all $(i, j) \in \rho$ and such that $\mathfrak{B} \models \vartheta_\ell^{\leftarrow}(c_1, \dots, c_\ell)$. By the construction of $\mathcal{T}(\mathfrak{B})$, there exists a node w such that $(v, w) \in E_\rho$ and $\pi(w) = (c_1, \dots, c_\ell)$. 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 X^α of the fixed point induction on φ,

$$\begin{aligned} X^0 &= \emptyset \\ X^{\alpha+1} &= \vartheta(X^\alpha) \;=\; \{x : \vartheta(X^\alpha)\} \\ \varphi^\delta &= \textstyle\bigcup_{\alpha<\delta} \varphi^\alpha \text{ for limit ordinals } \delta. \end{aligned}$$

Let similarly, for the stages of the simultaneous fixed point of the system S in $\varphi^{\leftarrow}$,

$$\begin{aligned} (X^{(k)})^0 &= \emptyset \\ (X^{(k)})^{\alpha+1} &= \vartheta_k^{\leftarrow}((X^{\leftarrow})^\alpha) \\ (X^{(k)})^\delta &= \textstyle\bigcup_{\alpha<\delta} (X^{(k)})^\alpha. \end{aligned}$$

By induction hypothesis, if $M^{\leftarrow}$ is the representation of M in $\mathfrak{B}$, then $(\vartheta_0^{\leftarrow}(M^{\leftarrow}), \dots, \vartheta_m^{\leftarrow}(M^{\leftarrow}))$ is the representation of $\vartheta(M)^{\mathcal{T}}$ in $\mathfrak{B}$. By induction on α, this implies that the stage $(X^{\leftarrow})^\alpha$ of $[\mathbf{lfp}\ X^{\leftarrow}. S]$ represents in $\mathfrak{B}$ the stage X^α of $\mu X.\vartheta$ in $\mathcal{T}$. Hence the same is true for the least fixed points. ☐

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

THEOREM 5.12. *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 $\psi^{\rightarrow}(x)$ be its translation into $\mathrm{MSO}[\tilde{\tau}]$. By Corollary 5.8 $\psi^{\rightarrow}(x)$ is bisimulation-invariant on consistent trees. Recall that the consistency condition for trees can be formulated by a monadic second-order (in fact even first-order) sentence γ, which is bisimulation invariant with respect to all trees. As a consequence, the formula $(\gamma \rightarrow \psi^{\rightarrow})(x)$ 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 the 0-component of its translation into $\mu\mathrm{GF}[\tau]$. Putting everything together, we have

$$\mathfrak{B} \models \psi \;\Leftrightarrow\; \mathfrak{D}(\mathcal{T}(\mathfrak{B})) \models \psi \;\Leftrightarrow\; \mathcal{T}(\mathfrak{B}) \models \psi^{\rightarrow}(\lambda) \;\Leftrightarrow\; \mathcal{T}(\mathfrak{B}), \lambda \models \varphi \;\Leftrightarrow\; \mathfrak{B} \models \varphi_0^{\leftarrow}.$$

The first equivalence uses guarded bisimulation invariance of ψ and Lemma 4.11; the second one is an application of Theorem 5.6; the third equivalence follows from the Janin Walukiewicz Theorem; the fourth is an application of Theorem 5.10. □

Remark. Again, there are straightforward extensions of all the results in this section at least to the case of structures with distinguished guarded lists of parameters, and to formulae whose free variables get interpreted by guarded lists. Compare in particular the claims 5.7 and 5.11.

6. PARAMETERS AND FORMULAE WITH FREE VARIABLES

6.1 Guarded lists, guarded tuples, and variable-guarded formulae

We look at extensions of our main results to the analysis of formulae with free variables, and to structures with parameters. The most natural case, in the spirit of the guarded scenario, is that of guarded parameter tuples. Most of our techniques and results, however, can meaningfully be generalised even to the case of formulae in arbitrary tuples of variables, and correspondingly to structures with arbitrary parameter tuples rather than guarded ones.

As pointed out above, a special case of parameter tuples, namely that of guarded lists, was actually implicit in the basic case as treated in Sections 4 and 5. One can put a distinguished guarded list from $\mathfrak{B}$ at the root of the tree representation $\mathcal{T}(\mathfrak{B})$, rather than using the empty guarded list as a default as we did above. For guarded lists of parameters – and for formulae whose free variables get interpreted as guarded lists – the above treatment in fact goes through almost verbatim. The claims 5.7 and 5.11 that were established in the proofs of Theorems 5.6 and 5.10, in particular, already deal with formulae whose free variables refer to guarded lists.

This approach immediately yields the further generalisation to the important case of variable-guarded formulae. We first describe the simple argument required for this extension, as a shortcut to the variable-guarded case for our main theorem. The most general case of formulae, whose free variables refer to arbitrary parameter tuples, is then discussed in Sections 6.2 and 6.3 below, thereby also indicating an alternative approach to the case of variable-guarded formulae.

Recall that a formula $\varphi(\overline{x})$ is variable-guarded if it is logically equivalent to $\varphi(\overline{x}) \wedge \mathrm{guarded}(\overline{x})$. Also recall from Section 3 that the GF formula $\mathrm{guarded}(\overline{x})$ is a disjunction over formulae $\xi_{\eta\alpha}(\overline{x})$ specifying the equality type $\eta(\overline{x})$ of $\overline{x}$ and an extension of $\overline{x}$ to an α-atom. Putting $\xi_{\eta\alpha}(\overline{x})$ into GF_0 normal form we may assume $\xi_{\eta\alpha}(\overline{x}) = \eta(\overline{x}) \wedge \big(\exists^{\neq}\overline{y}.\rho(\overline{x},\overline{y}) \wedge \alpha(\overline{y})\big)\alpha(\overline{y})$. For a variable-guarded GSO formula $\varphi(\overline{x})$, we therefore find that it is equivalent to a disjunction of formulae of the form

$$\eta(\overline{x}) \wedge \big(\exists^{\neq}\overline{y}.\rho(\overline{x},\overline{y}) \wedge \alpha(\overline{y})\big)\tilde{\varphi}(\overline{y}),$$

where the $\tilde{\varphi}(\overline{y})$ are obtained through suitable variable substitutions in $\varphi(\overline{x})$ (not all y necessarily free in $\tilde{\varphi}(\overline{y})$). The new variables $\overline{y}$ in $\tilde{\varphi}(\overline{y})$ stand for guarded lists. Assuming this normal form for variable-guarded $\varphi(\overline{x}) \in \mathrm{GSO}$, and using the formalisation of guarded bisimulation with parameters as discussed in connection with Definition 3.6, we find that $\varphi(\overline{x})$ is invariant under guarded bisimulation if and only if each $\tilde{\varphi}(\overline{y})$ is invariant under guarded bisimulation. The case of formulae whose free variables stand for guarded lists, and the translation machinery treated in Sections 4 and 5 are therefore sufficient to yield the following strengthening of Theorem 5.12.

THEOREM 6.1. *Every variable-guarded formula of* GSO *that is invariant under guarded bisimulation is equivalent to a formula in* μGF.

6.2 Back and forth with arbitrary parameters

We outline the modifications required in the back and forth translations of Sections 4 and 5 in order to incorporate arbitrary tuples of parameters right from the beginning. In particular we extend the correspondence between relational structures and suitable trees to incorporate arbitrary parameter tuples.

To see the motivation behind the following stipulations, consider the variation in the guarded bisimulation game that is required to capture equivalence with respect to arbitrary tuples of parameters. This variation merely concerns the start position in the game. For the game on $A, \overline{a}$ and $\mathfrak{B}, \overline{b}$, we initially put pebbles on $\overline{a}$ and $\overline{b}$. Player II has lost if $\overline{a} \mapsto \overline{b}$ is not a partial isomorphism. Otherwise, i.e., if the initial pebbling describes a partial isomorphism, we proceed as before. Player I can choose either structure and remove any number of pebbles from that structure, then put pebbles so as to end up with a pebbling of a guarded list; Player II must respond by first removing corresponding pebbles in the opposite structure, then putting pebbles in such a way that the resulting correspondence is again a partial isomorphism. Note that all information about the initial parameters is lost after this first round, as far as it is not carried over via pebbles that are left fixed by Player I in his first move.

Recall the vocabulary $\tilde{\tau}$ used for the trees $\mathcal{T}(\mathfrak{B})$ associated with τ-structures $\mathfrak{B}$: $\tilde{\tau}$ has monadic predicates $P_{\mathfrak{A}}$ for all $\mathfrak{A} \in S$, S the set of guarded τ-structures on universes $\{1, \ldots, k\}$, k bounded by the width m of τ, and binary relations E_ρ for $\rho \in F$, F the set of partial bijections on $\{1, \ldots, m\}$. In order to treat tuples of n distinct parameters $\overline{b} = (b_1, \ldots, b_n)$, which need not form a guarded tuple, we expand $\tilde{\tau}$ to $\tilde{\tau}_n$ by adding further predicates $P_{\mathfrak{A}/n}$ and $E_{\rho/n}$ for $\mathfrak{A} \in S_n$ and $\rho \in F_n$ where S_n is the set of all τ-structures on universe $\{1, \ldots, n\}$, F_n the set of partial 1-1 maps $\rho \colon \{1, \ldots, m\} \to \{1, \ldots, n\}$.

Let $\mathfrak{B}$ be a τ-structure, $\overline{b} = (b_1, \ldots, b_n)$ a tuple of *distinct* parameters in $\mathfrak{B}$. With $\mathfrak{B}, \overline{b}$ we associate a tree $\mathcal{T}(\mathfrak{B}, \overline{b})$ of type $\tilde{\tau}_n$, obtained as an unravelling of a transition system $\mathcal{K}(\mathfrak{B}, \overline{b})$, which captures the guarded bisimulation game on $\mathfrak{B}$ with initial position $\overline{b}$.

$\mathcal{K}(\mathfrak{B}, \overline{b})$ has nodes for all guarded lists g in $\mathfrak{B}$ and one extra distinguished node p for the parameter tuple $\overline{b}$. We use a separate node p to represent $\overline{b}$ even if $\overline{b}$ happens to be a guarded list itself.

The restriction of $\mathcal{K}(\mathfrak{B}, \overline{b})$ to the set of nodes $g \neq p$ is isomorphic with the old $\mathcal{K}(\mathfrak{B})$. In the neighbourhood of the distinguished node p we interpret the new predicates. Put $p \in P_{\mathfrak{A}/n}$ for that $\mathfrak{A} \in S_n$ that is isomorphic with $\mathfrak{B}\restriction\{b_1, \dots, b_n\}$ via $i \mapsto b_i$, and let $(p, g) \in E_{\rho/n}$ if g is a guarded list in $\mathfrak{B}$ such that ρ describes a partial isomorphism between $\mathfrak{B}\restriction g$ and $\mathfrak{B}\restriction\{b_1, \dots, b_n\}$. Note that we put no ingoing edges to p. These choices capture the special role that the parameters $\overline{b}$ play as initial conditions in the bisimulation game. In the first round the game passes to guarded lists extending a chosen sub-tuple of $\overline{b}$, and the identity of that sub-tuple needs to be preserved by the second player; any components of the parameter tuple that are not kept in this move lose their special identity.

For $\mathcal{T}(\mathfrak{B}, \overline{b})$ we take the unravelling of $\mathcal{K}(\mathfrak{B}, \overline{b})$ from the distinguished node p. The root λ of $\mathcal{T}(\mathfrak{B}, \overline{b})$ represents $\overline{b}$ and is uniquely distinguished by being coloured by a monadic predicate $P_{\mathfrak{A}/n}$ and by having outgoing edges of types $E_{\rho/n}$.

The corresponding class of consistent $\tilde{\tau}_n$-trees $\mathcal{T}$ is defined analogously to the class of consistent $\tilde{\tau}$-trees, with the additional stipulations that

(a) There is a unique $\mathfrak{A} \in S_n$ such that the root of $\mathcal{T}$ is in $P_{\mathfrak{A}/n}$; we denote it by $\mathfrak{A}_\lambda$.

(b) All edges from the root are $E_{\rho/n}$-edges for $\rho \in F_n$, and such edges only occur from the root; if $(\lambda, v) \in E_{\rho/n}$, then ρ is a partial isomorphism between $\mathfrak{A}_v$ and $\mathfrak{A}_\lambda$.

(c) If $\overline{c} = (c_1, \dots, c_k)$ is a guarded list in $\mathfrak{A}_\lambda$, then there is a node w such that $(\lambda, w) \in E_{\rho/n}$ for $\rho = \{(i, c_i) \colon 1 \leq i \leq k\}$.

It is clear how to construct a τ-structure $\mathfrak{D}(\mathcal{T})$ from a consistent $\tilde{\tau}_n$-tree $\mathcal{T}$, in complete analogy with the procedure described in Section 4. The distinguished parameters $\overline{d} = (d_1, \dots, d_n)$ for $\mathfrak{D}(\mathcal{T})$ and their quantifier free type are extracted from $\mathfrak{A}_\lambda$.

With these modifications we find that all the results of Section 4 remain valid in the extended setting. In particular,

—$\mathfrak{B}, \overline{b} \sim_g \mathfrak{B}', \overline{b}'$ iff $\mathcal{T}(\mathfrak{B}, \overline{b}) \sim \mathcal{T}(\mathfrak{B}', \overline{b}')$; cf. Lemma 4.7.

—Consistency is first-order definable and bisimulation invariant within the class of $\tilde{\tau}_n$-trees; cf. Lemma 4.9.

—For all $\mathfrak{B}, \overline{b}$: $\mathfrak{D}(\mathcal{T}(\mathfrak{B}, \overline{b})), \overline{d} \sim_g \mathfrak{B}, \overline{b}$; cf. Lemma 4.11.

—For all consistent $\tilde{\tau}_n$-trees $\mathcal{T}$ and $\mathcal{T}'$: if $\mathcal{T} \sim \mathcal{T}'$, then $\mathfrak{D}(\mathcal{T}), \overline{d} \sim_g \mathfrak{D}(\mathcal{T}'), \overline{d}'$; cf. Proposition 4.12.

We now address the issue of translating formulae between the guarded world of τ-structures with parameters and the modal world of their tree representations.

6.2.0.1 *From guarded to monadic second-order logic.* . In order to incorporate the not necessarily guarded tuple of parameters it is convenient to augment the GSO_0 normal form by a third set of new variables, $z_1, \dots, z_n$, which are only to be used as free variables for the parameters. A GSO-formula is in extended GSO_0 normal form if its free variables are among the z_i and if all its subformulae beginning with (guarded) first-order quantifications are in ordinary GSO_0 normal form. W.l.o.g. we consider only GSO formulae in this normal form which moreover require the free variables $\overline{z} = (z_1, \dots, z_n)$ to stand for pairwise distinct elements through conjuncts $z_i \neq z_j$ for $i \neq j$, so that their semantics fits the above format of τ-structures $\mathfrak{B}$ with tuples $\overline{b} = (b_1, \dots, b_n)$ of pairwise distinct parameters.

For a second-order variable Z of arity r, ranging over guarded relations over the τ-structures $\mathfrak{B}$, we again introduce a tuple $Z^{\rightarrow}$ of monadic second-order variables. These range over subsets of the $\mathcal{T}(\mathfrak{B},\overline{b})$ that do not contain the root. This latter condition is to be added to the correctness criterion, since the root of $\mathcal{T}(\mathfrak{B},\overline{b})$ does not in general represent a guarded tuple. With this modification Lemma 5.3 carries over directly.

The first-order formula F-part$(z,\overline{W})$ of Lemma 5.5 needs to be modified in order to take edges $E_{\rho/n}$ into account. It is convenient for the further translation to choose the $\overline{W}$ such that none of the W^k_ρ contain the root, and to stipulate this condition in F-part$(z,\overline{W})$. The compatibility conditions imposed by links through the root, then, have to be phrased in terms of pairs of nodes (both adjacent to the root) linked via $E_{\rho/n} \circ (E_{\rho'/n})^{-1}$ combinations.

With these straightforward modifications Lemma 5.5 carries over unchanged. For the translation from our GSO-formulae $\varphi(\overline{z})$ to its MSO-counterpart $\varphi^{\rightarrow}(z)$, we may now use exactly the same translation as in Section 5, with the minor difference that the quantifier free subformulae in variables $\overline{z}$ are translated into disjunctions over $P_{\mathfrak{A}/n}$ rather than $P_{\mathfrak{A}}$. We then obtain the analogue of Theorem 5.6, to the effect that

—for all consistent $\tilde{\tau}_n$-trees $\mathcal{T}$ with root λ and for all GSO-formulae $\varphi(\overline{z})$ in the normal form as considered above: $\mathfrak{D}(\mathcal{T}),\overline{d} \models \varphi(\overline{z})$ iff $\mathcal{T} \models \varphi^{\rightarrow}(\lambda)$.

As before, it follows that guarded bisimulation invariant φ translate into bisimulation invariant $\varphi^{\rightarrow}$, which by the Janin Walukiewicz Theorem are therefore equivalently expressible in the μ-calculus.

6.2.0.2 *From the $\boldsymbol{\mu}$-calculus to guarded fixed point logic.* . For the converse translation of formulae, which takes us from the trees $\mathcal{T}(\mathfrak{B},\overline{b})$ back to the $\mathfrak{B},\overline{b}$, we now need to deal with one additional issue. This concerns the fact that an arbitrary monadic second-order or L_μ formula over the tree may involve monadic second-order variables or monadic fixed points that contain the root. For L_μ we can simply eliminate these. Indeed, any L_μ formula over trees is logically equivalent to a Boolean combination of atomic propositions and L_μ formulae of the form $\Diamond\xi$. We may appeal to the same restricted syntax as used for the proof of Theorem 5.10, cf. [Niwinski and Walukiewicz 1997]. Note that the evaluation of an L_μ formula $\Diamond\xi$ at the root of a tree $\mathcal{T}$ only requires the evaluation of ξ in all sub-trees rooted at children of the root. For the $\mathcal{T}(\mathfrak{B},\overline{b})$ this means that the translations provided in Section 5, cf. in particular claim 5.11 in the proof of Theorem 5.10, can be used without changes for such ξ.

From the resulting tuple of formulae $\xi^{\leftarrow}_k(x_1,\dots,x_k)$ in μGF we may then obtain the correct translation of $\langle\rho/n\rangle\xi$ – where $\langle\rho/n\rangle$ is the modal operator associated with $E_{\rho/n}$ – as

$$\big(\langle\rho/n\rangle\xi\big)^{\leftarrow}(\overline{z}) := \bigvee_\alpha \big(\exists^{\neq}(x_1,\dots,x_k).\rho(\overline{z},x_1,\dots,x_k)\wedge\alpha(x_1,\dots,x_k)\big)\xi^{\leftarrow}_k(x_1,\dots,x_k),$$

where α ranges over all guards in variables $x_1,\dots,x_k$ for k such that $\mathrm{dom}(\rho) \subseteq \{1,\dots,k\}$.

We thus obtain the following extension of Theorem 5.10:

—For any τ-structure $\mathfrak{B}$ with n pairwise distinct parameters $\overline{b}$, and for any formula ψ in $\mathrm{L}_\mu[\tilde{\tau}_n]$: $\mathfrak{B} \models \psi^{\leftarrow}(\overline{z})$ iff $\mathcal{T}(\mathfrak{B}, \overline{b}), \lambda \models \psi$.

6.3 Main theorem: the case with arbitrary parameters

Putting the results of Section 6.2 together, we obtain the following variant of Theorem 5.12 for arbitrary formulae.

THEOREM 6.2. *Every formula of* GSO *that is invariant under guarded bisimulation is equivalent to a formula in* μGF*.*

In order to establish the full content of our characterisation theorem, Theorem 1.4, for formulae it remains to prove Proposition 3.5, which gives the semantic inclusion $\mu\mathrm{GF} \subseteq \mathrm{GSO}$ for formulae in arbitrary free variables. Recall that Proposition 3.5 states that any formula of μGF is logically equivalent to one in which all fixed points are strictly guarded, i.e., in which μ or ν operators are only applied to variable-guarded formulae.

Proof of Proposition 3.5. Consider a least fixed point formula $\psi(\overline{x}) := [\mathbf{lfp}\ X\overline{x} \,.\, \varphi](\overline{x})$. Inductively we assume that all fixed points within φ are in strictly guarded form. Looking at X-atoms in $\varphi(X, \overline{x})$, we distinguish the following:

(i) X-atoms in the scope of guarded first-order quantification.
(ii) X-atoms in the scope of least or greatest fixed point operators.
(iii) X-atoms at quantifier free level, $X\overline{z}$, $\overline{z} \subseteq \overline{x}$.

For occurrences of type (i) or (ii) we may replace X by its guarded part throughout the fixed point process. (The guarded part of X is the subset of X consisting of guarded tuples in X.) For type (ii) occurrences that are not of type (i) this relies on the inductive assumption that fixed points within φ are strictly guarded. As far as guarded tuples are concerned, even an atom of type (iii) can only evaluate to true for a guarded instantiation of $\overline{x}$ in $\varphi(\overline{x})$ if it would also evaluate to true for the guarded part of X rather than X. Hence, inductively, we find that the guarded part ψ^{g} of the fixed point $[\mathbf{lfp}\ X\overline{x} \,.\, \varphi]$ is definable as a strictly guarded fixed point, by

$$\psi^{\mathrm{g}}(\overline{x}) := [\mathbf{lfp}\ X\overline{x} \,.\, \varphi(X, \overline{x}) \wedge \mathrm{guarded}(\overline{x})](\overline{x}).$$

Let $\varphi(\psi^{\mathrm{g}}, X, \overline{x})$ be the result of substituting this formula for all occurrences of X in φ apart from those of type (iii). As $\psi(\overline{x})$ implies $\psi^{\mathrm{g}}(\overline{x})$ and by monotonicity, we clearly have

$$[\mathbf{lfp}\ X\overline{x} \,.\, \varphi(X, \overline{x})](\overline{x}) \equiv [\mathbf{lfp}\ X\overline{x} \,.\, \varphi(\psi^{\mathrm{g}}, X, \overline{x})](\overline{x}).$$

Note that the only remaining free occurrences of X in $\varphi(\psi^{\mathrm{g}}, X, \overline{x})$ are at the quantifier free level. It follows that the fixed point formula $[\mathbf{lfp}\ X\overline{x} \,.\, \varphi(\psi^{\mathrm{g}}, X, \overline{x})](\overline{x})$ is bounded. This means that the fixed point is attained within a uniformly bounded finite number of iterations, since there are only finitely many quantifier free types. By unravelling this finite number of iterations within GF we find that ψ is GF-definable from strictly guarded fixed points.

COROLLARY 6.3. $\mu\mathrm{GF} \subseteq \mathrm{GSO}$ *for formulae in arbitrary tuples of free variables.*

Together with Theorem 6.2 this finally gives the full content of our main theorem for arbitrary formulae: a formula of GSO is invariant under guarded bisimulation if and only if it is equivalent to a formula in μGF.

7. THE EXPRESSIVE POWER OF GSO

We conclude the paper with some results on the relationship between GSO and monadic second-order logic, and on the expressive power of guarded logics on graphs.

7.1 Guarded versus monadic second-order logic

We have seen that GSO is a strictly more expressive logic than MSO. However, GSO may nevertheless be viewed as a monadic logic, but over a different presentation of structures, via incidence graphs. We first explain the notion of the incidence graph of a relational structure and then prove Theorem 1.5.

Incidence graphs. Let $\tau = \{R_1, \dots, R_m\}$ be any finite relational vocabulary. With each τ-structure $\mathfrak{A} = (A, R_1, \dots, R_m)$ we associate a many-sorted structure $\mathfrak{A}^*$ with universe $A^* = (A \,\dot\cup\, R_1 \,\dot\cup \dots \dot\cup\, R_m)$. To put it differently, $\mathfrak{A}^*$ contains for every $R \in \tau$ and for every tuple $\overline{a} \in R$ a new element $e[R\overline{a}]$. The relations of $\mathfrak{A}^*$ are binary incidence relations $\mathrm{Inc}_{R,j}$ for $R \in \tau$, $1 \leq j \leq \mathit{arity}(R)$, recording identities between components of relational edges and base elements. More formally,

$$\begin{aligned}
\mathfrak{A}^* &= \big(A^*, (\mathrm{Inc}_{R,j})_{R\in\tau, 1\leq j\leq \mathit{arity}(R)}\big),\\
\text{where}\quad A^* &= A \,\dot\cup\, \dot{\textstyle\bigcup}_{R\in\tau} R,\\
\mathrm{Inc}_{R,j} &= \big\{(e,a) \in R \times A:\ \text{the } j\text{-th component of } e \text{ is } a\big\}.
\end{aligned}$$

The vocabulary of $\mathfrak{A}^*$ is denoted τ^*.

First-order logic, monadic second-order logic, etc., on incidence graphs are assumed to be typed, that is, element variables and set variables always range over a particular sort. Hence quantifiers are of the form $(\exists\, x \in A)$ or $(\exists\, z \in R_i)$ for element variables and $(\exists\, X \subseteq A)$ or $(\exists\, Y \subseteq R_i)$ for monadic second-order variables.

Let now $X \subseteq A^k$ be any *guarded* relation on $\mathfrak{A}$. We represent X by a tuple $\overline{X}$ of monadic predicates $X_{R,\rho}$ on $\mathfrak{A}^*$ where $R \in \tau$ and $\rho : \{1, \dots, k\} \to \{1, \dots, \mathit{arity}(R)\}$, and

$$X_{R,\rho} := \{e \in R : e = (e_1, \dots, e_s) \text{ such that } (e_{\rho(1)}, \dots, e_{\rho(k)}) \in X\}.$$

Conversely, every tuple $\overline{X}$ of monadic relations $X_{R,\rho} \subseteq R$ on $\mathfrak{A}^*$ represents a guarded k-ary relation

$$g(\overline{X}) := \bigcup_{R,\rho} \{(a_{\rho(1)}, \dots, a_{\rho(k)}) \in A^k : \overline{a} \in X_{R,\rho}\}$$

on $\mathfrak{A}$. Note that here we impose no consistency conditions on multiply represented tuples. We say that $\overline{X}$ is a representation of X if $g(\overline{X}) = X$. Every guarded relation on $\mathfrak{A}$ has a representation, but not necessarily a unique one.

The translation. The following proposition provides yet another natural characterisation of guarded second-order logic, as expressed in Theorem 1.5.

PROPOSITION 7.1. (i) *For every formula* $\psi(\overline{y}) \in \mathrm{GSO}(\tau)$ *there exists a formula* $\psi^*(\overline{y}) \in \mathrm{MSO}(\tau^*)$ *such that for every* τ*-structure* $\mathfrak{A}$ *and every guarded tuple* $\overline{a}$ *in* $\mathfrak{A}$

$$\mathfrak{A} \models \psi(\overline{a}) \iff \mathfrak{A}^* \models \psi^*(\overline{a}).$$

(ii) *Conversely, for every formula* $\psi(\overline{y}) \in \mathrm{MSO}(\tau^*)$, *with variables* $\overline{y}$ *ranging over the universe-sort, there exists a formula* $\psi^{\#}(\overline{y}) \in \mathrm{GSO}(\tau)$ *such that for every* τ*-structure* $\mathfrak{A}$ *and every guarded tuple* $\overline{a}$ *in* $\mathfrak{A}$

$$\mathfrak{A}^* \models \psi(\overline{a}) \iff \mathfrak{A} \models \psi^{\#}(\overline{a}).$$

PROOF. We first describe the translation from $\mathrm{GSO}(\tau)$ to $\mathrm{MSO}(\tau^*)$ required in (i).

(1) $(R\overline{y})^* := (\exists\, z \in R) \bigwedge_i \mathrm{Inc}_{R,i}(z, y_i)$.
(2) $(x = y)^* := (x = y)$.
(3) $(X\overline{y})^* := \bigvee_{R,\rho} (\exists\, z \in X_{R,\rho}) \bigwedge_i \mathrm{Inc}_{R,\rho(i)}(z, y_i)$.
(4) $(\neg\varphi)^* := \neg\varphi^*$ and $(\varphi \wedge \vartheta)^* := \varphi^* \wedge \vartheta^*$.
(5) $(\exists y \varphi)^* := (\exists\, y \in A)\varphi^*$.
(6) $(\exists X \varphi)^* := (\exists \overline{X}\,.\, \bigwedge_{R,\rho} X_{R,\rho} \subseteq R)\varphi^*$.

The converse translation from $\mathrm{MSO}(\tau^*)$ to $\mathrm{GSO}(\tau)$, as required for (ii), is defined as follows. Each variable z that ranges over a relational sort R (where R is k-ary) is replaced by a k-tuple $\overline{z}$ of variables. Similarly, each set variable $X \subseteq R$ is replaced by a k-ary relation symbol $X^{(k)}$.

(1) For $\varphi = \mathrm{Inc}_{R,j}(z, x)$, let $\varphi^{\#} := R\overline{z} \wedge z_j = x$.
(2) For a monadic second-order variable $X \subseteq A$, the atom Xy remains unchanged, whereas for $X \subseteq R$, Xz is replaced by $X^{(k)}\overline{z}$.
(3) Boolean combinations are translated in the obvious way.
(4) First-order quantification: $(\exists\, x \in A)\varphi$ is translated into $\exists x \varphi^{\#}$, whereas $(\exists\, z \in R)\varphi$ goes to $\exists \overline{z}(R\overline{z} \wedge \varphi^{\#})$.
(5) Second-order quantification: $(\exists\, X \subseteq A)\varphi$ corresponds to $\exists X \varphi^{\#}$, whereas $(\exists\, X \subseteq R)\varphi$ (with k-ary R) is replaced by a relativised quantification over the k-ary relation $X^{(k)}$, namely $\exists X^{(k)}(\forall \overline{y}(X^{(k)}\overline{y} \rightarrow R\overline{y}) \wedge \varphi^{\#})$.

The correctness of both translations is proved by straightforward inductions. □

Incidence graphs and query evaluation. The performance of a number of algorithms related to monadic second-order logic or to guarded logics depends strongly on the tree width of the input structure. Therefore it may often be interesting to work with incidence graphs rather than the usual relational structures. Indeed the tree width of $\mathfrak{A}^*$ is at most the tree width of $\mathfrak{A}$ plus one. On the other side, $\mathfrak{A}^*$ can have much smaller tree width than $\mathfrak{A}$ in cases where $\mathfrak{A}$ consists of 'tree-like' relations of high arity.

GSO on graphs. Besides the usual variant of monadic second-order logic on graphs, Courcelle has studied a more powerful variant, called MSO_2, allowing quantification not only over sets of vertices but also over sets of edges (see e.g. [Courcelle

1997] and the references there). It is not difficult to see that MSO_2 is actually equivalent to GSO on graphs. Courcelle's results show that, despite GSO being more expressive than MSO in general, there are a number of classes of graphs where the two logics are equivalent. This collapse occurs in particular over graphs of bounded degree, graphs of bounded tree-width, planar graphs and graphs with an excluded minor. A general result covering and generalising all these graph classes has recently been proved in [Courcelle].

THEOREM 7.2 COURCELLE. GSO *collapses to* MSO *on every class* $\mathcal{C}$ *of graphs that is closed under taking subgraphs and contains only* k*-sparse graphs, for some* $k \in \mathbb{N}$, *(i.e.* $|E| \leq k|V|$ *for every graph* $G = (V, E) \in \mathcal{C}$*).*

7.2 Guarded logics on graphs

Guarded logics extend modal logics to a general relational setting. Nevertheless the question arises whether, and to what extent, guarded logics are also more powerful on the structures on which modal logics operate, i.e., on transition systems or, equivalently, vertex and edge labelled graphs. Since ML, L_μ, and, more generally, standard modal bisimulation equivalence can only "see" the connected component of the regarded node(s), we restrict attention to *connected* graphs.

Atomic expansions of graphs. Consider graphs $G = (V, (E_a)_{a\in A}, (P_b)_{b\in B})$ in a vocabulary τ consisting of binary relations E_a and unary relations P_b. Let $\Phi := \{(\neg)E_a xy \ : \ a \in A\} \cup \{(\neg)E_a yx \ : \ a \in A\}$. The set T of *edge types* consists of all maximal consistent subsets of Φ that contain at least one positive literal. Note that the purely negative type of non-connected nodes is explicitly excluded. The edge type *realized* by a guarded pair of nodes u, v, $u \neq v$ of G is $t(u, v) := \{\varphi(x, y) \in \Phi \ : \ G \models \varphi(u, v)\}$.

Definition 7.3. The *atomic expansion* G^+ of the graph G is the expansion of G by new edge relations F_t for every edge type $t \in T$ and new unary relations Q_a for each $a \in A$. These are interpreted as

$$F_t := \{(u, v) \ : \ u \neq v \text{ and } (u, v) \text{ realizes } t\}$$
$$Q_a := \{u \ : \ G \models E_a uu\}.$$

Recall that we call a relational structure *connected* if its Gaifman graph is. That is $G = (V, (E_a)_{a\in A}, (P_b)_{b\in B})$ is connected if the undirected graph (V, E) with $E = \bigcup_{a\in A}(E_a \cup E_a^{-1})$ is connected in the usual sense. We first observe that guarded bisimulation on connected graphs is equivalent to ordinary bisimulation on the corresponding atomic expansions.

PROPOSITION 7.4. *Let* G, H *be connected graphs with nodes* $g \in G, h \in H$*. The following are equivalent.*

(i) $G, g \sim_g H, h$

(ii) $G^+, g \sim_g H^+, h$.

(iii) $G^+, g \sim H^+, h$.

PROOF.

(i) $\Leftrightarrow$ (ii). Note that a set is guarded in a graph G iff it is guarded in the atomic expansion G^+. It is immediate that any partial bijection $f : G \to H$ is a partial isomorphism for G and H iff it is one for G^+ and H^+. Consequently, a set of partial isomorphisms I is a guarded bisimulation between G and H iff it is one for G^+ and H^+.

(iii) $\Rightarrow$ (ii). Let $Z : G^+ \sim H^+$ be a bisimulation with $(g, h) \in Z$. Since G^+ and H^+ are connected, Z is in fact a total bisimulation.

Let I be the following collection of partial bijections (of sizes 1 or 2) from G^+ to H^+: $f = \{(x_1, y_1)\} \in I$ for all $(x_1, y_1) \in Z$; $f = \{(x_1, y_1), (x_2, y_2)\} \in I$ for all $(x_1, y_1), (x_2, y_2) \in Z$ such that $(x_1, x_2), (y_1, y_2) \in F_t$ for the same $t \in T$.

It is easily checked that these maps are in fact partial isomorphisms between G^+ and H^+. By assumption we also have $\{(g, h)\} \in I$.

Claim: I is a guarded bisimulation between G^+ and H^+.

We have to show that I satisfies the back and forth conditions. By symmetry it suffices to consider the forth property.

Let $f : X \to Y \in I$. We distinguish cases, according to whether the newly chosen guarded set X' has size one or two, and whether X and X' are disjoint. (The case $X = X'$ is trivial.)

Suppose $X' = \{x'\}$ is a singleton. We need to find an appropriate $y' \in H^+$ such that $f' : x' \mapsto y'$ is a partial isomorphism in I, compatible with f. If $x' \notin X$, we find a y' such that $(x', y') \in Z$, by totality of Z. Then $f' = \{(x', y')\} \in I$ is as desired. If $x' \in X$ we let f' be the restriction of f to X'.

Now let $X' = \{x'_1, x'_2\}$ contain two elements, of which one could occur in X too. We assume x'_1 to be that element (if any), and let $X'' = \{x'_1\}$. As above we argue that there is a y'_1 such that $f'' = \{(x'_1, y'_1)\} \in I$. X' is guarded in G^+, so there is a $t \in T$ such that $(x'_1, x'_2) \in F_t$. The forth property of the bisimulation Z yields a y'_2 that is connected to y'_1 via an F_t edge and $(x'_2, y'_2) \in Z$. It follows that $f = \{(x'_1, y'_1), (x'_2, y'_2)\} \in I$ is as required.

Note that for any pair of adjacent vertices $x \neq y$ there is a unique $t \in T$ such that $(x, y) \in F_t$. Clearly this t also determines the unique $t^{-1} \in T$ for which $(y, x) \in F_{t^{-1}}$. The chosen format is such that it is sufficient to consider only one of the edges involved, F_t or $F_{t^{-1}}$, between x and y without missing any atomic information.

(ii) $\Rightarrow$ (iii). Let $I : G \sim_g H$ be a guarded bisimulation connecting g to h. Define Z as the union of all graphs of the functions in I.

Claim: Z is a total bisimulation and $(g, h) \in Z$.

It is immediate that Z is total, that $(g, h) \in Z$ and that the atomic types of any $(x, y) \in Z$ match as required. We show that Z satisfies the back and forth conditions.

Let $(x, y) \in Z$ and let $(x, x') \in F_t$ for some $t \in T$. Since $(x, y) \in Z$, the function sending x to y already was a partial isomorphism in I. Further, as (x, x') is guarded there is an $f \in I$ with domain $\{x, x'\}$ that also sends x to y. For the image of y under f, denoted y', we have that $(y, y') \in Z$ by definition.

The case of E_a edges and the back condition are shown similarly. $\Box$

We next show that guarded logics on connected graphs essentially reduce to modal logics on the corresponding atomic expansions. This gives the answer to the question mentioned at the beginning of this section. On graphs, guarded logics extend modal logic precisely by the power to use all quantifier-free definable guarded binary predicates. For GF, the corresponding modal logic is not ML itself, but ML with the universal modality. We denote by ML+$\forall$ the extension of ML that permits to build new formulae $\forall\psi$ such that $G, g \models \forall\psi$ iff $G, h \models \varphi$ at all nodes h. For μGF we show the analogous statement for the fragment that allows only monadic fixed points, which we denote by $\mu\mathrm{GF}_{\mathrm{mon}}$. The question of whether this can be generalised to full μGF is open.

The following Lemma is a direct consequence of the syntax of $\mu\mathrm{GF}_{\mathrm{mon}}$ formulae.

LEMMA 7.5. *Every formula $\varphi(x, y) \in \mu\mathrm{GF}_{mon}$ is (on graphs) equivalent to a Boolean combination of quantifier free formulae and formulae in at most one free variable.*

We will present formulae with two free variables in a disjunctive normal form

$$\varphi(x, y) = \bigvee_i \big(\varphi_i^x(x) \wedge \varphi_i^y(y) \wedge \varphi_i^{xy}(x, y)\big)$$

where φ_i^{xy} contains only atoms whose free variables are exactly x and y, and $\varphi_i^{xy} \models x \neq y$.

PROPOSITION 7.6. *A monadic query on connected graphs is definable in* GF *if and only if it is definable in* ML + $\forall$ *over the corresponding atomic expansions. Similarly, a monadic query on connected graphs is definable in $\mu\mathrm{GF}_{mon}$ if and only if it is definable in the modal μ-calculus on the corresponding atomic expansions.*

PROOF. Let $\varphi(x) \in \mathrm{GF}[\tau]$ resp. $\varphi(x) \in \mu\mathrm{GF}_{\mathrm{mon}}$.

Note that any subformula of φ with two free variables necessarily occurs in the scope of a quantifier. This observation and Lemma 7.5 enable an explicit transformation.

Both translations require the following inductive rules. The modal quantifiers $[t]$ and $\langle t\rangle$ w.r.t. the new edge relations F_t are sufficient to simulate local guarded quantification[3].

$$\begin{aligned}
(\neg\psi)^+ &= \neg(\psi^+)\\
(\psi_1 \wedge \psi_2)^+ &= \psi_1^+ \wedge \psi_2^+\\
(\psi_1 \vee \psi_2)^+ &= \psi_1^+ \vee \psi_2^+\\
(P_b(y))^+ &= P_b\\
(E_a(y, y))^+ &= Q_a\\
(\forall y.\psi(xy))^+ &= \bigvee_i \Big((\psi_i^x(x))^+ \wedge \textstyle\bigwedge_{\{t\in T\,:\, t\models\psi_i^{xy}\}}\big([t](\psi_i^y(y))^+\big)\Big)
\end{aligned}$$

We do not assume negation normal form and assume that the $\neg$ operator is used to convert between $\exists$ and $\forall$. The translation from GF into ML + $\forall$ is completed

[3]We abbreviate guarded quantifications $(\exists y.\alpha(xy))\varphi(xy)$ as $\exists y.\psi(xy)$, i.e. $\psi(xy) := \alpha(xy) \wedge \varphi(xy)$, etc.

with the following rules.

$$
\begin{aligned}
(\forall x.\psi(x))^+ &= \forall((\psi(x))^+) \\
(\forall xy.\psi(xy))^+ &= \forall(\forall y.\psi(xy))^+
\end{aligned}
$$

The translation from μGF into L_μ uses a fixed-point iteration to quantify over all nodes. Connectedness of the given graphs is necessary for capturing the universal modality in L_μ.

$$
\begin{aligned}
(Xx)^+ &= X \\
([\mathbf{lfp}\ Xx\,.\,\psi](x))^+ &= \mu X.(\psi(X,x)^+) \\
(\exists x.\psi(x))^+ &= \mu X.\big(\psi(x)^+ \vee \textstyle\bigvee_{t\in T} \langle t\rangle X\big) \\
(\exists xy.\psi(xy))^+ &= \mu X.\big((\forall y.\psi(xy))^+ \vee \textstyle\bigvee_{t\in T} \langle t\rangle X\big)
\end{aligned}
$$

By a straightforward induction one shows that for all τ-graphs G and all nodes g, $G \models \varphi(g)$ iff $G^+, g \models \varphi^+$. For the other direction it is immediate that any monadic query in $\mathrm{ML} + \forall$ resp. L_μ on atomic expansions of graphs can be converted into a GF resp. $\mu\mathrm{GF}_{\mathrm{mon}}$ query on the original graphs. $\Box$

ACKNOWLEDGMENT

We are grateful to H.-D. Ebbinghaus and J. Flum for comments received on an earlier version of this paper, which helped to clarify some points.

REFERENCES

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

Arnold, A. and Niwiński, D. 2001. *Rudiments of μ-calculus.* North Holland.

Berwanger, D. and Grädel, E. 2001. Games and model checking for guarded logics. In *to appear in Proceedings of LPAR 2001.*

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

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

Bradfield, J. and Stirling, C. 2001. Modal logics and mu-calculi. In *Handbook of Process Algebra*, J. Bergstra, A. Ponse, and S. Smolka, Eds. Elsevier, 293–332.

Courcelle, B. The monadic second-order logic of graphs XIV: Uniformly sparse graphs and edge set quantifications. Submitted for publication.

Courcelle, B. 1997. On the expression of graph properties in some fragments of monadic second-order logic. In *Descriptive Complexity and Finite Models*, N. Immerman and P. Kolaitis, Eds. DIMACS Series in Discrete Mathematics, vol. 31. AMS, 33–62.

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

Ebbinghaus, H.-D. and Flum, J. 1999. *Finite Model Theory*, 2nd edition ed. Springer.

Gottlob, G., Grädel, E., and Veith, H. to appear. Datalog LITE: A deductive query language with linear time model checking. *ACM Transactions on Computational Logic*.

Grädel, E. 1999. On the restraining power of guards. *Journal of Symbolic Logic 64*, 1719–1742.

Grädel, E. 2001a. Guarded fixed point logics and the monadic theory of countable trees. *to appear in Theoretical Computer Science.*

Grädel, E. 2001b. Why are modal logics so robustly decidable? In *Current Trends in Theoretical Computer Science. Entering the 21st Century*, G. Paun, G. Rozenberg, and A. Salomaa, Eds. World Scientific, 393–498.

Grädel, E., Hirsch, C., and Otto, M. 2000. Back and forth between guarded and modal logics. In *Proc. 15th IEEE Symp. on Logic in Computer Science*. 217–228.

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

Harel, D., Kozen, D., and Tiuryn, J. 2000. *Dynamic Logic*. MIT Press.

Janin, D. and Walukiewicz, I. 1996. 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. Springer-Verlag, 263–277.

Kozen, D. 1983. Results on the propositional μ-calculus. *Theoretical Computer Science 27*, 333–354.

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

Otto, M. Modal and guarded characterisation theorems over finite transition systems. 2001.

Rosen, E. 1997. Modal logic over finite structures. *Journal of Logic, Language and Information 6*, 427–439.

Thomas, W. 1990. Automata on infinite objects. In *Handbook of Theoretical Computer Science*, J. van Leeuwen, Ed. Vol. B. Elsevier, Amsterdam, 133–191.

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

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

Received December 2000; revised October 2001; accepted February 2002