DECIDABILITY RESULTS FOR THE BOUNDEDNESS PROBLEM

ACHIM BLUMENSATH, MARTIN OTTO, AND MARK WEYER

Fachbereich Mathematik, Technische Universität Darmstadt
e-mail address: blumensath@mathematik.tu-darmstadt.de

Fachbereich Mathematik, Technische Universität Darmstadt
e-mail address: otto@mathematik.tu-darmstadt.de

note to layout editor: the third author is not affiliated to any institution, in particular not to the same one as the other authors
e-mail address: mark@weyer-zuhause.de

Abstract. We prove decidability of the boundedness problem for monadic least fixed-point recursion based on positive monadic second-order (MSO) formulae over trees. Given an MSO-formula $\varphi(X, x)$ that is positive in X, it is decidable whether the fixed-point recursion based on φ is spurious over the class of all trees in the sense that there is some uniform finite bound for the number of iterations φ takes to reach its least fixed point, uniformly across all trees. We also identify the exact complexity of this problem. The proof uses automata-theoretic techniques.

This key result extends, by means of model-theoretic interpretations, to show decidability of the boundedness problem for MSO and guarded second-order logic (GSO) over the classes of structures of fixed finite tree-width. Further model-theoretic transfer arguments allow us to derive major known decidability results for boundedness for fragments of first-order logic as well as new ones.

1. Introduction

In applications one frequently employs tailor-made logics to achieve a balance between expressive power and algorithmic manageability. Adding fixed-point operators to weak logics turned out to be a good way to achieve such a balance. Think, for example of the addition of transitive closure operators or more general fixed-point constructs to database query languages, or of various fixed-point defined reachability or recurrence assertions to logics used in verification, like linear or branching time temporal logics or the modal μ-calculus. Fixed-point operators introduce a measure of relational recursion and typically boost expressiveness in the direction of more dynamic and less local properties. They offer relational recursion based on the iteration of relation updates that are definable in the underlying logic. We here primarily consider monadic least fixed points, based on formulae $\varphi(X, x)$ that are monotone (positive) in the monadic recursion variable X. On a fixed structure $\mathfrak{A}$, any such φ induces a monotone operation $F_\varphi : P \mapsto \{ a \in \mathfrak{A} \mid \mathfrak{A} \models \varphi(P, a) \}$ on monadic relations $P \subseteq A$. The least fixed point of this operation over $\mathfrak{A}$, denoted as $\varphi^\infty(\mathfrak{A})$, is also the first stationary point of the monotone, ordinal-indexed iteration sequence

LOGICAL METHODS
IN COMPUTER SCIENCE

DOI:10.2168/LMCS-???

of stages $\varphi^\alpha(\mathfrak{A})$ starting from $\varphi^0(\mathfrak{A}) := \emptyset$, with updates $\varphi^{\alpha+1}(\mathfrak{A}) := F_\varphi(\varphi^\alpha(\mathfrak{A}))$ and unions in limits. The least α for which $\varphi^{\alpha+1}(\mathfrak{A}) = \varphi^\alpha(\mathfrak{A})$ is called the closure ordinal for this fixed-point iteration on $\mathfrak{A}$.

For a concrete fixed-point process it may be hard to tell whether the recursion employed is crucial or whether it is spurious and can be eliminated. Indeed this question comes in two versions: (a) one can ask whether a resulting fixed point is also uniformly definable in the base logic without fixed-point recursion (purely an expressiveness issue); (b) one may also be interested to know whether the given fixed-point iteration terminates within a uniformly bounded finite number of steps (an algorithmic issue, concerning the dynamics of the fixed-point recursion rather than its result).

The boundedness problem $\mathrm{BDD}(L, \mathcal{C})$ for a class of formulae L and a class of structures $\mathcal{C}$ concerns question (b): to decide, for a given formula $\varphi \in L$, whether there is a finite upper bound on its closure ordinal, uniformly across all structures $\mathfrak{A} \in \mathcal{C}$. We call such fixed-point iterations, or φ itself, *bounded over $\mathcal{C}$.*

Interestingly, for first-order logic, as well as for many natural fragments, the two questions concerning eliminability of least fixed points coincide – at least over the class of all structures. By a classical theorem of Barwise and Moschovakis [2], the only way that the fixed point $\varphi^\infty(\mathfrak{A})$ can be first-order definable for every $\mathfrak{A}$, is that there is some finite α for which $\varphi^\infty(\mathfrak{A}) = \varphi^\alpha(\mathfrak{A})$ for all $\mathfrak{A}$. The converse is clear from the fact that the unfolding of the iteration to any fixed finite depth α is easily mimicked in FO.

In other cases – and even for FO over other, restricted classes of structures, e.g., in finite model theory – the two problems can indeed be distinct, and of quite independent interest.

We here deal with the boundedness issue. Boundedness (even classically, over the class of all structures, and for just monadic fixed points as considered above) is undecidable for most first-order fragments of interest (see, e.g., [20]). Notable exceptions are monadic boundedness for positive existential formulae (Datalog) [8], for modal formulae [24], and for (a restricted class of) universal formulae without equality [25].

One common feature of these decidable cases of the boundedness problem is that the fragments concerned have a kind of tree-model property (not just for satisfiability in the fragment itself, but also for the fixed points and for boundedness). This is obvious for the modal fragment [24], but clearly also true for positive existential FO (derivation trees for monadic Datalog programs can be turned into models of bounded tree-width), and similarly also for the restricted universal fragment in [25].

Motivated by this observation, [23] has made a first significant step in an attempt to analyse the boundedness problem from the opposite perspective, varying the class of structures rather than the class of formulae. The hope is that this approach could go beyond an ad-hoc exposition of the decidability of the boundedness problem for individual syntactic fragments, and offer a unified model-theoretic explanation instead. [23] shows that boundedness is decidable for *all* monadic fixed points in FO over the class of all acyclic relational structures. Technically [23] expands on modal and locality-based proof ideas and reductions to the monadic second-order theory of trees from [24, 25] that also rest on the availability of a Barwise–Moschovakis equivalence. These techniques do not seem to extend to either the class of all trees (where Barwise–Moschovakis fails) or to bounded tree-width (where certain simple locality criteria fail).

The present investigation offers another step forward in the alternative approach to the boundedness problem, on a methodologically very different note. Its most important novel feature may be that it deals with a setting where neither locality nor Barwise–Moschovakis are available. On the one hand, the class of formulae considered is extended from first-order logic FO to full monadic second-order logic MSO – a leap which greatly increases the robustness of the results w.r.t. interpretations, and hence their model-theoretic impact. On the other hand, automata are crucially used and the underlying structures are restricted to trees. Using MSO-interpretations it follows that the boundedness problem for MSO is decidable over any MSO-definable class of bounded tree-width, and similarly even for guarded second-order logic GSO instead of MSO.

These ramifications demonstrate the strength and unifying explanatory power of our main decidability result in the wider context of the boundedness issue. One of our strongest concrete decidability results concerns the boundedness problem for GSO over GSO-definable classes of bounded tree-width, cf. Corollary 11.5. This, in its turn, encompasses all the major, previously known decidability results for natural fragments of FO and, furthermore, settles decidability of boundedness for the guarded fragment GF. Equally importantly it goes a long way to explain the perceived dichotomy between the many undecidability results, which may typically be understood in terms of reductions from the tiling problem over suitably grid-like structures, and the comparatively rare cases of decidability, which can now be systematically linked to some generalised tree-model property.

Among the classical and previously known decidability results for the boundedness of (systems of) monadic least fixed points, which can be integrated into this new picture, are those for

- monadic Datalog, or systems of monadic least fixed points for the purely existential–positive fragment of first-order logic, [8];
- dually, (systems of) monadic least fixed points in the purely universal-negative fragment of first-order logic (which may equivalently be phrased in terms of the boundedness for greatest fixed points for Datalog or for existential–positive first-order), [25];
- modal logic, [24];
- monadic least fixed points for unconstrained FO in restriction to the class of all acyclic relational structures, [23].

Our decidability results are based on a reduction of the monadic boundedness problem to the *limitedness problem* for *weighted parity automata,* whose decidability is due to Colcombet and Löding [7] (cf. Theorems 7.2 and 7.3 below). This reduction introduces a rather sophisticated annotation (of ternary tree structures) that records dependencies between the stages of a fixed-point iteration over these tree structures; it is established that, subject to a limitedness condition on a related cost function, these annotations can serve as certificates for boundedness.

The overall structure of the paper is as follows. We divide the material into two major parts: the first part, comprising Sections 3–7, is devoted to the development of the new techniques and leads up to the core technical result: the decidability of the boundedness problem for MSO on the class of all ternary trees through reduction to the limitedness problem for a certain class of automata. The ramifications of this result are investigated in the second half of the paper. Sections 8–11 develop transfer and reduction arguments that allow us to make links with previously known decidability results and to derive several new concrete decidability results. Section 12, finally, discusses complexity issues.

2. Preliminaries

We assume some familiarity with basic concepts of logic as can be found, e.g., in [12]. Throughout the paper we assume that all vocabularies are finite and that they contain only relation symbols and constant symbols, but no function symbols.

Consider a second-order formula $\varphi(X, \bar{x})$ with free variables as indicated in an underlying vocabulary τ. Suppose that $\varphi(X, \bar{x})$ is positive in the r-ary second-order variable X and $\bar{x} = (x_1, \dots, x_r)$ is a matching tuple of free first-order variables. Any X-positive formula of this format induces, over every τ-structure $\mathfrak{A}$, an operation on the power set of A^r:

$$P \mapsto \varphi(\mathfrak{A}, P) := \{\, \bar{a} \in A^r \mid (\mathfrak{A}, P, \bar{a}) \models \varphi \,\}\,.$$

As φ is X-positive, this operation is monotone ($P \subseteq P'$ implies $\varphi(\mathfrak{A}, P) \subseteq \varphi(\mathfrak{A}, P')$) and hence possesses a unique least fixed point, which we denote as $\varphi^\infty(\mathfrak{A})$. This least fixed point is obtained as the limit of the monotone sequence of inductive stages $\varphi^\alpha(\mathfrak{A})$ induced by φ on $\mathfrak{A}$. These stages are defined by transfinite induction, for all ordinals α, according to:

$$\begin{aligned} \varphi^0(\mathfrak{A}) &:= \emptyset\,, \\ \varphi^{\alpha+1}(\mathfrak{A}) &:= \varphi(\mathfrak{A}, \varphi^\alpha(\mathfrak{A}))\,, \\ \varphi^\delta(\mathfrak{A}) &:= \bigcup_{\alpha<\delta} \varphi^\alpha(\mathfrak{A}) \quad \text{for limits } \delta\,. \end{aligned}$$

The *finite stages* $\varphi^\alpha(\mathfrak{A})$, for $\alpha < \omega$, are uniformly definable by formulae, which we also denote by φ^α, obtained from $\varphi(X, \bar{x})$ by iterated substitution of φ for X in φ. Letting $\varphi[\psi(\bar{x})/X]$ stand for the result of replacing all free occurrences of X in atoms $X\bar{y}$ in φ by $\psi(\bar{y})$, we obtain formulae φ^α for $\alpha < \omega$, by

$$\varphi^0 := \bot \quad \text{and} \quad \varphi^{\alpha+1} := \varphi[\varphi^\alpha(\bar{x})/X]\,.$$

Clearly, for finite α, $\varphi^\alpha \in \mathrm{MSO}$ for $\varphi \in \mathrm{MSO}$, and similarly for all natural fragments of first- and second-order logic that are closed under this substitution operation. It is easy to see that φ^α defines the stage $\varphi^\alpha(\mathfrak{A})$ for finite α, uniformly across all $\mathfrak{A}$. We therefore need not distinguish between the two readings of $\varphi^\alpha(\mathfrak{A})$ for *finite* α. For infinite α, on the other hand, we do not regard φ^α as a formula (it would in general have to be a formula in some infinitary extension of the base logic), but only allow $\varphi^\alpha(\mathfrak{A})$ as shorthand notation for the corresponding stage of φ over $\mathfrak{A}$.

Because of monotonicity, $\varphi^\infty(\mathfrak{A}) = \bigcup_\alpha \varphi^\alpha(\mathfrak{A}) = \varphi^\gamma(\mathfrak{A})$ for the least ordinal γ for which $\varphi^{\gamma+1}(\mathfrak{A}) = \varphi^\gamma(\mathfrak{A})$. This ordinal γ is called the *closure ordinal* for φ on $\mathfrak{A}$, denoted $\|\varphi\|_{\mathfrak{A}}$. The *stage* of an individual $\bar{a} \in \varphi^\infty(\mathfrak{A})$ is the least ordinal α such that $\bar{a} \in \varphi^{\alpha+1}(\mathfrak{A})$; therefore, the closure ordinal could also be described as the least ordinal greater than the stages of all members of the fixed point $\varphi^\infty(\mathfrak{A})$.

The closure ordinal can in general only be bounded, for simple cardinality reasons, by the successor cardinal of the cardinality of $\mathfrak{A}$, or by $|A|^r$ for finite $\mathfrak{A}$.

For instance, the fixed-point induction based on $\varphi(X, x) = \forall y(Ryx \to Xy)$ yields as its fixed point over $\mathfrak{A} = (A, R)$ the set of elements $a \in A$ that are well-founded w.r.t. R; over the well-ordering $\mathfrak{A} = (\alpha, <)$, the closure ordinal is α. In fact, $\varphi^\infty(\mathfrak{A}) = \alpha$; the stage of $\beta \in \alpha$ is β.

The fixed-point induction based on φ, or for simplicity: φ itself, is said to be bounded if, for some finite $\alpha < \omega$, $\|\varphi\|_{\mathfrak{A}} \leq \alpha$ for all $\mathfrak{A}$. Similarly, φ is bounded over the class $\mathcal{C}$ if, for some $\alpha < \omega$, $\|\varphi\|_{\mathfrak{A}} \leq \alpha$ for all $\mathfrak{A} \in \mathcal{C}$.

Definition 2.1. (a) Let φ be a formula over τ, positive in X, and let $\alpha < \omega$. We say that φ is *bounded by* α over a class $\mathcal{C}$ if $\varphi^{\alpha}(\mathfrak{A}) = \varphi^{\alpha+1}(\mathfrak{A})$, for all $\mathfrak{A} \in \mathcal{C}$. We call φ *bounded* over $\mathcal{C}$ if it is bounded by some $\alpha < \omega$.

(b) The *boundedness problem* for a logic L over a class $\mathcal{C}$ is the problem to decide, given a formula $\varphi \in L$, whether φ is bounded over $\mathcal{C}$. We denote this decision problem as $\mathrm{BDD}(L, \mathcal{C})$.

The *monadic boundedness problem* is the corresponding problem where we only consider formulae φ with *monadic* variables X. We denote it as $\mathrm{BDD}^1(L, \mathcal{C})$.

If $\mathcal{C}$ is the class of all structures, we just write $\mathrm{BDD}(L)$ or $\mathrm{BDD}^1(L)$.

A vocabulary τ is called a *tree vocabulary,* if τ consists of one binary relation symbol E and, otherwise, only of constant symbols and unary relation symbols. A τ-structure $\mathfrak{T}$ is called a *tree structure,* or *tree* for short, if $E^{\mathfrak{T}}$ is a symmetric, acyclic, and connected relation on T. In particular, tree structures are undirected.

In Part I of the paper, we shall exclusively look at $\mathrm{BDD}^1(\mathrm{MSO}, \mathcal{T})$ for the class of MSO-formulae $\varphi(X, x)$ suitable for monadic fixed points (positive in the monadic variable X) over the class $\mathcal{T}$ of all tree structures and some of its subclasses. We refer to this core problem as the boundedness problem for MSO over trees for short.

Theorem 2.2 (Main theorem)**.** $\mathrm{BDD}^1(\mathrm{MSO}, \mathcal{T})$*, the monadic boundedness problem for* MSO *over the class of all tree structures, is decidable.*

In Part II we employ model-theoretic interpretations and similar transfer arguments to deduce from this result the decidability of many other boundedness problems. In particular, we obtain new proofs of many previous decidability results for boundedness, as well as some new results like the decidability for the guarded fragment of first-order logic and for full guarded second-order logic over structures of bounded tree-width.

Part I. The main result

In this first part we prove the main technical result, the decidability of the monadic boundedness problem for MSO on the class of all ternary trees. The ramifications of this result will then be investigated in the second half of the paper.

To help the reader through the later technicalities, we start with a simplified outline of the proof idea towards the main theorem. The key idea is to derive, for every formula φ, a bound $N = N(\varphi)$ that provides a uniform strict upper bound on the closure ordinals $\|\varphi\|_{\mathfrak{T}}$ over any tree structure $\mathfrak{T}$ in case φ *is* bounded. Then boundedness of φ is equivalent to the unsatisfiability of $\varphi^N \wedge \neg\varphi^{N-1}$ (over the class of all tree structures $\mathfrak{T}$). In other words, a formula which (on the class of all trees) is not bounded by this number N is not bounded at all. To reason towards such a uniform bound N, assume that for some tree $\mathfrak{T}$, some node v enters the fixed point in stage N. Then $(\mathfrak{T}, \varphi^N(\mathfrak{T}), v) \models \varphi$ but $(\mathfrak{T}, \varphi^{N-1}(\mathfrak{T}), v) \not\models \varphi$. Using a Feferman–Vaught style lemma (cf. Proposition 3.2), this change in the status of φ can be traced back to some other node w such that $(\mathfrak{T}, \varphi^N(\mathfrak{T}), w) \models Xx$ but $(\mathfrak{T}, \varphi^{N-1}(\mathfrak{T}), w) \not\models Xx$, which means that w entered the fixed point in stage $N - 1$. In this way we obtain a path of dependencies which travels through the tree and at places decreases the stage by 1. In a chain of N such jumps, we conclude that, if N is large in comparison to the number of types used in the Feferman–Vaught style lemma, then the path has repetitions and we can use a pumping argument to produce trees where some node enters the fixed point at arbitrarily large stages. Consequently, φ is unbounded.

The actual proof has to deal with further difficulties, so it does not exactly follow this outline. One difficulty is that a pumping lemma essentially requires that (in some very loose sense) we only use regular properties. In particular, we have to weaken the counting of stages and, consequently, we will slightly relax the concept of a dependency. Also, it is not sufficient to consider a single dependency path: we have to do the pumping such that it works for all paths simultaneously. Fortunately, there is already a suitable pumping theorem for a certain kind of weighted automaton that we can reduce our problem to. The main part of this paper describes this highly non-trivial reduction.

Convention. For technical reasons we choose in the following not to distinguish formally between (assignments to) free first and second-order variables (and interpretations of) constant or relation symbols. For instance, we shall often regard x and X, which in usual parlance occur free in $\varphi(X, x)$, as part of the vocabulary, and think of assignments $a \in A$ and $P \subseteq A$ over some $\mathfrak{A}$ in terms of the expansion $(\mathfrak{A}, P, a)$ of $\mathfrak{A}$.

3. A Feferman–Vaught theorem for positive types

For a vocabulary τ, we denote by $\mathrm{MSO}^n[\tau]$ the set of all MSO-formulae over τ with quantifier rank at most n (we count both first- and second-order quantifiers). If $X \in \tau$ is a unary predicate we write $\mathrm{MSO}^n_X[\tau]$ for the subset of all formulae where the predicate X occurs only positively. Recall that, for finite vocabularies τ, $\mathrm{MSO}^n[\tau]$, and hence also $\mathrm{MSO}^n_X[\tau]$, is finite up to logical equivalence.

Definition 3.1. Let τ be a vocabulary and $X \in \tau$. The *X-positive n-type* of a τ-structure $\mathfrak{A}$ is the set

$$\mathrm{tp}^n_X(\mathfrak{A}) := \{\, \varphi \in \mathrm{MSO}^n_X[\tau] \mid \mathfrak{A} \models \varphi \,\}\,.$$

We write $\mathrm{Tp}^n_X[\tau]$ for the set of all X-positive n-types of τ-structures.

Let $\mathfrak{T}_1$ and $\mathfrak{T}_2$ be tree structures. If T_1 and T_2 are disjoint, and if furthermore no constant symbol is interpreted in both trees, then we define a concatenation operation as follows: let c_1 and c_2 be constant symbols from the structures $\mathfrak{T}_1$ and $\mathfrak{T}_2$, respectively. Then we denote by $\mathfrak{T}_1 +_{c_1,c_2} \mathfrak{T}_2$ the tree obtained from the disjoint union of the trees $\mathfrak{T}_1$ and $\mathfrak{T}_2$ by adding an edge between $c_1^{\mathfrak{T}_1}$ and $c_2^{\mathfrak{T}_2}$. Note that every finite tree can be constructed from one-element trees using this operation and reduct operations.

If $\mathfrak{T}$ is a tree and vw an edge of $\mathfrak{T}$, then removing vw from $\mathfrak{T}$ produces two disjoint trees. Of these, we denote the one containing the vertex v by $\mathfrak{T}_{vw}$. Note that, if there are constants c and d for v and w, then $\mathfrak{T} = \mathfrak{T}_{vw} +_{c,d} \mathfrak{T}_{wv}$. If c is a constant symbol not interpreted by $\mathfrak{T}_{vw}$, then we set $\mathfrak{T}_{vw,c} := (\mathfrak{T}_{vw}, v)$, where the expansion interprets c by v.

We will frequently need a derived operation: let $\mathfrak{T}_1$ and $\mathfrak{T}_2$ be trees such that T_1 and T_2 are disjoint, and suppose that there is exactly one constant symbol c that is interpreted both in $\mathfrak{T}_1$ and in $\mathfrak{T}_2$. Let d be a constant symbol which is interpreted in neither. Then we denote by $\mathfrak{T}_1 \curlywedge_c \mathfrak{T}_2$ the reduct of $\mathfrak{T}_1 +_{c,d} \mathfrak{T}_2[d/c]$ that expels d from the vocabulary ($\mathfrak{T}_2[d/c]$ denotes the structure obtained from $\mathfrak{T}_2$ by renaming the constant symbol c to d).

Intuitively, c denotes the root of (directed versions of) the respective trees, and $\curlywedge_c$ appends its second argument as a new subtree below the root of its first argument. For a more uniform treatment, we allow the empty tree $\triangle$ as a neutral second argument to $\curlywedge_c$, and we use $\triangle$ also for its type.

Proposition 3.2. *For every $n < \omega$, there is a binary operation $\oplus^n_{c_1,c_2}$ on X-positive n-types such that, for all trees $\mathfrak{T}_1, \mathfrak{T}_2$ for which $\mathfrak{T}_1 +_{c_1,c_2} \mathfrak{T}_2$ is defined, we have*

$$\mathrm{tp}^n_X(\mathfrak{T}_1 +_{c_1,c_2} \mathfrak{T}_2) = \mathrm{tp}^n_X(\mathfrak{T}_1) \oplus^n_{c_1,c_2} \mathrm{tp}^n_X(\mathfrak{T}_2)\,.$$

Furthermore, $\oplus^n_{c_1,c_2}$ is monotone:

$$t_1 \subseteq t'_1 \text{ and } t_2 \subseteq t'_2 \quad \text{implies} \quad t_1 \oplus^n_{c_1,c_2} t_2 \subseteq t'_1 \oplus^n_{c_1,c_2} t'_2\,.$$

Finally, $t_1 \oplus^n_{c_1,c_2} t_2$ is computable from n, t_1, and t_2.

Proof. Computability of the operation will be evident, once we show how to compute with types in an effective way. For this sake, note that we can represent an n-type by a finite set of formulae where all maximal boolean combinations are in disjunctive normal form without repetition of clauses or of literals in clauses.

We proceed by induction on n. Assume that we already know how to compute $\oplus^m_{c_1,c_2}$ for all $m < n$ and all vocabularies. For convenience, we set

$$\mathfrak{T} := \mathfrak{T}_1 +_{c_1,c_2} \mathfrak{T}_2\,, \quad t_1 := \mathrm{tp}^n_X(\mathfrak{T}_1)\,, \quad t_2 := \mathrm{tp}^n_X(\mathfrak{T}_2)\,, \quad \text{and} \quad t := \mathrm{tp}^n_X(\mathfrak{T})\,.$$

We will describe t solely in terms of n, t_1, t_2, and the operations $\oplus^m_{c_1,c_2}$ with $m < n$. Each formula in an X-positive n-type is a positive boolean combination of atoms, negated atoms, and formulae of the form $\exists y\varphi$, $\forall y\varphi$, $\exists Y\varphi$, and $\forall Y\varphi$, where y is a first-order variable and Y is a set variable. Whether the full formula belongs to t is clearly determined by whether the individual formulae in the positive boolean combination do. Also, as the boolean combinations are positive, monotonicity is preserved. Hence it suffices to consider subformulae of the above form.

In the following we explicitly treat the cases of atomic and negated atomic formulae and of $\exists y\varphi$ and $\forall Y\varphi$. The remaining cases $\exists Y\varphi$ and $\forall y\varphi$ can be handled using combinations of the techniques used in these cases.

First, we consider atoms and negated atoms. Each (negated) atom that only uses constants from $\mathfrak{T}_i$ occurs in t iff it occurs in t_i. It remains to consider (negated) atoms involving constants from both $\mathfrak{T}_1$ and $\mathfrak{T}_2$. As E is the only relation symbol of arity more than 1, such an atom must be of the form $c = d$ or Ecd where, without loss of generality, c is from the vocabulary of $\mathfrak{T}_1$ and d from the vocabulary of $\mathfrak{T}_2$. In this case, we always have $c = d \notin t$ and, hence, $\neg(c = d) \in t$; so

$$Ecd \in t \quad \text{iff} \quad \neg Ecd \notin t \quad \text{iff} \quad c = c_1 \in t_1 \text{ and } d = c_2 \in t_2\,.$$

Next, let us consider a formula of the form $\exists y\varphi$ with $m := \mathrm{qr}(\varphi) < n$. We make use of $\oplus^m_{c_1,c_2}$. Let t'_1 and t'_2 be the X-positive m-types of $\mathfrak{T}_1$ and $\mathfrak{T}_2$, that is, $t'_1 = t_1 \cap \mathrm{MSO}^m_X$ and $t'_2 = t_2 \cap \mathrm{MSO}^m_X$. Further, let S_1 be the set of X-positive m-types of expansions of $\mathfrak{T}_1$ by some $a \in T_1$ interpreted for y, and let S_2 be the respective set of types of expansions of $\mathfrak{T}_2$. Clearly, $\exists y\varphi \in t$ iff $\varphi \in \mathrm{tp}^m_X(\mathfrak{T}, a)$ for some $a \in T$. For $a \in T_1$ and $t''_1 := \mathrm{tp}^m_X(\mathfrak{T}_1, a)$, the inductive hypothesis implies that

$$\begin{aligned}\mathrm{tp}^m_X(\mathfrak{T}, a) &= \mathrm{tp}^m_X(\mathfrak{T}_1 +_{c_1,c_2} \mathfrak{T}_2,\ a) \\ &= \mathrm{tp}^m_X((\mathfrak{T}_1, a) +_{c_1,c_2} \mathfrak{T}_2) \\ &= \mathrm{tp}^m_X(\mathfrak{T}_1, a) \oplus^m_{c_1,c_2} \mathrm{tp}^m_X(\mathfrak{T}_2) = t''_1 \oplus^m_{c_1,c_2} t'_2 \,.\end{aligned}$$

Note that $t''_1 \in S_1$. The case where $a \in T_2$ is similar. It follows that

$$\begin{aligned}\exists y\varphi \in t \quad \text{iff} \quad & \varphi \in t''_1 \oplus^m_{c_1,c_2} t'_2\,,\ \text{for some } t''_1 \in S_1\,, \\ \text{or } & \varphi \in t'_1 \oplus^m_{c_1,c_2} t''_2\,,\ \text{for some } t''_2 \in S_2\,.\end{aligned}$$

As an artifact of positivity in X, the set S_1 is not determined by t_1. The point is that, for instance, if $\exists x(Xx \wedge \chi(x)) \in t_1$, then S_1 may or may not contain a type t' such that $\chi \in t'$ but $Xx \notin t'$, because we do not know about the status of $\exists x(\neg Xx \wedge \chi(x))$.

Unlike S_1, the following superset of S_1 is determined by t_1:

$$S'_1 := \{\, t''_1 \in \mathrm{Tp}^m_X[\tau] \bigm| \exists y \bigwedge t''_1 \in t_1 \,\} \supseteq S_1\,.$$

(Recall that representations of types are finite, so $\bigwedge t''_1$ is in fact a formula.) Hence it suffices to show that

$$\varphi \in t''_1 \oplus^m_{c_1,c_2} t'_2\,,\ \text{for some } t''_1 \in S_1 \quad \text{iff} \quad \varphi \in t''_1 \oplus^m_{c_1,c_2} t'_2\,,\ \text{for some } t''_1 \in S'_1\,.$$

(The corresponding statement for $\mathfrak{T}_2$ then follows by symmetry.)

($\Rightarrow$) is trivial since $S_1 \subseteq S'_1$. For ($\Leftarrow$), assume that t''_1 is a type such that $\exists y \bigwedge t''_1 \in t_1$ and $\varphi \in t''_1 \oplus^m_{c_1,c_2} t'_2$. Let $a \in T_1$ be an element with $(\mathfrak{T}_1, a) \models \bigwedge t''_1$, and set $t'''_1 := \mathrm{tp}^m_X(\mathfrak{T}_1, a)$. Clearly, $t''_1 \subseteq t'''_1$. Hence, monotonicity of $\oplus^m_{c_1,c_2}$ implies that $\varphi \in t'''_1 \oplus^m_{c_1,c_2} t'_2$, as desired.

It remains to show monotonicity of $\oplus^n_{c_1,c_2}$ (as far as the formula $\exists y\varphi$ is concerned). We need to establish that, if $\exists y\varphi \in t_1 \oplus^n_{c_1,c_2} t_2$, then this still holds after increasing t_1 or t_2. This follows from the fact that the sets S'_1 and S'_2 (defined analogously to S'_1) are monotone in t_1 and t_2.

Finally, let us consider a formula of the form $\forall Y\varphi$ with $m := \mathrm{qr}(\varphi) < n$. This time let S_1 be the set of X-positive m-types of expansions of $\mathfrak{T}_1$ by some unary predicate $P \subseteq T_1$ interpreted for Y, and let S_2 be the respective set for $\mathfrak{T}_2$. Using the equality

$$(\mathfrak{T}, P) = (\mathfrak{T}_1, P \cap T_1) +_{c_1,c_2} (\mathfrak{T}_2, P \cap T_2)$$

we obtain, similarly to the case above, that

$$\forall Y\varphi \in t \quad \text{iff} \quad \varphi \in t''_1 \oplus^m_{c_1,c_2} t''_2 \quad \text{for all } t''_1 \in S_1 \text{ and } t''_2 \in S_2\,.$$

Let us call a pair (S'_1, S'_2) *good for* t_1, t_2, if the following conditions hold:

- S'_1 is a set of X-positive m-types of the vocabulary used for expansions of $\mathfrak{T}_1$ by Y and S'_2 is a corresponding set for of $\mathfrak{T}_2$.
- $\forall Y \bigvee_{s_1 \in S'_1} \bigwedge s_1 \in t_1$ and $\forall Y \bigvee_{s_2 \in S'_2} \bigwedge s_2 \in t_2$.
- For all $s_1 \in S'_1$ and $s_2 \in S'_2$ we have $\varphi \in s_1 \oplus^m_{c_1,c_2} s_2$.

If $\forall Y\varphi \in t$, then (S_1, S_2) is good, whence a good pair exists. We claim that the converse also holds, i.e., that the existence of a good pair implies $\forall Y\varphi \in t$. Thus, we obtain a characterisation of whether $\forall Y\varphi \in t$ solely in terms of t_1, t_2, and $\oplus^m_{c_1,c_2}$. Furthermore, being good for t_1, t_2 is clearly monotone in t_1 and t_2.

To prove the claim, suppose that (S'_1, S'_2) is a good pair and let $t''_1 \in S_1$ and $t''_2 \in S_2$ be arbitrary. We need to show that $\varphi \in t''_1 \oplus^m_{c_1,c_2} t''_2$. Fix a predicate P_1 such that $t''_1 =$

$\mathrm{tp}^m_X(\mathfrak{T}_1, P_1)$. By the second condition on good pairs, we have $(\mathfrak{T}_1, P_1) \models \bigvee_{s_1 \in S'_1} \bigwedge s_1$. Hence, there is some $s_1 \in S'_1$ such that $(\mathfrak{T}_1, P_1) \models \bigwedge s_1$. This implies that $s_1 \subseteq t''_1$. Analogously, we obtain some $s_2 \in S'_2$ such that $s_2 \subseteq t''_2$. By the third condition on good pairs, we have $\varphi \in s_1 \oplus^m_{c_1,c_2} s_2$. Therefore, monotonicity of $\oplus^m_{c_1,c_2}$ implies that $\varphi \in t''_1 \oplus^m_{c_1,c_2} t''_2$. $\square$

The previous proof needed to consider different vocabularies. From now on, a single vocabulary nearly suffices. Let τ be a fixed tree vocabulary without any constant symbols. Let X be a unary relation symbol and x a constant symbol such that $x, X \notin \tau$. We will consider fixed points with respect to X and x. The fixed points are evaluated in trees of vocabulary τ. Stages of the fixed-point induction are evaluated in trees of vocabulary $\tau \cup \{X\}$. In order to determine whether a single tree node belongs to some iteration for the fixed point, we consider trees of vocabulary $\tau \cup \{X, x\}$. If x is present in the vocabulary, its interpretation can be thought of as the root of the tree.

Let φ be a $\tau \cup \{X, x\}$-formula positive in X and let n be the quantifier rank of φ.

Corollary 3.3. *Let $y \notin \tau \cup \{X, x\}$ be a new constant symbol. We define a binary operation $\leftthreetimes^n$ on X-positive n-types of $\tau \cup \{X, x\}$-structures by*

$$s \leftthreetimes^n t := \big(s \oplus^n_{x,y} t[y/x]\big) \cap \mathrm{MSO}^n_X[\tau \cup \{X, x\}]\,.$$

The operation $\leftthreetimes^n$ is monotone and satisfies

$$\mathrm{tp}^n_X(\mathfrak{S} \leftthreetimes_x \mathfrak{T}) = \mathrm{tp}^n_X(\mathfrak{S}) \leftthreetimes^n \mathrm{tp}^n_X(\mathfrak{T})\,,$$

for all non-empty tree structures $\mathfrak{S}$ and $\mathfrak{T}$ of vocabulary $\tau \cup \{X, x\}$. $\square$

We extend $\leftthreetimes^n$ by adjoining the X-positive n-type $\triangle$ of the empty tree as a right-neutral element. This does not hurt monotonicity: without loss of generality, assume that $n \geq 1$. Then only $\triangle$ contains $\forall y \bot$ and only this type does not contain $\exists y \top$, so it is incomparable to any other type.

In the first part, which contains the technical heart of the article, we will only consider ternary trees, that is, undirected trees where each node has degree at most 3. We assume that each such tree $\mathfrak{T}$ is implicitly equipped with an edge-colouring using 3 colours $\{1, 2, 3\}$. That means that, for every colour d, each vertex v of $\mathfrak{T}$ has at most one neighbour that is connected to v via an edge of colour d. We call this neighbour "the neighbour of v in *direction* d" and we denote it by v^d. If there is no such neighbour, we set $v^d := \triangle$.

To account for missing neighbours we extend the above definition of $\mathfrak{T}_{vw,x}$ by setting $\mathfrak{T}_{v\triangle,x} := (\mathfrak{T}, v)$ and letting $\mathfrak{T}_{\triangle w,x} := \triangle$. Furthermore, let $\mathfrak{T}_{\{v\}} := (\mathfrak{T} \restriction \{v\}, v)$. With this notation we have

$$(\mathfrak{T}, v) = \mathfrak{T}_{\{v\}} \leftthreetimes_x \mathfrak{T}_{v^1 v,x} \leftthreetimes_x \mathfrak{T}_{v^2 v,x} \leftthreetimes_x \mathfrak{T}_{v^3 v,x}\,,$$

where we assume that the operation $\leftthreetimes_x$ is associative to the left.

We also need a variant of Proposition 3.2 that concerns a decomposition into a possibly infinite number of subtrees. We omit the proof, which is similar to that of Proposition 3.2.

Proposition 3.4. *Let $\mathfrak{T}$ be a $\tau \cup \{X, x\}$-tree and $(v_1, d_1), (v_2, d_2), \ldots$ a sequence of pairwise distinct pairs (v_i, d_i), such that $v_i \in T$ and $v_i^{d_i} = \triangle$. Further, let $\mathfrak{S}_1, \mathfrak{S}_2, \ldots$ and $\mathfrak{S}'_1, \mathfrak{S}'_2, \ldots$ be sequences of $\tau \cup \{X, x\}$-trees such that $\mathrm{tp}^n_X(\mathfrak{S}_i) = \mathrm{tp}^n_X(\mathfrak{S}'_i)$ for all i. Finally, let $\mathfrak{U}$ be the tree obtained from $\mathfrak{T}$ by adding $\mathfrak{S}_i$ as a child of v_i in direction d_i for all i, and define $\mathfrak{U}'$ analogously using $\mathfrak{S}'_i$ instead of $\mathfrak{S}_i$. Then, $\mathrm{tp}^n_X(\mathfrak{U}) = \mathrm{tp}^n_X(\mathfrak{U}')$.* $\square$

4. Tilings

We are now in a position to provide a second, more precise proof outline. Given a tree structure $\mathfrak{T}$ of vocabulary τ, we consider the fixed-point induction of φ. For every stage α and every vertex v of $\mathfrak{T}$ we consider the type $\mathrm{tp}^n_X(\mathfrak{T}, \varphi^\alpha(\mathfrak{T}), v)$. We annotate $\mathfrak{T}$ with all these types. At each vertex v we write down the list of these types for all stages α. These annotations can be used to determine the fixed-point rank of elements of $\mathfrak{T}$. A vertex v enters the fixed point at stage α if the α-th entry of the list is the first one containing a type t with $Xx \in t$.

We can regard the annotation as consisting of several layers, one for each stage of the induction. At a vertex v each change between two consecutive layers is caused by some change at some other vertex in the previous step. In this way we can trace back changes of the types through the various layers.

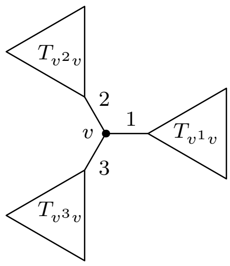
In order to determine whether the fixed-point inductions of the formula are bounded, we construct a *weighted automaton* (see Section 7 below) that recognises (approximations of) such annotations and that computes (an approximation of) the length of the longest path of changes in the annotation.

Actually, the annotations we use do not consist of single types but of tuples of such types, called a *tile.* In this section we consider single layers of such tiles. In the next section we will then introduce annotations consisting of several such layers.

Definition 4.1. A *letter* is a one-element $\tau \cup \{x\}$-tree.

Observe that, for each letter $\mathfrak{L}$, there are exactly two $\tau \cup \{X, x\}$-expansions of $\mathfrak{L}$: one where the element belongs to X and one where it does not. Let us denote their X-positive n-types by $1_{\mathfrak{L}}$ and $0_{\mathfrak{L}}$, respectively. Note that $0_{\mathfrak{L}} \subseteq 1_{\mathfrak{L}}$ and that $Xx \in 1_{\mathfrak{L}} \setminus 0_{\mathfrak{L}}$, for every $\mathfrak{L}$. We omit the index $\mathfrak{L}$ whenever it is irrelevant.

We can decompose a $\tau \cup \{X\}$-tree $\mathfrak{T}$ into its one-element substructures $\mathfrak{T}_{\{v\}}$, i.e., its letters. Each of these letters $\mathfrak{T}_{\{v\}}$ can be labelled with its type and the types of the subtrees $\mathfrak{T}_{v^d v}$.



For convenience, we will not only use the types $t_{\lhd 0}$ and $t_{\lhd d}$ of $\mathfrak{T}_{\{v\}}$ and $\mathfrak{T}_{v^d v}$, $d = 1, 2, 3$, respectively, but also the types $t_{\rhd d}$ of $\mathfrak{T}_{v v^d}$, $d = 1, 2, 3$ and the type $t_{\rhd 4}$ of the whole tree $(\mathfrak{T}, v)$. Our intuition regards the vertex v as a processing unit that receives as its inputs the types $t_{\lhd 0}$, $t_{\lhd 1}$, $t_{\lhd 2}$, $t_{\lhd 3}$ and produces as output the types $t_{\rhd 1}$, $t_{\rhd 2}$, $t_{\rhd 3}$, $t_{\rhd 4}$. The vertex v receives from its neighbours v^d, $d = 1, 2, 3$, the inputs $t_{\lhd d}$ and it passes back to v^d the outputs $t_{\rhd d}$.

Definition 4.2. (a) Let $\mathfrak{L}$ be a letter. An $\mathfrak{L}$-*tile* is an 8-tuple

$$(t_{\lhd 0}, \dots, t_{\lhd 3}, t_{\rhd 1}, \dots, t_{\rhd 4})$$

of X-positive n-types over $\tau \cup \{X, x\}$ where

- $t_{\triangleleft 0} \in \{0_{\mathfrak{L}}, 1_{\mathfrak{L}}\}$,
- $t_{\triangleright 1} = t_{\triangleleft 0} \curlywedge^n t_{\triangleleft 2} \curlywedge^n t_{\triangleleft 3}$,
- $t_{\triangleright 2} = t_{\triangleleft 0} \curlywedge^n t_{\triangleleft 1} \curlywedge^n t_{\triangleleft 3}$,
- $t_{\triangleright 3} = t_{\triangleleft 0} \curlywedge^n t_{\triangleleft 1} \curlywedge^n t_{\triangleleft 2}$, and
- $t_{\triangleright 4} = t_{\triangleleft 0} \curlywedge^n t_{\triangleleft 1} \curlywedge^n t_{\triangleleft 2} \curlywedge^n t_{\triangleleft 3}$.

If we do not want to mention the letter, we refer to an $\mathfrak{L}$-tile simply as a *tile.* When γ is a tile, we denote its components by $\gamma_{\triangleleft 0}$ through $\gamma_{\triangleright 4}$.

(b) Let $\mathfrak{T}$ be a τ-tree. A *$\mathfrak{T}$-tiling* is a mapping c that assigns to each vertex $v \in T$ a $\mathfrak{T}_{\{v\}}$-tile $c(v)$.

(c) Let $\mathfrak{T}$ be a $\tau \cup \{X\}$-tree. The *canonical tiling* $t_{\mathfrak{T}}$ of $\mathfrak{T}$ is the function assigning to a vertex v the tile

$$\begin{aligned} t_{\mathfrak{T}}(v)_{\triangleleft 0} &:= \mathrm{tp}^n_X(\mathfrak{T}_{\{v\}})\,, & t_{\mathfrak{T}}(v)_{\triangleleft d} &:= \mathrm{tp}^n_X(\mathfrak{T}_{v^d v})\,, & &\text{for } 1 \leq d \leq 3\,, \\ t_{\mathfrak{T}}(v)_{\triangleright 4} &:= \mathrm{tp}^n_X(\mathfrak{T}, v)\,, & t_{\mathfrak{T}}(v)_{\triangleright d} &:= \mathrm{tp}^n_X(\mathfrak{T}_{v v^d})\,, & &\text{for } 1 \leq d \leq 3\,. \end{aligned}$$

Intuitively the $\triangleleft d$-component of a tile contains information *incoming* from direction d, whereas the $\triangleright d$-component contains the information passed on in that direction. Similarly, the $\triangleright 4$-component contains information passed on to the next stage. The $\triangleleft 0$-component is special, since it contains local information about the current vertex.

Note that the canonical tiling is indeed a tiling.

Lemma 4.3. *Let $\mathfrak{T}$ be a $\tau \cup \{X\}$-tree and $\mathfrak{T}_0$ its τ-reduct. Then $t_{\mathfrak{T}}$ is a $\mathfrak{T}_0$-tiling.*

Proof. Let $v \in T$. Since $\mathfrak{T}_{\{v\}}$ is an expansion of $\mathfrak{L} := (\mathfrak{T}_0)_{\{v\}}$, its type $t_{\mathfrak{T}}(v)_{\triangleleft 0}$ must be one of $0_{\mathfrak{L}}$ and $1_{\mathfrak{L}}$.

For the equalities concerning $t_{\mathfrak{T}}(v)_{\triangleright d}$ with $1 \leq d \leq 3$, we may by symmetry assume that $d = 3$. Then

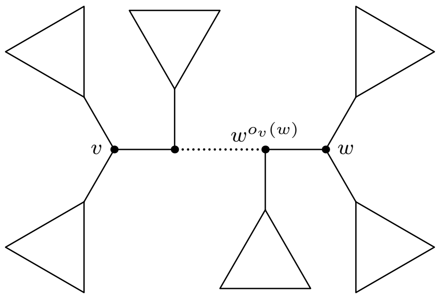
$$\begin{aligned} t_{\mathfrak{T}}(v)_{\triangleright 3} &= \mathrm{tp}^n_X(\mathfrak{T}_{v v^3}) \\ &= \mathrm{tp}^n_X(\mathfrak{T}_{\{v\}} \curlywedge_x \mathfrak{T}_{v^1 v} \curlywedge_x \mathfrak{T}_{v^2 v}) \\ &= \mathrm{tp}^n_X(\mathfrak{T}_{\{v\}}) \curlywedge^n \mathrm{tp}^n_X(\mathfrak{T}_{v^1 v}) \curlywedge^n \mathrm{tp}^n_X(\mathfrak{T}_{v^2 v}) \\ &= t_{\mathfrak{T}}(v)_{\triangleleft 0} \curlywedge^n t_{\mathfrak{T}}(v)_{\triangleleft 1} \curlywedge^n t_{\mathfrak{T}}(v)_{\triangleleft 2}\,, \end{aligned}$$

as desired. The equality for $\triangleright 4$ is obtained similarly. $\square$

Not every tiling stems from an actual tree. In the next definition we collect some simple consistency properties a tiling should satisfy. Note that these properties can be checked by an automaton.

Definition 4.4. Let $\mathfrak{T}$ be a τ-tree and $v \in T$ a vertex.

(a) The *orientation of $\mathfrak{T}$ towards v* is the mapping $o_v : T \to \{1, \dots, 4\}$ such that $o_v(v) = 4$ and, for vertices $w \in T \setminus \{v\}$, we define $1 \leq o_v(w) \leq 3$ such that the neighbour $w^{o_v(w)}$ is closer to v than w.



(b) A $\mathfrak{T}$-tiling c is *locally consistent towards* v if, for all $w \in T$ and all directions $1 \leq d \leq 3$ with $d \neq o_v(w)$, we have

$$c(w)_{\lhd d} = \begin{cases} c(w^d)_{\rhd d} & \text{if } w^d \neq \triangle\,, \\ \triangle & \text{otherwise}\,. \end{cases}$$

(c) A $\mathfrak{T}$-tiling c is *globally consistent towards* v if, for all vertices $w \in T$ and all directions $1 \leq d \leq 3$ with $d \neq o_v(w)$, we have

$$c(w)_{\lhd d} = \mathrm{tp}^n_X((\mathfrak{T}, P)_{w^d w})\,,$$

where $(\mathfrak{T}, P)$ is the expansion of $\mathfrak{T}$ by the set $P := \{\, v \in T \mid c(v)_{\lhd 0} = 1 \,\}$ interpreted for X.

Of course, canonical tilings are globally consistent.

Lemma 4.5. *Let $\mathfrak{T}$ be a τ-tree and $P \subseteq T$. The $\mathfrak{T}$-tiling $t_{(\mathfrak{T},P)}$ is globally consistent towards each vertex $v \in T$.*

Proof. We have already seen in Lemma 4.3 that $t_{(\mathfrak{T},P)}$ is a $\mathfrak{T}$-tiling. Let $v \in T$. For global consistency, note that

$$\begin{aligned} P &= \big\{\, v \in T \bigm| (\mathfrak{T}, P, v) \models Xx \,\big\} \\ &= \big\{\, v \in T \bigm| \mathrm{tp}^n_X((\mathfrak{T}, P)_{\{v\}}) = 1 \,\big\} \\ &= \big\{\, v \in T \bigm| t_{\mathfrak{T},P}(v)_{\lhd 0} = 1 \,\big\}\,, \end{aligned}$$

as desired. □

Finally, let us show that global consistency implies local consistency.

Lemma 4.6. *Let $\mathfrak{T}$ be a τ-tree and $v \in T$. Every $\mathfrak{T}$-tiling that is globally consistent towards v is locally consistent towards v.*

Proof. Let c be a $\mathfrak{T}$-tiling globally consistent towards v and let $\mathfrak{T}'$ be the $\tau \cup \{X\}$-expansion of $\mathfrak{T}$ by the set $P := \{\, v \in T \mid c(v)_{\lhd 0} = 1 \,\}$. Let $w \in T$ and $d \neq o_v(w)$ be given. Without loss of generality, we may assume that $d = 3$. If $w^3 = \triangle$, then $c(w)_{\lhd 3}$ is the type of $\mathfrak{T}'_{\triangle v} = \triangle$. Otherwise, let $u := w^3 \neq \triangle$. Since c is a $\mathfrak{T}$-tiling, $c(u)_{\lhd 0}$ is either $0_{\mathfrak{T}_{\{u\}}}$ or $1_{\mathfrak{T}_{\{u\}}}$. By definition of $\mathfrak{T}'$ it follows that $c(u)_{\lhd 0} = \mathrm{tp}^n_X(\mathfrak{T}'_{\{u\}})$. Consequently,

$$\begin{aligned} c(w)_{\lhd 3} = \mathrm{tp}^n_X(\mathfrak{T}'_{uw}) &= \mathrm{tp}^n_X(\mathfrak{T}'_{\{u\}}) \curlywedge^n \mathrm{tp}^n_X(\mathfrak{T}'_{u^1 u}) \curlywedge^n \mathrm{tp}^n_X(\mathfrak{T}'_{u^2 u}) \\ &= c(u)_{\lhd 0} \curlywedge^n c(u)_{\lhd 1} \curlywedge^n c(u)_{\lhd 2} \\ &= c(u)_{\rhd 3}\,. \end{aligned}$$

□

5. Annotations

Ideally we would like to annotate a given tree with one tiling for each stage of the fixed-point induction. Since this is an infinite amount of data we have to opt for something less: at each vertex of the tree we do not store the full sequence of tiles for each stage, but only a shortened sequence obtained by removing all duplicates. This is a finite amount of information we can label the tree with. The drawback of this method is that, by removing duplicates, we lose synchronisation between the sequences from adjacent vertices. Here are the formal definitions.

For a τ-tree $\mathfrak{T}$ and an ordinal α, let $\mathfrak{T}^\alpha := (\mathfrak{T}, \varphi^\alpha(\mathfrak{T}))$ be the $\tau \cup \{X\}$-expansion of $\mathfrak{T}$ by the αth stage of the fixed-point induction. Similarly, we set $\mathfrak{T}^\alpha_{vw,x} := (\mathfrak{T}^\alpha)_{vw,x}$ and $\mathfrak{T}^\alpha_{\{v\}} := (\mathfrak{T}^\alpha)_{\{v\}}$.

We extend the order $\subseteq$ on X-positive n-types to tiles by requiring that $\subseteq$ holds component-wise.

Definition 5.1. (a) Let $\mathfrak{L}$ be a letter. An $\mathfrak{L}$-*history* is a strictly increasing sequence $h = \big(h^0 \subsetneq \ldots \subsetneq h^m\big)$ of $\mathfrak{L}$-tiles such that

(1) $h^0_{\lhd 0} = 0_{\mathfrak{L}}$ and
(2) $h^{i+1}_{\lhd 0} = 1_{\mathfrak{L}}$ iff $\varphi \in h^i_{\rhd 4}$, for $0 \leq i < m$.

The number m is the *length* of the history, denoted $|h|$.

(b) Let $\mathfrak{T}$ be a τ-tree and $v \in T$ a vertex. *The history* of $\mathfrak{T}$ at v is the sequence $h_{\mathfrak{T}}(v)$ of tiles $t_{\mathfrak{T}^\alpha}(v)$, for all ordinals α, with duplicates removed.

Example 5.2. For simplicity, we give an example of a fixed-point induction on a path, instead of a tree, i.e., a tree where no vertex has a neighbour in direction 3. We consider the fixed-point of the formula $\varphi(X, x)$ stating that

$$x^1 = \triangle \quad \text{or} \quad x^2 = \triangle \quad \text{or} \quad T_{x^1x} \subseteq X \quad \text{or} \quad T_{x^2x} \subseteq X\,.$$

Figure 1 shows the histories of the first 4 elements of a finite path of length at least 9. All further elements, except for the last two, have the same history as the third and fourth elements. Here, we assume that the edges are alternatingly labelled by 1 and 2 and the tiles are drawn in the format

$\rhd 4$	$\rhd 1$	$\rhd 2$	$\rhd 3$
$\lhd 0$	$\lhd 1$	$\lhd 2$	$\lhd 3$

where

- $\triangle$ denotes the type of the empty tree,
- φ denotes any type containing φ,
- $\times$ denotes any type not containing φ,
- $\forall$ denotes any type containing the formula $\forall y Xy$,
- $\exists$ denotes any type containing $\exists y Xy$, but not $\forall y Xy$, and
- $-$ denotes any type not containing the formula $\exists y Xy$.

Of course, the history of $\mathfrak{T}$ at v is indeed a history.

Lemma 5.3. *Let $\mathfrak{T}$ be a τ-tree and $v \in T$ a vertex. Then $h_{\mathfrak{T}}(v)$ is a $\mathfrak{T}_{\{v\}}$-history.*

Proof. Let $h := h_{\mathfrak{T}}(v)$. We have already seen in Lemma 4.3 that each h^i is a $\mathfrak{T}_{\{v\}}$-tile. The sequence is increasing, because we are considering positive (hence monotone) types and the sequence $\varphi^\alpha(\mathfrak{T})$ is increasing. It is strictly increasing because we have removed

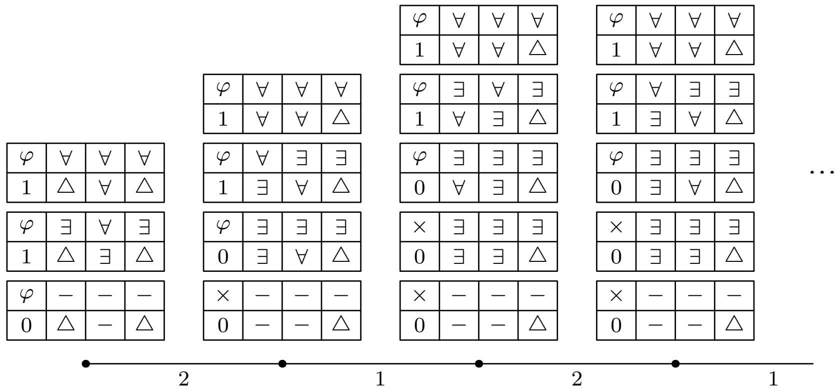


Figure 1: Annotation for $\varphi(X, x)$

duplicates. For ordinals α, let $k(\alpha)$ be the index at which the αth stage appears in h, i.e., $h^{k(\alpha)} = t_{\mathfrak{T}^\alpha}(v)$. As the sequence is increasing, so is k.

Since $\varphi^0(\mathfrak{T}) = \emptyset$, we have

$$h^0_{\lhd 0} = h^{k(0)}_{\lhd 0} = t_{\mathfrak{T}^0}(v)_{\lhd 0} = \mathrm{tp}^n_X(\mathfrak{T}^0_{\{v\}}) = 0_{\mathfrak{L}}\,.$$

For $0 \leq i < |h|$, let α be the minimal ordinal with $k(\alpha) = i + 1$. Then

$$h^{i+1}_{\lhd 0} = 1 \quad \text{iff} \quad \mathrm{tp}^n_X(\mathfrak{T}^\alpha_{\{v\}}) = 1 \quad \text{iff} \quad v \in \varphi^\alpha(\mathfrak{T})\,.$$

Since elements enter the fixed point only at successor stages, we have

$$\begin{aligned} v \in \varphi^\alpha(\mathfrak{T}) \quad &\text{iff} \quad v \in \varphi^{\beta+1}(\mathfrak{T}) \quad \text{for some } \beta < \alpha\,, \\ &\text{iff} \quad (\mathfrak{T}^\beta, v) \models \varphi \\ &\text{iff} \quad \varphi \in \mathrm{tp}^n_X(\mathfrak{T}^\beta, v) \subseteq h^i_{\rhd 4}\,. \end{aligned}$$

□

We would like to annotate each vertex v of a tree $\mathfrak{T}$ by the sequence $(t_{\mathfrak{T}^\alpha}(v))_\alpha$. To obtain a finite object, we have to remove duplicates and, therefore, we work with the history $h_{\mathfrak{T}}(v)$ instead. For each α, we would like to have an automaton that can recover the tiling $t_{\mathfrak{T}^\alpha}$ from $h_{\mathfrak{T}}$. In general, this is not possible.

For instance, in Example 5.2 the 'real' tilings $t_{\mathfrak{T}^\alpha}(v)$ for a path $\mathfrak{T}$ of even length are words of the form $ux^n y^n v$ where $y^n v$ is the 'mirror image' of ux^n. This language is not regular.

Hence, we use an approximation. For each vertex v, each index i of $h_{\mathfrak{T}}(v)$, and each direction d, we record the index j of $h_{\mathfrak{T}}(v^d)$ such that $h_{\mathfrak{T}}(v^d)^j$ and $h_{\mathfrak{T}}(v)^i$ belong to the same ordinal α. Of course, given i, there are several choices of α and, hence, of j, so we lose information. It will turn out that these two pieces of data, the function h and the function $(v, i, d) \mapsto j$, are sufficient for our purposes.

Definition 5.4. (a) An *annotated tree* is a tuple $(\mathfrak{T}, h, s)$, where

(1) $\mathfrak{T}$ is a τ-tree,
(2) h is a mapping that assigns to each vertex $v \in T$ a $\mathfrak{T}_{\{v\}}$-history $h(v)$, and
(3) s is a mapping assigning a natural number $s(v, i, d)$ to each vertex $v \in T$, each index $0 \leq i \leq |h(v)|$, and every direction $1 \leq d \leq 3$ with $v^d \neq \triangle$.

We call h the *history map* and s the *synchronisation* of the annotated tree.

(b) Let $(\mathfrak{T}, h, s)$ be an annotated tree. For $v \in T$ and $0 \leq i \leq |h(v)|$, the *section* at v, i is the tiling c defined inductively as follows:

(1) $c(v) := h(v)^i$.
(2) For $w \in T \setminus \{v\}$, let $u := w^{o_v(w)}$. We assume by induction that $c(u)$ is already defined. Let j be the index such that $c(u) = h(u)^j$. Then we set $c(w) := h(w)^{s(u,j,o_v(w))}$.

Of course, not every annotated tree $(\mathfrak{T}, h, s)$ encodes the 'real' fixed-point induction. The next definition collects some necessary conditions.

Definition 5.5. Let $\mathcal{A} = (\mathfrak{T}, h, s)$ be an annotated tree.

(a) $\mathcal{A}$ is *locally consistent* if, for all vertices $v \in T$, indices $0 \leq i \leq |h(v)|$, and directions $1 \leq d \leq 3$ the following conditions are satisfied:

(1) If $v^d = \triangle$, then $h(v)^i_{\lhd d} = \triangle$.
(2) Otherwise, $s(v, i, d) \leq \left|h\left(v^d\right)\right|$ and $h(v)^i_{\lhd d} = h\left(v^d\right)^{s(v,i,d)}_{\rhd d}$.

(b) $\mathcal{A}$ is *globally consistent* if it is locally consistent and if, for all v, i as above, the section at v, i is globally consistent towards v.

Lemma 5.6. *Let $(\mathfrak{T}, h, s)$ be a locally consistent annotated tree. Every section c at some v, i is locally consistent towards v.*

Proof. Let $v, w \in T$ be distinct vertices, $d \neq o_v(w)$, and let i be the index such that $c(w) = h(w)^i$. Then we have $w^d = \triangle$ and $c(w)_{\lhd d} = h(w)^i_{\lhd d} = \triangle$, or

$$c(w)_{\lhd d} = h(w)^i_{\lhd d} = h(w^d)^{s(w,i,d)}_{\rhd d} = c(w^d)_{\rhd d}\,.$$

□

We have not yet defined the 'real annotation' of a tree. In fact, due to the choices involved in defining the synchronisation there are several possible 'real' annotations. We obtain them by fixing an ordinal β and selecting that synchronisation that selects from among all possible choices the stage that is closest to β.

Definition 5.7. Let $\mathfrak{T}$ be a τ-tree and $\beta < \omega$. We denote by $\mathcal{A}_\beta(\mathfrak{T})$ the annotated tree $(\mathfrak{T}, h_{\mathfrak{T}}, s)$ where the synchronisation s is defined as follows. For $v \in T$, $0 \leq i \leq |h_{\mathfrak{T}}(v)|$, and $1 \leq d \leq 3$ with $w := v^d \neq \triangle$, we define $s(v, i, d)$ such that

$$h_{\mathfrak{T}}(w)^{s(v,i,d)} = t_{\mathfrak{T}^\alpha}(w)\,,$$

where the ordinal α is chosen as follows:

(1) if $h_{\mathfrak{T}}(v)^i = t_{\mathfrak{T}^\beta}(v)$, then $\alpha = \beta$,
(2) if $h_{\mathfrak{T}}(v)^i \subsetneq t_{\mathfrak{T}^\beta}(v)$, then $\alpha \leq \beta$ is maximal such that $t_{\mathfrak{T}^\alpha}(w)_{\rhd d} = h_{\mathfrak{T}}(v)^i_{\lhd d}$, and
(3) if $h_{\mathfrak{T}}(v)^i \supsetneq t_{\mathfrak{T}^\beta}(v)$, then $\alpha \geq \beta$ is minimal such that $t_{\mathfrak{T}^\alpha}(w)_{\rhd d} = h_{\mathfrak{T}}(v)^i_{\lhd d}$.

We start with a technical lemma containing a monotonicity property for the sections of an annotation.

Lemma 5.8. *Let $\mathfrak{T}$ be a tree with vertices $v, w \in T$, let c be the section of $\mathcal{A}_\beta(\mathfrak{T})$ at v, i, and set $d := o_v(w)$.*

(a) *If $c(w) = t_{\mathfrak{T}^\beta}(w)$, then $c(u) = t_{\mathfrak{T}^\beta}(u)$, for all $u \in T_{ww^d}$.*
(b) *Let $\alpha < \beta$. If $t_{\mathfrak{T}^\alpha}(w) \subseteq c(w)$, then $t_{\mathfrak{T}^\alpha}(u) \subseteq c(u)$, for all $u \in T_{ww^d}$.*

(Here we set $T_{ww^4} := T$.)

Proof. We only prove the claims for $w \neq v$. The argument for $w = v$ is similar. We prove both claims by induction on the distance between $u \in T_{ww^d}$ and v. The claims are immediate for $u = w$.

For the inductive step assume that the claims hold for u and let u' be a neighbour of u which is further away from v than u, so that $u = (u')^{o_v(u')}$. It follows that $u' = u^{d'}$ for some $d' \neq o_v(u)$. Let i be the index such that $c(u) = h_{\mathfrak{T}}(u)^i$. By definition of c and s, respectively, we have

$$c(u') = h_{\mathfrak{T}}(u')^{s(u,i,d')} = t_{\mathfrak{T}^{\alpha'}}(u')\,,$$

for some ordinal α'.

For (a), using the inductive hypothesis, we have $h_{\mathfrak{T}}(u)^i = c(u) = t_{\mathfrak{T}^\beta}(u)$, which implies that $\alpha' = \beta$. Hence, $c(u') = t_{\mathfrak{T}^\beta}(u')$.

Similarly, for (b), we have $h_{\mathfrak{T}}(u)^i = c(u) \supseteq t_{\mathfrak{T}^\alpha}(u)$, which implies that $\alpha' \geq \alpha$. Hence, $c(u') \supseteq t_{\mathfrak{T}^\alpha}(u')$. □

Let us also show that $\mathcal{A}_\beta(\mathfrak{T})$ is always globally consistent.

Lemma 5.9. *For all τ-trees $\mathfrak{T}$ and all $\beta < \omega$, $\mathcal{A}_\beta(\mathfrak{T})$ is a globally consistent annotated tree.*

Proof. We have seen in Lemma 5.3 that $h_{\mathfrak{T}}(v)$ is a $\mathfrak{T}_{\{v\}}$-history. Hence, $\mathcal{A}_\beta(\mathfrak{T})$ is an annotated tree. For local consistency, fix v, i, d and let α be the ordinal from the definition of s at v in $\mathcal{A}_\beta(\mathfrak{T})$ (cf. Definition 5.7). Then

$$h_{\mathfrak{T}}(v^d)^{s(v,i,d)}_{\triangleright d} = t_{\mathfrak{T}^\alpha}(v^d)_{\triangleright d} = h_{\mathfrak{T}}(v)^i_{\triangleleft d}\,,$$

as desired.

It remains to prove global consistency. Fix a vertex $v \in T$ and an index $0 \leq i \leq |h_{\mathfrak{T}}(v)|$, and let c be the section at v, i. For $w \in T$, let $\alpha(w)$ be the ordinal closest to β such that $c(w) = t_{\mathfrak{T}^{\alpha(w)}}(w)$. Let $\mathfrak{T}'$ be the expansion of $\mathfrak{T}$ by the set $P := \{\, w \in T \mid c(w)_{\triangleleft 0} = 1 \,\}$. We need to show that $c(w)_{\triangleleft d} = \mathrm{tp}^n_X(\mathfrak{T}'_{w^d w})$, for all $w \in T$ and $d \neq o_v(w)$. (Here, $\mathfrak{T}'_{v^4 v} := \mathfrak{T}'$.) By local consistency of c (which holds by Lemma 5.6), it is sufficient to show that $c(w)_{\triangleright d} = \mathrm{tp}^n_X(\mathfrak{T}'_{ww^d})$, for all $w \in T$ and $d := o_v(w)$. We do this by induction on the distance between $\alpha(w)$ and β.

First, suppose that $\alpha(w) = \beta$. Then Lemma 5.8 (a) implies that $c(u) = t_{\mathfrak{T}^\beta}(u)$, for all $u \in T_{ww^d}$. Consequently $\mathfrak{T}'_{ww^d} = \mathfrak{T}^\beta_{ww^d}$, and hence $\mathrm{tp}^n_X(\mathfrak{T}'_{ww^d}) = t_{\mathfrak{T}^\beta}(w)_{\triangleright d} = c(w)_{\triangleright d}$, as desired.

It remains to consider the case that $\alpha(w) \neq \beta$. By symmetry, we may assume that $\alpha(w) > \beta$. Let $\mathfrak{S}$ be the maximal subtree of $\mathfrak{T}'_{ww^d}$ that contains the vertex w and such that $\alpha(u) = \alpha(w)$ for all $u \in S$. Let $(x_1, d_1), (x_2, d_2), \dots$ be the finite or infinite list of all pairs (x, d) such that $x \in S$ and $x^d \in T \setminus S$. Let $y_k := x_k^{d_k}$ be the missing neighbour. Note that, by definition of s, $\alpha(y_k)$ is the minimal ordinal α such that $\mathrm{tp}^n_X(\mathfrak{T}^\alpha_{y_k x_k}) = \mathrm{tp}^n_X(\mathfrak{T}^{\alpha(x_k)}_{y_k x_k})$. Hence, $\alpha(y_k) \leq \alpha(x_k)$ and it follows that $\alpha(x_k) = \alpha(w)$ and $\alpha(y_k) < \alpha(w)$. By local consistency and the inductive hypothesis, we have

$$c(x_k)_{\triangleleft d_k} = c(y_k)_{\triangleright d_k} = \mathrm{tp}^n_X(\mathfrak{T}'_{y_k x_k})\,,$$

while, by definition of S, we have

$$c(x_k)_{\triangleleft d_k} = t_{\mathfrak{T}^{\alpha(w)}}(x_k)_{\triangleleft d_k} = \mathrm{tp}^n_X(\mathfrak{T}^{\alpha(w)}_{y_k x_k})\,.$$

It follows that $\mathrm{tp}^n_X(\mathfrak{T}'_{y_k x_k}) = \mathrm{tp}^n_X(\mathfrak{T}^{\alpha(w)}_{y_k x_k})$ for each index k. As the subtrees of $\mathfrak{T}'_{ww^d}$ and $\mathfrak{T}^{\alpha(w)}_{ww^d}$ induced by S agree, we can use Proposition 3.4 to deduce that

$$c(w)_{\triangleright d} = \mathrm{tp}^n_X(\mathfrak{T}^{\alpha(w)}_{ww^d}) = \mathrm{tp}^n_X(\mathfrak{T}'_{ww^d})\,.$$

□

6. Ranks

It remains to compute the length of the fixed-point iteration from a given annotated tree. The goal essentially is to obtain an estimate for the stage of a designated element of the fixed point; this estimate is extracted from an annotation in terms of the weight of an accepting run of a weighted automaton which checks consistency of the annotation. The appropriate kind of weighted automata for this purpose will be presented in the next section.

Definition 6.1. (a) Let $(\mathfrak{T}, h, s)$ be an annotated tree and $v \in T$ a node. We say that there is a *jump* at v if there is some index i such that $h(v)^i_{\triangleleft 0} = 0$ and $h(v)^{i+1}_{\triangleleft 0} = 1$. Observe that this value of i is uniquely determined. We call the jump a *base jump* if $i = 0$.

(b) Suppose that there is a jump at v that is not a base jump. We say that this jump *depends* on another jump at a node w if $c(w)_{\triangleleft 0} = 0$ where c is the section at v and $i-1$. The *rank* of a jump is the minimal number of jumps on any dependency chain from this jump to some base jump.

(c) An annotated tree $(\mathfrak{T}, h, s)$ is *jump-consistent,* if the set of vertices with a jump equals $\varphi^\infty(\mathfrak{T})$.

The notion of *dependency* in (b) may warrant some comment, because the terminology could easily be misunderstood. What the criterion is meant to capture is not that there must be a (causal or temporal) dependence of the appearance of v in the fixed point on the (prior) appearance of w; rather, it says that such a dependence cannot be ruled out. At least any w that v does *not* depend on in the sense of the definition can have had no influence on the appearance of v. In this sense our dependency relation provides a generous upper bound on any intuitive 'real' dependency: it may be useful to think of w as a *potential* trigger for v.

Let us compare the rank of a jump with the stage of the corresponding vertex in the fixed point (the stage at which the vertex enters the fixed point). First, we show that in every annotation the latter bounds the former.

Lemma 6.2. *Let $\mathcal{A} = (\mathfrak{T}, h, s)$ be a globally consistent and jump-consistent annotated tree, let v be a vertex with a jump in $\mathcal{A}$, and $\alpha < \omega$. If $v \in \varphi^\alpha(\mathfrak{T})$, then the rank of v is at most α.*

Proof. We proceed by induction on α. For $\alpha = 0$ there is nothing to do since $\varphi^0(\mathfrak{T}) = \emptyset$. Hence, we may assume that $\alpha > 0$ and that the claim already holds for smaller ranks. Let i be the index such that $h(v)^i_{\triangleleft 0} = 0$ and $h(v)^{i+1}_{\triangleleft 0} = 1$.

If $i = 0$, then there is a base jump at v and its rank is $1 \leq \alpha$.

For $i > 0$, let c be the section at $v, i-1$ and let $P := \{\, w \in T \mid c(w)_{\triangleleft 0} = 1 \,\}$. From $h(v)^i_{\triangleleft 0} = 0$ we conclude that $\varphi \notin c(v)_{\triangleright 4}$. As c is globally consistent, it follows that $\varphi \notin \mathrm{tp}^n_X(\mathfrak{T}, P, v)$. On the other hand, $\varphi \in \mathrm{tp}^n_X(\mathfrak{T}, \varphi^{\alpha-1}(\mathfrak{T}), v)$. By monotonicity, there must be some vertex $w \in \varphi^{\alpha-1}(\mathfrak{T}) \smallsetminus P$, which, by jump-consistency, has a jump. As $w \notin P$, we have $c(w)_{\lhd 0} = 0$. Consequently, v depends on w. By inductive hypothesis, w has rank at most $\alpha - 1$. Therefore, v has rank at most α. □

Some form of converse is true for annotations of the form $\mathcal{A}_\beta(\mathfrak{T})$.

Lemma 6.3. *Let $\mathfrak{T}$ be a τ-tree, $v \in \varphi^\infty(\mathfrak{T})$, and $\alpha < \beta < \omega$. If the rank of v in $\mathcal{A}_\beta(\mathfrak{T})$ is at most α, then $v \in \varphi^\alpha(\mathfrak{T})$.*

Proof. We proceed by induction on α. As all ranks are positive, $\alpha > 0$. Let i be the index such that $h_\mathfrak{T}(v)^i_{\lhd 0} = 0$ and $h_\mathfrak{T}(v)^{i+1}_{\lhd 0} = 1$. If $i = 0$, then $\varphi \in h_\mathfrak{T}(v)^0_{\rhd 4} = t_{\mathfrak{T}^0}(v)$. Therefore, $v \in \varphi^1(\mathfrak{T}) \subseteq \varphi^\alpha(\mathfrak{T})$ and we are done. Hence we may assume that $i > 0$, i.e., the jump at v is not a base jump.

Let c be the section at $v, i-1$. As the jump at v is not a base jump and its rank is finite, there is some vertex w with a jump rank at most $\alpha - 1$ such that v depends on w. By the inductive hypothesis, we have $w \in \varphi^{\alpha-1}(\mathfrak{T})$. Consequently, $t_{\mathfrak{T}^{\alpha-1}}(w)_{\lhd 0} = 1$. On the other hand, we have $c(w)_{\lhd 0} = 0$ by choice of w. Therefore, $c(w) \subsetneq t_{\mathfrak{T}^{\alpha-1}}(w)$. By Lemma 5.8, this implies that

$$h_\mathfrak{T}(v)^{i-1} = c(v) \subsetneq t_{\mathfrak{T}^{\alpha-1}}(v)\,.$$

It follows that $\alpha' < \alpha - 1 < \beta < \omega$ for any α' such that $h_\mathfrak{T}(v)^{i-1} = t_{\mathfrak{T}^{\alpha'}}(v)$. Therefore, there exists a maximal such ordinal α' and, moreover, $\alpha' + 2 \leq \alpha$.

As i is maximal such that $h_\mathfrak{T}(v)^i_{\lhd 0} = 0$, it follows that $i - 1$ is maximal such that $\varphi \notin h_\mathfrak{T}(v)^{i-1}_{\rhd 4}$. Accordingly, α' is maximal such that $\varphi \notin t_{\mathfrak{T}^{\alpha'}}(v)_{\rhd 4}$. Thus, $\varphi \in t_{\mathfrak{T}^{\alpha'+1}}(v)_{\rhd 4}$ and $t_{\mathfrak{T}^{\alpha'+2}}(v)_{\lhd 0} = 1$. It follows that $v \in \varphi^{\alpha'+2}(\mathfrak{T}) \subseteq \varphi^\alpha(\mathfrak{T})$. □

It follows that boundedness of the fixed-point iteration is equivalent to the existence of a finite bound on the ranks of all annotations.

Definition 6.4. A *proposal* is a tuple $(\mathfrak{T}, h, s, v)$, where $(\mathfrak{T}, h, s)$ is a globally consistent and jump consistent annotated tree and $v \in \varphi^\infty(\mathfrak{T})$. The *rank* of such a proposal is the rank of the jump at v in $(\mathfrak{T}, h, s)$.

Proposition 6.5. *A formula φ is bounded over the class of all ternary trees if, and only if, there is some number $N < \omega$ such that the rank of each proposal is at most N.*

Proof. ($\Leftarrow$) Suppose there exists a bound $N < \omega$ on the ranks of proposals. Let $\mathfrak{T}$ be some ternary tree, and $v \in \varphi^\infty(\mathfrak{T})$. By Lemma 5.9, $(\mathcal{A}_{N+1}(\mathfrak{T}), v)$ is a proposal. By choice of N, the rank of v is at most N. Hence, Lemma 6.3 implies that $v \in \varphi^N(\mathfrak{T})$. As v was arbitrary, it follows that $\varphi^N(\mathfrak{T}) = \varphi^\infty(\mathfrak{T})$.

($\Rightarrow$) Suppose that φ is bounded by some number $N < \omega$. Let $(\mathfrak{T}, h, s, v)$ be an arbitrary proposal. Then $v \in \varphi^\infty(\mathfrak{T}) = \varphi^N(\mathfrak{T})$ and Lemma 6.2 implies that the rank of the proposal is at most N. □

7. Weighted automata

In order to decide the boundedness problem for MSO we reduce it to the so-called *limitedness problem* for a certain kind of weighted automaton. These automata have Σ-*labelled directed trees* as inputs. Such a tree is a triple (T, E, λ) where $\lambda : T \to \Sigma$ is a labelling of T and (T, E) is a directed tree (meaning that $E \cap E^{-1} = \emptyset$, $\big(T, E \cup E^{-1}\big)$ is a tree structure, and there is some $r \in T$ called the *root* of the tree such that $r\ E^*\ t$ for all $t \in T$).

Definition 7.1. (a) A *weighted parity automaton* $\mathcal{A} = (Q, \Sigma, \Delta, I, \Omega, w)$ consists of a finite *state space* Q, a finite *input alphabet* Σ, a set $I \subseteq Q$ of *initial states,* a finite *transition relation* $\Delta \subseteq \Sigma \times \omega^Q \times Q$, a *priority function* $\Omega : Q \to \omega$, and a *weight function* $w : \Delta \to \omega$.

A weighted parity automaton $\mathcal{A}$ takes as input Σ-labelled directed trees (T, E, λ). Let $\pi_3 : \Sigma \times \omega^Q \times Q \to Q$ be the projection to the third component. A *run* of $\mathcal{A}$ on this tree is a mapping $\varrho : T \to \Delta$ satisfying, for all vertices $v \in T$, the following condition:

$$\varrho(v) = (c, f, q) \quad \text{implies} \quad \begin{array}{l} c = \lambda(v) \text{ and } f \text{ is the function mapping } p \in Q \text{ to} \\ \text{the number of children } u \text{ of } v \text{ with } \pi_3(\varrho(u)) = p\,. \end{array}$$

A run ϱ is *accepting,* if

- $\pi_3(\varrho(r)) \in I$ for the root r of (T, E, λ) and
- for every branch β of (T, E), the limit

$$\liminf_{v \in \beta} \Omega(\pi_3(\varrho(v))) \quad \text{is even.}$$

The *language* $L(\mathcal{A})$ recognised by $\mathcal{A}$ is the set of all Σ-labelled directed trees (T, E, λ) on which there is an accepting run of $\mathcal{A}$.

For a run ϱ on some tree (T, E, λ) and a branch β of the tree, we set

$$w_{\mathcal{A}}(\varrho, \beta) := \sum_{v \in \beta} w(\varrho(v)) \quad \text{and} \quad w_{\mathcal{A}}(\varrho) := \sup_{\beta} w_{\mathcal{A}}(\varrho, \beta)\,,$$

with values in $\omega \cup \{\infty\}$.

The associated *cost function* $w_{\mathcal{A}}$ maps $(T, E, \lambda) \in L(\mathcal{A})$ to the minimum of $w_{\mathcal{A}}(\varrho)$ taken over all accepting runs ϱ on (T, E, λ). If $(T, E, \lambda) \notin L(\mathcal{A})$, $w_{\mathcal{A}}$ returns ∞.

(b) We say that the automaton $\mathcal{A}$ is *limited,* if there is some bound $N < \omega$ such that $w_{\mathcal{A}}(T, E, \lambda) \leq N$ for all $(T, E, \lambda) \in L(\mathcal{A})$. We say that $\mathcal{A}$ is *limited in the finite,* if there is a bound $N < \omega$ such that $w_{\mathcal{A}}(T, E, \lambda) \leq N$ for all *finite* $(T, E, \lambda) \in L(\mathcal{A})$.

Note that, if we only consider finite trees as input, we can omit the priority function Ω from the automaton. Weighted automata as defined above are a special case of so-called *cost tree automata* introduced in [7]. In that paper it is shown than the limitedness problem for cost tree automata over finite trees is decidable. Hence, the following is a direct consequence of [7].

Theorem 7.2 (Colcombet and Löding)**.** *It is decidable whether a weighted parity automaton $\mathcal{A}$ is limited in the finite.* □

Colcombet and Löding have also announced a decidability result for the general limitedness problem, but this result has not been published yet.

Theorem 7.3 (Colcombet and Löding)**.** *It is decidable whether a weighted parity automaton $\mathcal{A}$ is limited.* □

Although the proof is still not published, its key arguments appear in [28, 6]. The following sketch of how they fit together was communicated to the authors by Colcombet and Löding.

A *cost function* $f : \mathcal{T} \to \omega \cup \{\infty\}$ associates with every tree a natural number or ∞. We say that such a cost function f is *dominated* by g if f is bounded over every subset $X \subseteq \mathcal{T}$ over which g is bounded. We denote this domination relation by $f \preceq g$.

We can state Theorem 7.3 in terms of the domination relation as follows. Let $\mathcal{A}$ be a weighted automaton and let L be the language defined by $\mathcal{A}$ if we consider it as an ordinary parity automaton without weight function. Let f be the cost function $w_{\mathcal{A}}$ associated with $\mathcal{A}$ and let g be the cost function that maps every tree in L to 0 and every other tree to ∞. Then Theorem 7.3 states that it is decidable whether $f \preceq g$.

We would like to reduce this statement to Corollaire 8.11 of [6], which states – in the terminology of [6] – that the domination relation $f \preceq g$ between cost functions f and g is decidable, provided that f is given by a nondeterministic S-Muller automaton and g is given by a nondeterministic B-Muller automaton. The function g from above is given by a parity automaton without weight function. Such an automaton can trivially be converted into a B-Muller automaton. Hence, to complete the proof it remains to find an S-Muller automaton recognising f. This can be done in the same way as in the proof of Theorem 4.28 of [28], where the author shows how to transform an alternating B-Büchi automaton into a nondeterministic one. This proof uses game-theoretic techniques. One key argument is the fact that the games, which correspond to the automata in question, are positionally determined. To adapt the proof to our case, one needs positional determinacy for games whose winning condition is a disjunction between an unboundedness condition and a parity condition. This can be shown as in Proposition 7.14 of [6], which treats winning conditions consisting of a conjunction of a boundedness condition and a Rabin condition. One further step of adaptation consists in the construction of a so-called 'history-deterministic' automaton that checks whether a given positional strategy is winning. For finite words, the underlying translation of nondeterministic automata into history-deterministic ones can be found in an unpublished note (cf. Lemma 58 of [5]) on the author's web-page.

Using the results of the previous sections we can reduce the boundedness problem for MSO on ternary trees to Theorem 7.3. To do so, we construct a weighted automaton computing the rank of a proposal $(\mathfrak{T}, h, s, v)$. In order to use $(\mathfrak{T}, h, s, v)$ as input for a tree automaton, we encode it as a labelled directed tree with root v. The labelling contains information about the unary predicates in τ, the histories, and the synchronisation. As there is only a finite number of types, there is a uniform bound on the length of histories and we only need finitely many labels.

First we show that the set of all proposals is regular.

Lemma 7.4. *Given a formula φ, we can effectively construct a parity automaton $\mathcal{A}$ recognising the set of all proposals for φ.*

Proof. Let n be the quantifier rank of φ. It is sufficient to show that the set of proposals can be defined in MSO. Being a locally consistent annotated tree can be expressed even in FO since it is a purely local property.

For global consistency, note that we can encode a section c by a tuple of unary predicates $\bar{C}$ (the precise number depends on the maximal length of a history) such that there is an FO-formula $\vartheta_i(v)$ stating that $\bar{C}$ encodes the section at v, i. Thus, the section at v, i is MSO-definable and the corresponding tiling is MSO-interpretable. In this tiling it is of course possible by means of MSO to determine the MSO-type (of quantifier rank at most n) of a subtree. Consequently, we can express the global consistency of the tiling and, hence, also the global consistency of the annotated tree.

As the set of jumps can be inferred from the tree labelling, it is easy to check whether there is a jump at the root ($v \in \varphi^\infty$).

It remains to consider jump-consistency, that is, it remains to define $\varphi^\infty(\mathfrak{T})$ (where $\mathfrak{T}$ is the first component of the prospective proposal). As φ is positive in X, this can be achieved by

$$\psi(x) := \forall X[\forall y(\varphi(X, y) \rightarrow Xy) \rightarrow Xx] \,.$$

□

Lemma 7.5. *Given a formula φ, we can effectively construct a weighted parity automaton $\mathcal{A}$ such that*

(1) *$L(\mathcal{A})$ is the set of proposals of finite rank;*
(2) *if P is a proposal and $r < \omega$ its rank, then $\frac{1}{2}\log r \le w_{\mathcal{A}}(P) \le r$.*

Proof. Let n be the quantifier rank of φ and let $\mathcal{A}_1$ be the automaton from Lemma 7.4. We will construct the desired automaton $\mathcal{A}$ as a product of $\mathcal{A}_1$ and a weighted parity automaton $\mathcal{A}_2$, where the weight function of $\mathcal{A}$ is that of $\mathcal{A}_2$.

Recall that the rank of a proposal $P = (\mathfrak{A}, h, s, v)$ is the minimal number of jumps on a dependency chain from the jump at v to some base jump. By this minimality condition we can restrict our attention to chains without cycles. Each dependency in the chain, say from u on u', corresponds to a path in the section at u, i for a suitable i. By minimality again, we only need to consider pairwise disjoint paths, one for each dependency in the chain. (If two paths intersected, we could form a new path witnessing the dependency of some former jump in the chain to a latter one. This could be used to shorten the dependency chain.) These paths can be concatenated to form a single path in the annotated tree. For a dependency path p and a tree node u, we say that u is *active* if there is at least one jump on p in the subtree rooted at u.

Since the tree is ternary, we can encode dependency paths by a tuple of unary predicates. We first construct a weighted parity automaton $\mathcal{A}_3$ that takes as input a proposal together with such a path. It checks that the path follows the synchronisation (except for the jumps), and that it is indeed a single path. Furthermore, $\mathcal{A}_3$ is such that from its state at a node u one can deduce whether u is active. We define the weight function of $\mathcal{A}_3$ such that all transitions have weight 0 or 1, where we assign a weight of 1 if at least two children of the current node are active or if there is a jump at the current node.

For a dependency path p in a proposal P, let us compare its number r of jumps with the weight computed by $\mathcal{A}_3$. Let ϱ be any accepting run of $\mathcal{A}_3$ on the input (P, p). We claim that $\frac{1}{2}\log r \le w_{\mathcal{A}_3}(\varrho) \le r$.

For the second inequality, let β be a branch of P which realizes the maximum for ϱ, that is, $w_{\mathcal{A}_3}(\varrho) = w_{\mathcal{A}_3}(\varrho, \beta)$. With each node $u \in \beta$ such that $w_{\mathcal{A}_3}(\varrho(u)) = 1$ we associate a jump in p as follows: if there is a jump at u, we just take this jump. Otherwise, u has at least two active children, so it has at least one active child not in β. We take some jump from the subtree rooted at that child. It is clear that, for different $u \in \beta$, we have chosen different jumps. Hence, $r \ge w_{\mathcal{A}_3}(\varrho, \beta) = w_{\mathcal{A}_3}(\varrho)$.

For the other inequality, we construct a branch β as follows: the branch starts at the root and, whenever we have constructed β up to some node u which is not a leaf, we extend β

with a child u' of u such that the number of jumps on p in the subtree rooted at u' is at least as large as the respective number for any other child of u. Let us trace this number along β. Initially, it is r. It never increases and, whenever it decreases, the respective transition has weight 1 by construction of $\mathcal{A}_3$. As we always descend into the fattest subtree, the number cannot decrease indefinitely: if it is m for some node, it is at least $\frac{m-1}{3}$ for its child (recall that the original undirected tree is ternary, so the directed tree has branching at most 3, and even at most 2 apart from the root). A very rough analysis gives that, if $r \geq 4^k$, then at least k decreasing steps occur on β. Hence, $w_{\mathcal{A}_3}(\varrho) \geq w_{\mathcal{A}_3}(\varrho, \beta) \geq \frac{1}{2} \log r$.

Finally, we obtain the desired automaton $\mathcal{A}_2$ from $\mathcal{A}_3$ by nondeterministically guessing the extra component p. To see that the product automaton $\mathcal{A}$ has the claimed properties, let P be an input for $\mathcal{A}$. If P is a proposal of finite rank, then it is in particular a proposal. Hence, $\mathcal{A}_1$ accepts P. As the rank is finite, there is some dependency path p for P. Therefore, $\mathcal{A}_3$ accepts (P, p) and $\mathcal{A}_2$ accepts P. Consequently, also $\mathcal{A}$ accepts P. For the converse, assume that $\mathcal{A}$ accepts P. Then P is a proposal since $\mathcal{A}_1$ accepts P. Furthermore, there is some p such that $\mathcal{A}_3$ accepts (P, p). Thus, p is a dependency path in P and P has finite rank. Now, assume that P is a proposal of rank r and let p be a dependency path in P with r' jumps. Let ϱ be the accepting run of $\mathcal{A}_3$ on (P, p) and let ϱ' be the corresponding accepting run of $\mathcal{A}$ on P. For each accepting run of $\mathcal{A}$ on P there is such a p by construction of $\mathcal{A}$. If p is such that $w_{\mathcal{A}}(\varrho')$ is minimal, then we can deduce $\frac{1}{2} \log r \leq \frac{1}{2} \log r' \leq w_{\mathcal{A}_3}(\varrho) = w_{\mathcal{A}}(\varrho') = w_{\mathcal{A}}(P)$. If, on the other hand, p is such that r' is minimal, we obtain $w_{\mathcal{A}}(P) \leq w_{\mathcal{A}}(\varrho') = w_{\mathcal{A}_3}(\varrho) \leq r' = r$. □

Combining our results we obtain a proof of the following theorem.

Theorem 7.6. *The boundedness problem for* MSO *on the class of all ternary trees is decidable.*

Proof. Given an MSO-formula φ, we construct the weighted automaton $\mathcal{A}$ from Lemma 7.5. By Proposition 6.5, it follows that φ is bounded if, and only if, $\mathcal{A}$ is limited. The latter we can decide with the help of Theorem 7.3. □

Part II. Ramifications

The boundedness problem has long been of interest both in classical model theory and in the study of the algorithmic properties of various fragments, which in turn is partly motivated by applications in computer science. The seminal result in the classical model theory of the boundedness problem is the theorem of Barwise and Moschovakis [2] (see Theorem 8.1 below); the main interest in boundedness as a decision problem, on the other hand, stems from an interest in Datalog query optimisation as highlighted in the first positive and negative results in [14, 20]. In both contexts, the natural emphasis was on (not necessarily monadic) monotone inductions based on first-order formulae or formulae in specific fragments of first-order logic. Even in the study of rather weak fragments of first-order logic, undecidability of the boundedness problem turned out to be the rule, decidability the rare exception.

In this second part we link our new results to the wider setting of the boundedness problem. After a short introduction to this wider setting, we employ some rather more traditional tools from model theory, like transfer results and interpretations, to generalise the technical core results of Part I and to reap a number of further specific decidability results. Some of these answer key open questions raised in the more traditional setting, concerning, for instance, decidability of boundedness for the guarded fragment or for the modal μ-calculus.

To this end, we first review the shift in perspective from boundedness for syntactically restricted fragments of FO to boundedness over restricted classes of structures; a shift that was first explicitly proposed in [23] where boundedness for otherwise unconstrained monadic FO is treated over the class of acyclic structures. The class $\mathcal{A}$ of acyclic structures consists of those structures whose Gaifman graph is acyclic.

Theorem ([23])**.** The boundedness problem for monadic least fixed points of arbitrary X-positive FO-formulae over the class of all acyclic relational structures, $\mathrm{BDD}^1(\mathrm{FO}, \mathcal{A})$, is decidable. $\square$

The interest here was due to the observation that reductions to settings involving tree-like structures seem to be a common theme in most decidability results for boundedness. On the other hand, availability of grid-like structures can be widely used to show undecidability of boundedness issues via reductions from tilings [22]. This suggested a rough dichotomy to explain the borderline for decidability of (monadic) boundedness problems for fragments of FO. On the positive side, our present results bring this approach to fruition in the much wider and unifying setting of MSO. Part of this success draws on the above-mentioned change of perspective, which allows us to re-chart the relevant fragments with a decidable boundedness problem into a taxonomy of relevant classes of structures to which we can lift and extend our decidability results from Part I.

We link the more traditional approach to the boundedness problem to this new perspective in the following section: in particular, we discuss some of the more prominent fragments that have featured in the quest for decidability of boundedness so far, and review key results from that tradition.

In Sections 9 and 10 we discuss the natural model-theoretic techniques that can be used to translate and extend our results: transfer properties and reductions (Section 9) and interpretations (Sections 10). In view of the above discussion this yields results both in terms of applicability of our key result to wider classes of structures, and in terms of decidability results for new fragments.

Proviso. In this part all vocabularies are (finite and) purely relational.

8. Boundedness in the classical setting

The key result concerning boundedness from classical model theory is the following.

Theorem 8.1 (Barwise–Moschovakis [2])**.** *The following are equivalent for least fixed points based on any X-positive $\varphi(X, \bar{x}) \in \mathrm{FO}$:*

(1) *φ is bounded.*
(2) *φ^∞ is uniformly* FO*-definable.*
(3) *$\varphi^\infty(\mathfrak{A})$ is* FO*-definable in each $\mathfrak{A}$.* $\square$

The classical proof is based on compactness arguments and works with $\aleph_0$-saturated models for the crucial implication from (3) to (1). It is immediate that this argument relativises to natural fragments of FO. For formulae φ from some such fragment of FO we may replace FO-definability by definability in the fragment if that fragment has the natural closure properties that render the finite stages definable; for truly natural fragments like those to be considered below, however, FO-definability will imply definability within the fragment by classical preservation theorems.

While these considerations offer some guidelines as to what the right candidates $L \subseteq \mathrm{FO}$ for decidable $\mathrm{BDD}(L)$ might be, our results from Part I take us beyond the limitations of FO and compactness – which also means that boundedness becomes divorced from definability of the fixed point.

We start this section with a brief review of some logics and fragments that feature prominently in connection with the boundedness problem – be it in classical results or in new results flowing from our main theorem. These may be grouped into three main categories:

Existential/universal fragments: certain limited, purely existential/purely universal fragments $\mathrm{FO}_+(\exists^*) \subseteq \mathrm{FO}$ and $\mathrm{FO}_-(\forall^*) \subseteq \mathrm{FO}$: these are the natural candidates for a decidable monadic boundedness problem $\mathrm{BDD}^1(L)$ in terms of quantifier prefix classes $L \subseteq \mathrm{FO}$ (cf. the classical decision problem, [4]). For decidability of the boundedness problem extra restrictions on the polarities of the given relations, which are statically used in the fixed-point recursion, and on equality, are necessary. See Section 8.1 below.

Modal fragments: the modal fragments of first-order and monadic second-order logic: basic modal logic $\mathrm{ML} \subseteq \mathrm{FO}$ and its monadic fixed-point extension $\mathrm{L}_\mu \subseteq \mathrm{MSO}$, the bisimulation invariant fragments of FO and MSO, respectively. See Section 8.2 below.

Guarded fragments: the corresponding but more general guarded fragments: the basic guarded fragment $\mathrm{GF} \subseteq \mathrm{FO}$ and its fixed-point extension $\mu\mathrm{GF} \subseteq \mathrm{GSO}$. These correspond to the fragments of FO and guarded second-order logic GSO, respectively, that are invariant under guarded bisimulation. With these logics we also extend the scope of our discussion beyond monadic fixed points. See Section 8.3 below.

In relation to $\mathrm{BDD}(L)$ or $\mathrm{BDD}(L, \mathcal{C})$ it is useful to have in mind the following observation, which severely limits the expectations regarding decidability but also points to natural candidates.

Observation 8.2. Assume that $\mathrm{BDD}(L)$ is non-trivial in the sense that there are unbounded formulae $\varphi \in L$. Then simple closure properties of L – as for instance closure under monadic relativisation and under conjunctions – imply that the satisfiability problem $\mathrm{SAT}(L)$ reduces to the boundedness problem $\mathrm{BDD}(L)$. An analogous reduction applies w.r.t. to restricted classes of models, i.e., for $\mathrm{SAT}(L, \mathcal{C})$ and $\mathrm{BDD}(L, \mathcal{C})$ provided $\mathcal{C}$ also satisfies some simple closure requirements – as for instance closure under disjoint unions and trivial expansions by unary predicates.

We sketch one typical argument to this effect. Fix some $\varphi(X, x) \in L$ that is unbounded. Then a sentence $\psi \in L$ is unsatisfiable if, and only if, the formula $\varphi(X, x)^Q \wedge \psi^P$ is bounded; here $\varphi(X, x)^Q$ and ψ^P stand for the relativisations to two distinct unary predicates P and Q, which do not occur in either formula. Clearly, unsatisfiability of ψ implies that $\varphi(X, x)^Q \wedge \psi^P$

is unsatisfiable and hence has closure ordinal 0. Conversely, if ψ is satisfiable, then structures obtained as the disjoint union of a P-coloured model of ψ and a Q-coloured part show $\varphi(X,x)^Q \wedge \psi^P$ to be unbounded. The basic idea can be modified to suit various other situations. For instance, for modal logic, where disjoint unions are not the right choice, one could look at boundedness for $\varphi^Q \wedge \Diamond(P \wedge \psi^P)$ to decide satisfiability of ψ.

We turn to the above-mentioned groups of logics.

8.1. **Purely existential and universal fragments.** $\mathrm{FO}_+(\exists^*)[\tau] \subseteq \mathrm{FO}[\tau]$ is the fragment of positive, purely existential prenex first-order formulae (with equality), where for BDD we also allow (positive occurrences of) monadic second-order variables. Dually, we let $\mathrm{FO}_-(\forall^*)[\tau] \subseteq \mathrm{FO}[\tau]$ be the fragment of prenex universal first-order formulae that are negative in all relation symbols from the underlying relational vocabulary τ and equality, but of course we allow positive occurrences of monadic second-order variables.

The first interest in boundedness as a decision problem concerned the query language Datalog corresponding to the evaluation of systems of least fixed points of relational Horn clauses of the form

$$X\bar{x} \leftarrow \exists \bar{y} \textstyle\bigwedge_i \alpha_i(\bar{x},\bar{y})$$

with relational atomic formulae α_i. This Horn clause translates into

$$\varphi(X,\bar{x}) = \exists \bar{y} \textstyle\bigwedge_i \alpha_i(\bar{x},\bar{y}) \in \mathrm{FO}_+(\exists^*)$$

in our framework. In this connection the first decidability results were obtained in [8], and also the strict limitations for this decidable case became apparent [14, 20].

Theorem 8.3. (a) *The monadic boundedness problem* $\mathrm{BDD}^1(\mathrm{FO}_+(\exists^*))$ *is decidable* [8].

(b) *Boundedness for binary least fixed points in* $\mathrm{FO}_+(\exists^*)$ *is undecidable; so is boundedness even for monadic least fixed points in the extension of* $\mathrm{FO}_+(\exists^*)$ *that allows negated equalities (or negative and positive occurrences of some of the static relations)* [14, 20]. □

As for $\mathrm{BDD}^1(\mathrm{FO}_-(\forall^*))$, whose decidability was established in [25], it should be noted that the fragment $\mathrm{FO}_-(\forall^*)$ is strictly dual to $\mathrm{FO}_+(\exists^*)$; but as duality of fixed points links least to greatest fixed points, trivial dualisation of the Datalog result would just cover boundedness for greatest fixed points over $\mathrm{FO}_-(\forall^*)$. Indeed, the techniques employed in [25] for decidability of $\mathrm{BDD}^1(\mathrm{FO}_-(\forall^*))$ owe more to a reduction inspired by the guarded fragment (see Section 8.3 below) and also do not seem to carry over directly to $\mathrm{BDD}^1(\mathrm{FO}_+(\exists^*))$ or vice versa.

Theorem 8.4 ([25]). $\mathrm{BDD}^1(\mathrm{FO}_-(\forall^*))$ *is decidable, and both the restriction to monadic least fixed points and the polarity restriction built into* $\mathrm{FO}_-(\forall^*)$ *are necessary for decidability.* □

W.r.t. polarity restrictions on the static predicates in τ, it should be noted that, as long as we consider the class of all τ-structures, it does not matter which polarity is prescribed, since we can replace each predicate by its complement to switch between polarities (this does not carry over from to $\mathrm{BDD}(L)$ to $\mathrm{BDD}(L,\mathcal{C})$ unless $\mathcal{C}$ is closed under predicate complementation). What does matter, even over the class of all τ-structures, however, is whether we allow some predicates to appear both positively *and* negatively in φ.

8.2. **Logics of modal character.** For a relational vocabulary τ consisting of only unary and binary relation symbols, $\mathrm{ML}[\tau] \subseteq \mathrm{FO}[\tau]$ stands for the *modal fragment* of first-order logic. $\mathrm{ML}[\tau]$ is obtained as the closure of monadic atomic formulae (where we also allow monadic second-order variables besides unary relation symbols in τ) in a single free first-order variable under boolean connectives and modal quantification of the form

$$\psi(x) = \exists y(Rxy \wedge \varphi(y)) \quad \text{and, dually,} \quad \psi(x) = \forall y(Rxy \to \varphi(y))$$

for any $\varphi(y) \in \mathrm{ML}[\tau]$ and binary relation symbol $R \in \tau$.

The *modal μ-calculus* $\mathrm{L}_\mu[\tau]$ is obtained as the natural fixed-point extension of $\mathrm{ML}[\tau]$ through additional closure under least fixed points: if $\varphi(X, x) \in \mathrm{L}_\mu[\tau]$ is positive in X, then $\psi(x) = \mu_X \varphi \in \mathrm{L}_\mu[\tau]$ defines the least fixed-point φ^∞.

Our definition of ML is the usual embedding of basic modal logic into FO by means of the standard translation $\varphi \mapsto \varphi^*$, which translates the modal formula $\Box_R\varphi$ into $(\Box\varphi)^*(x) = \forall y(Rxy \to \varphi^*(y))$. By van Benthem's classical result in [27], $\mathrm{ML}[\tau]$ provides equivalent syntax for exactly those first-order formulae in a single free element variable whose semantics is preserved under bisimulation equivalence. In this sense ML *is* the bisimulation invariant (read: modal) fragment of first-order logic. (For these and other basic facts in the model theory of modal logic compare e.g. [15]).

We have similarly translated the μ-calculus in a manner that in particular turns it into a fragment of MSO. In fact L_μ *is* the modal fragment of MSO, in just the sense that ML is the modal fragment of FO, by an important result of Janin and Walukiewicz [21].

For us it will be important that $\mathrm{ML} \subseteq \mathrm{L}_\mu \subseteq \mathrm{MSO}$ and that ML and L_μ are preserved under bisimulation, which entails the tree-model property. Decidability of $\mathrm{BDD}^1(\mathrm{ML})$ was first shown in [24]; note, however, that although that paper shows more generally that it is decidable for an arbitrary formula of L_μ whether it is equivalent to any formula in plain modal logic (of which $\mathrm{BDD}^1(\mathrm{ML})$ is a special case, by the modal variant of the Barwise–Moschovakis Theorem), it does *not* deal with $\mathrm{BDD}^1(\mathrm{L}_\mu)$.

As will be reviewed in Section 9.1 below, decidability of $\mathrm{BDD}^1(\mathrm{ML})$ and $\mathrm{BDD}^1(\mathrm{L}_\mu)$ can be essentially attributed to the tree-model property stemming from bisimulation invariance. Decidability of $\mathrm{BDD}^1(\mathrm{L}_\mu)$ is new here; see Corollary 11.5 below. This result obviously implies the result of [24] concerning decidability of $\mathrm{BDD}^1(\mathrm{ML})$ (but not as far as the problem of equivalence of a given L_μ-formula to some ML-formula is concerned).

Theorem 8.5. $\mathrm{BDD}^1(\mathrm{L}_\mu)$ *and hence* $\mathrm{BDD}^1(\mathrm{ML}) \subseteq \mathrm{BDD}^1(\mathrm{L}_\mu)$ *are decidable.*

8.3. **Guarded logics.** The *guarded fragment* $\mathrm{GF} \subseteq \mathrm{FO}$ of first-order logic extends the idea of the local, relativised quantification of modal logic to the setting of higher-arity relations. Since its inception in [1] the guarded fragment and its extensions have been shown to mirror many of the nice model-theoretic properties of modal logic in this more general setting. Just like ML and its fixed-point extension L_μ, GF as well as its fixed-point extension μGF are decidable for satisfiability, cf. [1, 16, 18]. Their roles as the guarded bisimulation invariant fragments of FO and a suitable guarded second-order logic are strictly analogous to those of ML and L_μ as bisimulation invariant fragments of FO and MSO: $\mathrm{GF} \subseteq \mathrm{FO}$ captures precisely those FO definable properties that are preserved under guarded bisimulation [1], and similarly for $\mu\mathrm{GF} \subseteq \mathrm{GSO}$ w.r.t. the natural guarded second-order logic GSO, [17]. Like ML, GF still has the finite model property, and both GF and μGF have a generalised tree-model property [16, 18], which implies in particular that every satisfiable formula of $\mu\mathrm{GF}[\tau]$

is satisfiable in a model whose tree-width is bounded by the width of τ (maximal arity of relations in τ). But note that μGF does not have the finite model property, in fact this is already true of the extension of L_μ that admits modal operators along backward edges (inverse or past modalities). GF has long been considered a good candidate for decidability of BDD(GF).

Let us define these logics and the concept of guardedness in more detail. A subset of a τ-structure $\mathfrak{A}$ is called *guarded* if it is a singleton set or a set of the form $\{ a \mid a \in \bar{a} \}$ for some $\bar{a} \in R^{\mathfrak{A}}$, $R \in \tau$. Clearly the cardinality of guarded subsets in τ-structures is bounded by the width of τ. A tuple is guarded if the set of its components is contained in some guarded subset. A subset $W \subseteq A^r$ is called a *guarded relation* over $\mathfrak{A}$ if all tuples $\bar{a} \in W$ are guarded in $\mathfrak{A}$.

Syntactically, a *guard* for variables $\bar{x}$ is an atomic formula $\alpha(\bar{x}) \in \mathrm{FO}[\tau]$ (relational atom or equality) in which precisely the variables $x \in \bar{x}$ occur (as free variables).

Guarded quantification is relativised first-order quantification of the form

$$\exists \bar{y}(\alpha(\bar{x}) \wedge \varphi(\bar{x})) \quad \text{and, dually,} \quad \forall \bar{y}(\alpha(\bar{x}) \rightarrow \varphi(\bar{x}))\,,$$

where α is an atom (viz., a guard for $\bar{x}$), $\mathrm{free}(\varphi) \subseteq \mathrm{free}(\alpha) = \{ x \mid x \in \bar{x} \}$ and $\bar{y}$ is any tuple of (distinct) variables from $\mathrm{free}(\alpha)$.

Definition 8.6. (a) $\mathrm{GF}[\tau] \subseteq \mathrm{FO}[\tau]$, the *guarded fragment* of first-order logic, is obtained as the closure of atomic $\mathrm{FO}[\tau]$-formulae under boolean connectives and guarded quantification. We stress that, even if we admit a second-order variable X, X may *not* be used as a guard for quantificational purposes.

(b) *Guarded fixed-point logic* μGF is the natural extension of GF that is additionally closed under the formation of least fixed points over X-positive formulae. Note again that second-order variables, which may occur free or bound in formulae of $\mu\mathrm{GF}[\tau]$, must not be used as guards.

(c) Also define *strictly guarded* formulae of these logics to be those formulae whose free first-order variables are explicitly guarded: $\varphi(\bar{x})$ is strictly guarded if it can only be satisfied by guarded assignments to $\bar{x}$ (a syntactic normal form can be obtained with the help of the GF-formula $\mathrm{gdd}(\bar{x})$ below). We denote these restrictions as $\mathrm{GF}^* \subseteq \mathrm{GF}$ and $\mu\mathrm{GF}^* \subseteq \mu\mathrm{GF}$.

It is clear that $\mathrm{ML} \subseteq \mathrm{GF}^*$ and $\mathrm{L}_\mu \subseteq \mu\mathrm{GF}^*$. We also note in passing that there is, for every finite τ and arity r, a $\mathrm{GF}^*[\tau]$-formula $\mathrm{gdd}(x_1, \ldots, x_r)$ that uniformly defines the set of all guarded r-tuples in τ-structures $\mathfrak{A}$:

$$\{ \bar{a} \in A^r \mid (\mathfrak{A}, \bar{a}) \models \mathrm{gdd}(\bar{x}) \} = \{ \bar{a} \in A^r \mid \bar{a} \text{ guarded in } \mathfrak{A} \}\,.$$

Clearly these formulae can be used to restrict arbitrary relations to their guarded parts. For strictly guarded formulae we thus obtain a normal form of

$$\mathrm{gdd}(\bar{x}) \wedge \varphi(\bar{x})$$

where $\bar{x}$ is the tuple of all the free first-order variables of φ.

For *guarded second-order logic* there are several formalisations, which were shown to be equally expressive *in the absence of free second-order variables* in [17]. As we shall see as a consequence of Theorems 8.8 and 8.9 below, this equivalence breaks down if free second-order variables (for the generation of non-monadic least fixed points) are admitted.

Specifically, one can define GSO as the extension of either GF or FO by second-order quantifiers ranging over guarded relations. This can be enforced syntactically by means of the formulae $\mathrm{gdd}(\bar{x})$ that uniformly define the sets of all guarded r-tuples; alternatively one can stick with ordinary second-order syntax and modify the semantics to admit just guarded relations as instantiations for second-order variables (guarded semantics). The equivalence between these two definitions according to [17] breaks down in the presence of free second-order variables of arity greater than 1, since such variables are not allowed to serve as guards. Therefore, we introduce two variants of guarded second-order logic. As we shall see below, the corresponding boundedness problems are different: one is decidable for arbitrary fixed points, while the other one is only decidable for monadic fixed points.

Definition 8.7. *Guarded second-order logic* GSO[τ] is the extension of FO[τ] by quantification over guarded relations. We denote by GGSO[τ] the fragment of GSO[τ] where all first-order quantifications are guarded.

Again, we denote by GSO* and GGSO* the respective fragments of *strictly guarded* formulae, in which the tuple of free first-order variables is explicitly guarded.

Clearly GF $\subseteq$ μGF $\subseteq$ GGSO $\subseteq$ GSO. Similar inclusions hold for the corresponding strict fragments. Furthermore, MSO $\subseteq$ GSO since monadic relations are guarded (by the equality predicate). We shall see that the restriction to least fixed points that are guarded – i.e., fixed points of formulae in the starred logics – is the right counterpart, in the guarded world, for monadic fixed points. For the boundedness problem, moreover, we shall have reductions from BDD(GF) to BDD(GF*) and from BDD(GGSO) to BDD(GGSO*), see Section 10.1.

The guarded fragment GF as well as its fixed-point extension μGF are preserved under *guarded bisimulation,* the infinitary game equivalence associated to the restricted quantification pattern of guarded quantification. Guarded bisimulation equivalence plays a role for guarded logics that is analogous to the role of ordinary bisimulation for modal logics. In fact, just as modal logic is the bisimulation-invariant fragment of first-order logic [27], so GF corresponds to the fragment of first-order logic that is invariant under guarded bisimulation [1]; and just as L_μ is the bisimulation-invariant fragment of monadic second-order logic [21], so μGF corresponds to the fragment of GSO that is invariant under guarded bisimulation [17]. Note that, despite its name, GSO is not invariant under guarded bisimulation. The model theory and crucial algorithmic properties of GF and μGF are discussed in [16] and [18]. For both logics, much of their well-behavedness is due to invariance under guarded bisimulation equivalence, and, consequently, the 'generalised tree-model property' [16]: by means of a natural process of guarded tree unfolding, any structure can be transformed into a guarded bisimilar structure that admits a tree-decomposition based on guarded subsets. Hence any satisfiable formula of GF or μGF has a model which is guarded tree-decomposable so that its tree-width is bounded by the width of the underlying vocabulary.

Because of its vicinity to the modal fragment, GF has been a promising candidate for decidability of boundedness, even not just for monadic least fixed points. Approaches to BDD(GF) along those lines that worked for ML and even for FO$_-(\forall^*)$ – viz., the use of invariance under guarded bisimulation and the guarded version of the Barwise–Moschovakis theorem – have not been successful. Our present techniques do indeed yield decidability of BDD(GF), see Corollary 11.5, and thus settle a major open problem in the classical context. As we do not rely on either compactness or locality criteria in our approach, we do get a much stronger result in Theorem 11.4, concerning the decidability of BDD(GSO*, $\mathcal{W}_k$), the boundedness problem for least fixed points over GSO*-formulae over the class of all relational structures of tree-width up to k. This decidability is even uniform w.r.t. tree-width, so that both the X-positive GSO*-formula and the tree-width parameter k may be regarded as input to a single algorithm.

Theorem 8.8. *The following are decidable:* BDD(GF)*,* BDD(μGF)*,* BDD(GGSO, $\mathcal{W}_k$)*,* BDD(GSO*, $\mathcal{W}_k$)*,* BDD^1(GSO, $\mathcal{W}_k$)*.*

The transfer and reduction techniques to be discussed below immediately show that decidability for BDD(GF*) and BDD(μGF*) are an immediate consequence of decidability for BDD(GSO*, $\mathcal{W}_k$). These results essentially invoke the generalised tree-model property of GF.

As far as undecidability results are concerned, we have the following fundamental result, which follows from the proof given in [14].

Theorem 8.9. BDD(FO, $\mathcal{P}$) *is undecidable, where* $\mathcal{P}$ *is the class of all finite paths.* □

Corollary 8.10. BDD(GSO, $\mathcal{W}_k$) *is undecidable.*

In the same way, we obtain undecidability of BDD($L, \mathcal{C}$) for every logic $L \supseteq$ FO and class $\mathcal{C} \supseteq \mathcal{P}$ in which the class of all finite paths is L-definable. Examples include boundedness of MSO over the class of all trees, over the class of all finite trees, or over the class of all structures of tree-width k.

The fragments discussed so far are closed under (at least) positive boolean connectives and relativisation to unary predicates. They are also closed under the substitution operation used in defining the finite stages of fixed points. So Observation 8.2 applies to all of them and highlights the role of FO$_+(\exists^*)$, FO$_-(\forall^*)$, ML, L$_\mu$, GF and μGF as natural candidates for decidability of BDD(L). For FO, MSO and GSO on the other hand, not BDD(L) but at best BDD($L, \mathcal{C}$) for suitably restricted classes $\mathcal{C}$ can be decidable.

9. Transfer properties for BDD

Model-theoretic transfer results involving special, restricted classes of models are often useful. Key examples are provided by the finite model property or the tree-model property, which, as transfer results for satisfiability, can be useful towards establishing decidability of SAT(L). The following introduces a similar notion in connection with the boundedness problem. The most far-reaching among these properties, which in the light of our key result yields the strongest decidability consequences for the boundedness problem, is the bounded-tree-width property. We first define a general notion of transfer, then several concrete specialisations that we need in the sequel.

Definition 9.1. A logic L allows *$\mathcal{C}$-to-$\mathcal{C}'$ transfer for* BDD if, for all $\varphi \in L$, φ is bounded over $\mathcal{C}$ iff it is bounded over $\mathcal{C}'$: BDD($L, \mathcal{C}$) = BDD($L, \mathcal{C}'$).

A logic L has the *$\mathcal{C}$-property for* BDD if it allows transfer from the class of all structures to $\mathcal{C}$; i.e., if BDD(L) = BDD($L, \mathcal{C}$).

Let $\mathcal{W}_k$ stand for the class of all relational structures of tree-width up to k; similarly $\mathcal{T}_k$ stands for the class of tree models of branching degree up to k.

In accordance with the above, we say that L has the *tree-width-k property* for BDD for some concrete bound k if BDD(L) = BDD($L, \mathcal{W}_k$). In a similar spirit, one could consider transfer properties from the class of all tree models to the class of k-branching tree models, for concrete bounds k. In both cases, however, our decidability arguments require just a computable dependence of the width parameter on the input $\varphi \in L$, rather than a uniform constant bound. This motivates the following.

Definition 9.2. We say that L has the *bounded-tree-width property for* BDD if, for some computable function f, $\varphi \in L$ is bounded iff φ is bounded over $\mathcal{W}_{f(\varphi)}$ (transfer to models of bounded tree-width).

Similarly, L has the *bounded-branching property for* BDD *over trees* if, for some computable function f, $\varphi \in L$ is bounded over the class of all tree models iff it is bounded over $\mathcal{T}_{f(\varphi)}$ (transfer to tree models of bounded branching).

In all natural cases a $\mathcal{C}$-model property (transfer for $\mathrm{SAT}(L)$) implies a $\mathcal{C}$-property for BDD. This is clearly the case if L is closed under the kind of substitution used to define the finite stages and under boolean connectives. In that case, the finite stages φ^α for $\alpha < \omega$ and the finite stage increments $\varphi^{\alpha+1} \wedge \neg\varphi^\alpha$ are definable by formulae in L and φ is unbounded iff all these formulae are satisfiable.

Concerning the finite model property for BDD, note that (even for fragments $L \subseteq \mathrm{FO}$) it does not imply decidability of $\mathrm{BDD}(L)$: one still would need to check satisfiability for each member of the infinite family $\varphi^{\alpha+1} \wedge \neg\varphi^\alpha$ (albeit just in finite models).

9.1. **Transfer results for classical fragments.**

We collect some transfer results for the fragments and logics discussed in the last section.

Observation 9.3.
(a) $\mathrm{FO}_+(\exists^*)$, $\mathrm{FO}_-(\forall^*)$, ML, L_μ and GF have the *finite model property* for BDD just as for SAT.
(b) ML and L_μ have the *tree-property* for SAT and BDD; ML even allows transfer to *finite* tree-models of bounded branching.
(c) $\mathrm{FO}_+(\exists^*)$, $\mathrm{FO}_-(\forall^*)$, ML, L_μ, GF and μGF all have the *bounded-tree-width property* for SAT and BDD. Among these, the modal logics ML, L_μ even allow transfer to tree models of bounded branching; $\mathrm{FO}_+(\exists^*)$, $\mathrm{FO}_-(\forall^*)$, GF and μGF allow transfer to models of bounded tree-width, in the case of $\mathrm{FO}_+(\exists^*)$, $\mathrm{FO}_-(\forall^*)$, GF even to *finite* models of bounded tree-width.

More specifically, the necessary tree-width k in (c) can be bounded by the width of the underlying vocabulary τ in the modal and guarded cases, and (for a rough bound) by the size of the given formula φ in the case of $\mathrm{FO}_+(\exists^*)$, $\mathrm{FO}_-(\forall^*)$.

Most of these statements follow from corresponding properties for $\mathrm{SAT}(L)$, which are well known from the literature (cf. in particular Observation 8.2 above). The bounded-tree-width property for BDD in the case of GF and μGF is a direct consequence of preservation of these logics under guarded bisimulation. Guarded tree-unfoldings [16, 17] of arbitrary models yield models possessing a tree decomposition whose bags are guarded subsets, hence of width bounded by the width of τ. For the assertions concerning the fragments $\mathrm{FO}_+(\exists^*)$ and $\mathrm{FO}_-(\forall^*)$, which are not closed under negation, we prove the following lemma.

Lemma 9.4. *$\mathrm{FO}_+(\exists^*)[\tau]$ and $\mathrm{FO}_-(\forall^*)[\tau]$ allow transfer for* BDD[1] *to finite models of bounded tree-width.*[1]

[1]Here tree-width can be bounded by the size of the given prenex formula $\varphi(X, x)$; a better bound would be the tree-width of the quantifier-free kernel formula.

Proof. We explicitly treat the case of $\mathrm{FO}_+(\exists^*)$; the argument for $\mathrm{FO}_-(\forall^*)$ is strictly analogous.

For X-positive $\varphi(X,x) \in \mathrm{FO}_+(\exists^*)[\tau]$ and finite $\alpha < \omega$, the stage increment $\varphi^{\alpha+1}(\mathfrak{A}) \setminus \varphi^{\alpha}(\mathfrak{A})$ is uniformly definable by a conjunction of a purely existential formula $\varphi^{\alpha+1}(x) \in \mathrm{FO}_+(\exists^*)[\tau]$ and a purely universal formula in $\mathrm{FO}_-(\forall^*)[\tau]$ equivalent to the negation of $\varphi^{\alpha}(x)$. Formulae of this kind are known to have the finite model property:[2] from an arbitrary model $(\mathfrak{A}, a)$ of some conjunction of a prenex $\exists^*$-formula $\psi_1(x)$ and a prenex $\forall^*$-formula $\psi_2(x)$, one obtains a finite model by restricting $\mathfrak{A}$ to a together with any chosen instantiation for the existentially quantified variables in ψ_1; this restriction still satisfies ψ_1, and as an induced substructure of $(\mathfrak{A}, a) \models \psi_2$ it also still satisfies the universal formula ψ_2.

To obtain suitable (finite) models of bounded tree-width, though, we need to consider the stronger preservation properties of the formulae $\varphi^{\alpha}(x) \in \mathrm{FO}_+(\exists^*)[\tau]$, and to some extent use the polarity constraints in $\mathrm{FO}_+(\exists^*)$ and $\mathrm{FO}_-(\forall^*)$. The following argument also makes an interesting connection with GF.

Let w.l.o.g. φ be of the form

$$\varphi(X,x) = \exists \bar{y} \bigvee_i \big(\rho_i(x,\bar{y}) \wedge \bigwedge_{j \in s_i} X y_j\big)$$

where $\bar{y} = (y_1, \ldots, y_k)$, the ρ_i are conjunctions of relational τ-atoms (not involving X), and $s_i \subseteq \{1, \ldots, k\}$. For any τ-structure $\mathfrak{A}$ let $\hat{\mathfrak{A}}$ be its expansion to a $\hat{\tau}$-structure by new relations R_i of arity $k+1$, with R_i defined by ρ_i. In $\hat{\mathfrak{A}}$, φ is equivalent to the GF^*-formula

$$\hat{\varphi}(X,x) = \exists \bar{y} \bigvee_i \big(R_i x \bar{y} \wedge \bigwedge_{j \in s_i} X y_j\big)\,.$$

An analogous equivalence obtains for formulae $\varphi^{\alpha}(x) \in \mathrm{GF}[\tau]$ and $\hat{\varphi}^{\alpha}(x) \in \mathrm{GF}[\hat{\tau}]$ defining the finite stages w.r.t. φ and $\hat{\varphi}$.

Obviously

$$\bigwedge_i \forall x \forall \bar{y} \big(R_i x \bar{y} \to \rho_i(x,\bar{y})\big) \models \forall x \big(\hat{\varphi}^{\alpha}(x) \to \varphi^{\alpha}(x)\big)\,, \tag{$*$}$$

where the formula on the left-hand side is in $\mathrm{GF}[\hat{\tau}]$. Note, however, that implications of the form $\forall x \forall \bar{y} \big(\rho_i(x,\bar{y}) \to R_i x \bar{y}\big)$, which would be needed towards the equivalence between φ^{α} and $\hat{\varphi}^{\alpha}$ cannot in general be expressed in GF.

Let $\hat{\mathfrak{A}}^*$ be a guarded bisimilar unfolding of $\hat{\mathfrak{A}}$. Its tree-width is bounded by the maximum of the width of τ and $k+1$. We also write $\mathfrak{A}^*$ for the τ-reduct of $\hat{\mathfrak{A}}^*$. Let $\pi : \hat{\mathfrak{A}}^* \to \hat{\mathfrak{A}}$ be the projection from the unfolding onto the base structure; π is a homomorphism inducing the natural guarded bisimulation between $\hat{\mathfrak{A}}^*$ and $\hat{\mathfrak{A}}$. Preservation of $\mathrm{GF}[\hat{\tau}]$ under guarded bisimulations implies that, for all $\alpha < \omega$,

$$\hat{\mathfrak{A}}^*, a \models \hat{\varphi}^{\alpha} \quad \text{iff} \quad \hat{\mathfrak{A}}, \pi(a) \models \hat{\varphi}^{\alpha}.$$

Since $\mathfrak{A}, \pi(a) \models \varphi^{\alpha}$ implies $\hat{\mathfrak{A}}, \pi(a) \models \hat{\varphi}^{\alpha}$ and, therefore, also $\hat{\mathfrak{A}}^*, a \models \hat{\varphi}^{\alpha}$, it follows with $(*)$ above that $\mathfrak{A}, \pi(a) \models \varphi^{\alpha}$ implies $\mathfrak{A}^*, a \models \varphi^{\alpha}$.

In the opposite direction, since the φ^{α}, as existential positive formulae, are preserved under homomorphisms, the implication $\mathfrak{A}^*, a \models \varphi^{\alpha} \Rightarrow \mathfrak{A}, \pi(a) \models \varphi^{\alpha}$ is straightforward.

[2] They fall in particular within the Bernays–Schönfinkel class of prenex FO-formulae with quantifier prefix $\exists^*\forall^*$, cf. [4], but a more direct argument suffices here.

Therefore, for all $a \in \mathfrak{A}^*$ and all $\alpha < \omega$,

$$\mathfrak{A}^*, a \models \varphi^\alpha \quad \text{iff} \quad \mathfrak{A}, \pi(a) \models \varphi^\alpha,$$

whence $\|\varphi\|_{\mathfrak{A}} = \|\varphi\|_{\mathfrak{A}^*}$. Hence φ is bounded iff it is bounded over structures whose tree-width is bounded by the maximum of the width of τ and $k+1$. In order to get back to finite models of bounded tree-width, we may apply the simple argument from above to find a finite induced substructure within some $(\mathfrak{A}^*, a)$ that still satisfies the corresponding $\exists^*/\forall^*$-conjunction $\varphi^{\alpha+1}(x) \wedge \neg\varphi^\alpha(x)$. ☐

9.2. **Transfer for MSO over trees.** At the level of MSO we obtain a bounded-branching property for BDD^1 over trees. The availability of transfer at least down to countable branching is essential to make a connection via interpretations with our core result that was formulated over ternary trees.

Proposition 9.5. MSO *has a countable branching property for monadic* BDD^1 *over trees.*

This statement follows immediately from Proposition 9.8 below, whose proof relies on the availability of tree automata for MSO and involves, as a key step, a Löwenheim–Skolem property for MSO-theories of trees. We employ a certain kind of tree automata introduced by Walukiewicz [29].

Definition 9.6. An MSO-*automaton* is a tuple $\mathcal{A} = \langle Q, \Sigma, q_0, \delta, \Omega \rangle$ with a finite set of states Q, an input alphabet Σ, an initial state q_0, a parity function $\Omega : Q \to \omega$, and a transition function $\delta : Q \times \Sigma \to \mathrm{MSO}$ that, given a state q and a letter c, returns an MSO-formula $\delta(q, c)$ over the signature $\{ P_q \mid q \in Q \}$.

Such an automaton takes a Σ-labelled directed tree $t = (T, E, \lambda)$ as input. A *run* of $\mathcal{A}$ on t is a function $\varrho : T \to Q$ with $\varrho(r) = q_0$ for the root r of (T, E) such that

$$\langle U_v, \bar{P} \rangle \models \delta(q, \lambda(v)), \quad \text{for all } v \in T,$$

where the universe U_v of the structure is the set of all children of v and the unary predicates are $P_p := \varrho^{-1}(p) \cap U_v$. The run ϱ is *accepting* if, and only if, for all infinite branches $v_0 v_1 \dots$ of (T, E)

$$\liminf_{n \to \infty} \Omega(\varrho(v_n)) \text{ is even.}$$

The *language recognised by* $\mathcal{A}$ is the set $L(\mathcal{A})$ of all trees t such that there exists an accepting run of $\mathcal{A}$ on t.

Over trees these automata have the same expressive power as monadic second-order logic.

Theorem 9.7 (Walukiewicz [29])**.** *A class $\mathcal{C}$ of directed trees is definable by an* MSO-*sentence φ if, and only if, it is recognised by some* MSO-*automaton $\mathcal{A}$.* ☐

We use MSO-automata to prove the following Löwenheim-Skolem theorem.

Proposition 9.8. *For every tree structure $\mathfrak{T}$ there exists a countable tree structure $\mathfrak{T}_0 \subseteq \mathfrak{T}$ with the same* MSO*-theory.*

Proof. We prove the proposition for directed trees. Then the corresponding claim for undirected trees follows. Suppose that $\mathfrak{T}$ is a directed tree with root r. Let us call a substructure $\mathfrak{T}_0 \subseteq \mathfrak{T}$ a *subtree* of $\mathfrak{T}$ if $\mathfrak{T}_0$ is a tree and it contains the root r.

To prove the claim, we construct a countable subtree $\mathfrak{T}_0 \subseteq \mathfrak{T}$ such that every MSO-automaton accepting $\mathfrak{T}$ also accepts $\mathfrak{T}_0$. Since every MSO-formula is equivalent (on trees) to an MSO-automaton and since MSO is closed under complement, it follows that $\mathfrak{T}$ and $\mathfrak{T}_0$ have the same MSO-theory.

To construct $\mathfrak{T}_0$ we proceed as follows. For every MSO-automaton $\mathcal{A}$ that accepts $\mathfrak{T}$ and every vertex $v \in T$, we fix a countable set $S_{\mathcal{A}}(v) \subseteq T$ of children of v such that the following holds:

($*$) Every subtree $\mathfrak{T}_0 \subseteq \mathfrak{T}$ such that

$$v \in T_0 \quad \text{implies} \quad S_{\mathcal{A}}(v) \subseteq T_0$$

is accepted by $\mathcal{A}$.

Let us call a subtree $\mathfrak{T}_0$ *$S_{\mathcal{A}}$-closed* if $v \in T_0$ implies $S_{\mathcal{A}}(v) \subseteq T_0$. We take for T_0 the minimal subset of T containing the root r that is $S_{\mathcal{A}}$-closed for every $\mathcal{A}$ accepting $\mathfrak{T}$. The subtree $\mathfrak{T}_0$ induced by T_0 is countable and has the desired property.

To define $S_{\mathcal{A}}$ we fix an accepting run ϱ of $\mathcal{A}$ on $\mathfrak{T}$. Let $v \in T$ be a vertex with label c and let U be the set of children of v in T. For each state $q \in \varrho(v)$, ϱ induces a structure $\langle U, \bar{P} \rangle$ satisfying the transition formula $\delta(q, c)$. For every state $p \in Q$, we select a set $X_p^q \subseteq P_p = \varrho^{-1}(p) \cap U$ as follows. If P_p is countable, we set $X_p^q := P_p$. Otherwise, we choose an arbitrary countably infinite subset $X_p \subseteq P_p$. Then we set

$$S_{\mathcal{A}}(v) := \bigcup_{q \in Q} \bigcup_{p \in Q} X_p^q \,.$$

We claim that, for every $S_{\mathcal{A}}$-closed subtree $\mathfrak{T}_0 \subseteq \mathfrak{T}$, the restriction of ϱ to T_0 is an accepting run of $\mathcal{A}$ on $\mathfrak{T}_0$. Obviously, every infinite branch of $\mathfrak{T}_0$ is an infinite branch of $\mathfrak{T}$ and, hence, satisfies the parity condition. So we only need to check that the transition formulae hold at each vertex. Let $v \in T$ be a vertex with label c and with set of children U, and let $\langle U, \bar{P} \rangle$ be the structure induced by ϱ. Since ϱ is a run, we have

$$\langle U, \bar{P} \rangle \models \delta(\varrho(v), c) \,.$$

Note that the structure $\langle U, \bar{P} \rangle$ has only unary relations. There is a well-known Ehrenfeucht-Fraïssé argument showing that an MSO-sentence of quantifier rank m cannot distinguish two such structures $\langle U, \bar{P} \rangle$ and $\langle U', \bar{P}' \rangle$, provided that each quantifier-free 1-type is realised the same number of times in both structures, or it is realised at least 2^m times in each structure.

By definition of $S_{\mathcal{A}}(v)$, it follows that, for all subsets $U_0 \subseteq U$ containing $S_{\mathcal{A}}(v)$, the structures

$$\langle U, \bar{P} \rangle \quad \text{and} \quad \langle U_0, \bar{P}|_{U_0} \rangle$$

have the same MSO-theory. Consequently,

$$\langle U_0, \bar{P}|_{U_0} \rangle \models \delta(\varrho(v), c) \,.$$

In particular, this is the case for $U_0 := U \cap T_0$. Therefore, $\varrho \restriction T_0$ is a run. $\square$

Remark 9.9. If, instead of the full MSO-theory, we are only interested in the preservation of a single MSO-sentence, the construction of the theorem yields a tree that is finitely branching.

Proof of Proposition 9.5. Clearly, if an MSO-formula $\varphi(X, x)$ is bounded over the class of all trees, it is also bounded over the class of all countable trees. Conversely, suppose that $\varphi(X, x)$ is unbounded over arbitrary trees. Then we can find, for every $\alpha < \omega$, a tree $\mathfrak{T}_\alpha$ satisfying the formula $\psi_\alpha := \exists x[\varphi^{\alpha+1}(x) \wedge \neg\varphi^\alpha(x)]$. By the above proposition, we can choose $\mathfrak{T}_\alpha$ to be countably branching. Hence, $\varphi(X, x)$ is also unbounded over the class of all countably branching trees. $\square$

10. Interpretations and reductions

In the preceding section we have considered transfer of $\mathrm{BDD}(L, \mathcal{C})$ from one class $\mathcal{C}$ to a subclass $\mathcal{C}_0 \subseteq \mathcal{C}$. In this section we will study more general reductions of $\mathrm{BDD}(L, \mathcal{C})$ to $\mathrm{BDD}(L', \mathcal{C}')$ where both the logic L and the class $\mathcal{C}$ may change.

10.1. A reduction for GF.

We start by reducing $\mathrm{BDD}(\mathrm{GGSO}, \mathcal{C})$ to $\mathrm{BDD}(\mathrm{GGSO}^*, \mathcal{C})$. The following normal form for GGSO-formulae is used in the proof of the proposition below.

Lemma 10.1. *Let $\varphi(\bar{R}, X, \bar{x})$ be a* GGSO*-formula with free second-order variables $\bar{R}$, X and free first-order variables $\bar{x}$ that is positive in X. We can effectively construct* GGSO*-formulae ψ_i^0, ψ_i^1, for $i < n$, such that*

$$\varphi(\bar{R}, X, \bar{x}) \equiv \bigvee_{i<n} \left[\psi_i^0(X, \bar{x}) \wedge \psi_i^1(\bar{R}, X, \bar{x})\right],$$

where

- *the formulae ψ_i^0 are quantifier-free and positive in X,*
- *the formulae ψ_i^1 are positive in X and such that X only appears in subformulae of the form $\forall\bar{y}\vartheta$ and $\exists\bar{y}\vartheta$.*

Furthermore, if φ is a GF*-formula, then so are the formulae ψ_i^0, ψ_i^1, $i < n$.*

Proof. We may assume that φ is in negation normal form. The claim follows by induction on the structure of φ. All other cases being trivial, we present only the case of second-order quantifiers.

Hence, let us assume that $\varphi = \mathsf{Q}Z\vartheta$, for $\mathsf{Q} \in \{\forall, \exists\}$. By inductive hypothesis, we may assume that

$$\vartheta = \bigvee_{i<n} [\psi_i^0(X, \bar{x}) \wedge \psi_i^1(\bar{R}, Z, X, \bar{x})]$$

with ψ_i^0 and ψ_i^1 as in the statement of the lemma. In case of an existential quantifier, we are done since

$$\exists Z\vartheta \equiv \bigvee_{i<n} [\psi_i^0(X, \bar{x}) \wedge \exists Z\psi_i^1(\bar{R}, Z, X, \bar{x})].$$

For a universal quantifier, note that

$$\begin{aligned}
\forall Z\vartheta &= \forall Z \bigvee_{i<n} \big[\psi_i^0(X,\bar{x}) \wedge \psi_i^1(\bar{R},Z,X,\bar{x})\big] \\
&\equiv \forall Z \bigwedge_{\sigma\in 2^n} \bigvee_{i<n} \psi_i^{\sigma(i)} \\
&\equiv \bigwedge_{\sigma\in 2^n} \forall Z \Big[\bigvee_{i\in\sigma^{-1}(0)} \psi_i^0 \vee \bigvee_{i\in\sigma^{-1}(1)} \psi_i^1 \Big] \\
&\equiv \bigwedge_{\sigma\in 2^n} \Big[\bigvee_{i\in\sigma^{-1}(0)} \psi_i^0 \vee \forall Z \bigvee_{i\in\sigma^{-1}(1)} \psi_i^1 \Big] .
\end{aligned}$$

Hence, the claim follows by another application of the distributive law. □

Proposition 10.2. *For every formula $\varphi(X,\bar{x}) \in \mathrm{GGSO}[\tau]$, we can effectively construct a formula $\varphi^g(X,\bar{x}) \in \mathrm{GGSO}^*[\tau]$ such that $\varphi(X,\bar{x})$ is bounded if, and only if, $\varphi^g(X,\bar{x})$ is.*

Furthermore, if φ is a GF*-formula, then so is φ^g.*

Proof. By the lemma we may assume that the formula $\varphi(X,\bar{x})$ has the form

$$\varphi(X,\bar{x}) = \bigvee_{i<n} \big[\chi_i(X,\bar{x}) \wedge \psi_i(X,\bar{x})\big] ,$$

where the formulae χ_i are quantifier-free and in the formulae ψ_i every occurrence of X is in a subformula of the form $\forall\bar{y}\vartheta$ and $\exists\bar{y}\vartheta$.

Note that any occurrence of an atom $X\bar{y}$ that is in the scope of some (guarded!) first-order quantification may be replaced by the formula $X^g\bar{y} := X\bar{y} \wedge \mathrm{gdd}(\bar{y})$ without changing its semantics. Therefore,

$$\varphi(X,\bar{x}) \equiv \bigvee_{i<n} \big[\chi_i(X,\bar{x}) \wedge \psi_i(X^g,\bar{x})\big] ,$$

where $\psi_i(X^g,\bar{x}) := \psi_i(X^g/X,\bar{x})$ is the formula obtained from ψ_i by replacing X by its guarded restriction X^g without affecting the semantics. In the following a superscript ^{g} is always used to indicate syntactic and/or semantic restriction to the guarded part.

The fixed-point induction of $\varphi(X,\bar{x})$ is closely related to the fixed-point induction of the strictly guarded formula

$$\varphi^g(X,\bar{x}) := \mathrm{gdd}(\bar{x}) \wedge \varphi(X,\bar{x}) \equiv \bigvee_{i<n} \big[\mathrm{gdd}(\bar{x}) \wedge \chi_i(X,\bar{x}) \wedge \psi_i(X^g,\bar{x})\big] .$$

Since φ^g implies φ and both formulae are positive in X, it follows that the stages of φ^g are included in those for φ. In fact, it follows by a simple induction on α that $(\varphi^g)^\alpha(\mathfrak{A}) = (\varphi^\alpha(\mathfrak{A}))^g$. Consequently, we have

$$\|\varphi^g\|_{\mathfrak{A}} \leq \|\varphi\|_{\mathfrak{A}} \quad \text{and} \quad (\varphi^\infty(\mathfrak{A}))^g = (\varphi^g)^\infty(\mathfrak{A}) .$$

If we can show that there exists a constant $n < \omega$, depending only on φ, such that, for all structures $\mathfrak{A}$,

$$\|\varphi\|_{\mathfrak{A}} \leq \|\varphi^g\|_{\mathfrak{A}} + n ,$$

then it follows that φ is bounded if, and only if, φ^g is bounded.

To find the constant n, we consider the auxiliary formula

$$\xi(Z, X, \bar{x}) := \bigvee_{i<n} [\chi_i(X, \bar{x}) \wedge \psi_i(Z, \bar{x})]$$

in vocabulary $\tau \cup \{Z\}$ (with a new second-order variable Z of the same arity r as X, which is regarded as a parameter) and we consider its fixed-point induction in the expansion $(\mathfrak{A}, P_0)$ of $\mathfrak{A}$ where Z is interpreted by the relation $P_0 := (\varphi^g)^\infty(\mathfrak{A})$. We claim that

$$\|\varphi\|_\mathfrak{A} \leq \|\varphi^g\|_\mathfrak{A} + \|\xi\|_{(\mathfrak{A}, P_0)} .$$

Let $\gamma := \|\varphi^g\|_\mathfrak{A}$. We have shown above that $P_0 = (\varphi^g)^\gamma(\mathfrak{A}) \subseteq \varphi^\gamma(\mathfrak{A})$. Using monotonicity, it follows by a simple induction on α that

$$\varphi^\alpha(\mathfrak{A}) \subseteq \xi^\alpha(\mathfrak{A}, P_0) \subseteq \varphi^{\gamma+\alpha}(\mathfrak{A}) .$$

The first inclusion implies that $\varphi^\infty(\mathfrak{A}) \subseteq \xi^\infty(\mathfrak{A}, P_0)$ while the second inclusion implies that $\xi^\infty(\mathfrak{A}, P_0) \subseteq \varphi^\infty(\mathfrak{A})$. Setting $\beta := \|\xi\|_{(\mathfrak{A}, P_0)}$, it follows that

$$\varphi^\infty(\mathfrak{A}) = \xi^\beta(\mathfrak{A}, P_0) \subseteq \varphi^{\gamma+\beta}(\mathfrak{A}) \subseteq \varphi^\infty(\mathfrak{A}) .$$

Hence, $\|\varphi\|_\mathfrak{A} \leq \gamma + \beta$, as desired.

We have shown that, for every structure $\mathfrak{A}$,

$$\|\varphi^g\|_\mathfrak{A} \leq \|\varphi\|_\mathfrak{A} \leq \|\varphi^g\|_\mathfrak{A} + \|\xi\|_{(\mathfrak{A}, P_0)} .$$

To conclude the proof it remains to prove that $\|\xi\|_{(\mathfrak{A}, P_0)}$ is uniformly bounded. Note that ξ treats Z as a static parameter, and only involves its fixed-point variable X outside the scope of any quantifiers. It follows that ξ is trivially bounded (with a bound that is given by the number of quantifier-free r-types in vocabulary $\tau \cup \{Z\}$). □

Hence we may restrict attention to fixed points over strictly guarded formulae. This means that BDD(GGSO) reduces to BDD(GGSO*). Let us remark that a corresponding result for GSO fails.

An argument analogous to the above also applies to μGF: according to [17], every μGF-formula is equivalent to one where every fixed-point operator is applied to a strictly guarded formula. For μGF-formulae of this form, a variant of Lemma 10.1 holds. This is all we need for the proof of Proposition 10.2. Consequently, BDD(μGF) reduces to BDD(μGF*).

10.2. **MSO-interpretations in trees.** In the first part we have obtained the decidability of $\mathrm{BDD}^1(\mathrm{MSO}, \mathcal{T}_3)$. In this section, we use model-theoretic interpretations to reduce the decidability of $\mathrm{BDD}^1(\mathrm{MSO}, \mathcal{W}_k)$ to this problem.

Definition 10.3. Let σ and τ be relational signatures.

(a) A *definition scheme* for an MSO-interpretation from σ to τ is a list

$$\mathcal{I} = \langle \chi, \delta(x), \varepsilon(x, y), (\varphi_R(\bar{x}))_{R\in\sigma} \rangle$$

of MSO[τ]-formulae where χ is a sentence, $\delta(x)$ has one free variable, $\varepsilon(x, y)$ has two, and the number of free variables of $\varphi_R(\bar{x})$ equals the arity of the relation symbol R.

(b) The *operation defined* by a definition scheme $\mathcal{I}$ maps τ-structures $\mathfrak{A}$ to σ-structures $\mathcal{I}(\mathfrak{A})$. A τ-structure $\mathfrak{A}$ such that $\mathfrak{A} \models \chi$, $\delta[\mathfrak{A}] \neq \emptyset$ and such that ε defines an equivalence relation $\sim$ on $\delta[\mathfrak{A}]$, is mapped to the σ-structure $\mathfrak{B}$ with universe

$$B := \{ [a]_\sim \in A/{\sim} \mid \mathfrak{A} \models \delta(a) \}$$

and, for each n-ary relation $R \in \sigma$, the relation

$$R^{\mathfrak{B}} := \{\, [\bar{a}]_\sim \in A^n/{\sim} \mid \mathfrak{A} \models \varphi_R(\bar{a}) \,\}\,.$$

For any other τ-structure $\mathfrak{A}$, we let $\mathcal{I}(\mathfrak{A})$ be undefined.

(c) An MSO-*interpretation* is an operation defined by a definition scheme $\mathcal{I}$. If $\mathcal{C}$ is a class of τ-structures, we set

$$\mathcal{I}(\mathcal{C}) := \{\, \mathcal{I}(\mathfrak{A}) \mid \mathfrak{A} \in \mathcal{C} \text{ such that } \mathcal{I}(\mathfrak{A}) \text{ is defined} \,\}\,.$$

For the proof of Proposition 10.5 below, let us recall the following well-known lemma. We include a proof, so that we may refer to a precise format of the formulae $\psi^{\mathcal{I}}$ later.

Lemma 10.4 (Interpretation Lemma). *Let $\mathcal{I} = \big\langle \chi, \delta(x), \varepsilon(x,y), (\varphi_R(\bar{x}))_{R\in\sigma} \big\rangle$ be an* MSO-*interpretation. For every* MSO$[\sigma]$*-formula ψ, there exists an* MSO$[\tau]$*-formula $\psi^{\mathcal{I}}$ such that, for all τ-structures $\mathfrak{A}$ and every tuple $\bar{a}$ in A, we have*

$$\mathfrak{A} \models \psi^{\mathcal{I}}(\bar{a}) \quad \text{iff} \quad \mathcal{I}(\mathfrak{A}) \text{ is defined, } \mathfrak{A} \models \delta(a_i) \text{ for all } i, \text{ and } \mathcal{I}(\mathfrak{A}) \models \psi([\bar{a}]_\sim)\,.$$

If ψ is positive in a predicate X and the formula $\varphi_X(\bar{x})$ from $\mathcal{I}$ is positive in a predicate Y, then $\psi^{\mathcal{I}}$ is also positive in Y.

Proof. First, we define a formula ψ^* by induction on ψ as follows:

$$(R\bar{c})^* := \exists\bar{z}\big[\textstyle\bigwedge_i \varepsilon(z_i,c_i) \wedge \varphi_R(\bar{z})\big]\,, \qquad (\exists y\vartheta)^* := \exists y[\delta(y) \wedge \vartheta^*]\,,$$
$$(c = d)^* := \varepsilon(c,d)\,, \qquad (\forall y\vartheta)^* := \forall y[\delta(y) \to \vartheta^*]\,,$$

and the translation $\cdot^*$ commutes with boolean operations and set quantifiers.

Then we can set

$$\psi^{\mathcal{I}} := \chi' \wedge \bigwedge_i \delta(x_i) \wedge \psi^*$$

where the conjunction is over all free variables of ψ and $\chi' := \chi \wedge \eta$ is the conjunction of χ with a formula η stating that ε defines an equivalence relation on δ. □

Proposition 10.5. *Let $\mathcal{I}$ be an* MSO*-interpretation and $\mathcal{C}$ a class of τ-structures. If* BDD^1(MSO$,\mathcal{C}$) *is decidable then so is* BDD^1(MSO$,\mathcal{I}(\mathcal{C})$)*.*

Proof. We use the same notation as in the proof of Lemma 10.4. Suppose that $\mathcal{I} = \langle \chi, \delta, \varepsilon, (\varphi_R)_R \rangle$ and let $\psi(x,X)$ be a formula over the signature $\sigma \cup \{x, X\}$. We extend the notation ϑ^* from above to formulae containing a free set variable X by treating X as a relation defined by the formula Xx, i.e., we set

$$(Xc)^* := \exists z[\varepsilon(z,c) \wedge Xz]\,.$$

Let $\mathfrak{A} \in \mathcal{C}$ be a structure such that $\mathcal{I}(\mathfrak{A})$ is defined. Note that Lemma 10.4 implies that

$$\mathfrak{A} \models \forall x \forall y\big[\varepsilon(x,y) \to (\vartheta^*(x) \leftrightarrow \vartheta^*(y))\big]\,, \quad \text{for every formula } \vartheta(x)\,.$$

For formulae $\vartheta(x)$ and $\psi(X,x)$, it follows by induction on the structure of ψ that

$$\mathfrak{A} \models \forall x\big[\delta(x) \to \big((\psi[\vartheta/X])^* \leftrightarrow \psi^*[\vartheta^*/X]\big)\big]\,.$$

A simple induction on α yields

$$\mathfrak{A} \models \forall x\big[\delta(x) \to \big((\psi^\alpha)^* \leftrightarrow (\psi^*)^\alpha\big)\big]\,.$$

Since $\mathfrak{A} \models \chi'$, it follows by the definition of the mapping $\vartheta \mapsto \vartheta^{\mathcal{I}}$ that, for every $\alpha < \omega$, we have

$$\begin{aligned}
&\mathcal{I}(\mathfrak{A}) \models \forall x(\psi^{\alpha+1} \leftrightarrow \psi^{\alpha}) \\
\text{iff}\quad &\mathfrak{A} \models \forall x(\psi^{\alpha+1} \leftrightarrow \psi^{\alpha})^{\mathcal{I}} \\
\text{iff}\quad &\mathfrak{A} \models \chi' \wedge \forall x\big[\delta(x) \rightarrow ((\psi^{\alpha+1})^* \leftrightarrow (\psi^{\alpha})^*)\big] \\
\text{iff}\quad &\mathfrak{A} \models \forall x\big[(\chi' \wedge \delta(x) \wedge (\psi^{\alpha+1})^*) \leftrightarrow (\chi' \wedge \delta(x) \wedge (\psi^{\alpha})^*)\big] \\
\text{iff}\quad &\mathfrak{A} \models \forall x\big[(\chi' \wedge \delta(x) \wedge (\psi^*)^{\alpha+1}) \leftrightarrow (\chi' \wedge \delta(x) \wedge (\psi^*)^{\alpha})\big] \\
\text{iff}\quad &\mathfrak{A} \models \forall x\big[(\chi' \wedge \delta(x) \wedge \psi^*)^{\alpha+1} \leftrightarrow (\chi' \wedge \delta(x) \wedge \psi^*)^{\alpha}\big] \\
\text{iff}\quad &\mathfrak{A} \models \forall x[(\psi^{\mathcal{I}})^{\alpha+1} \leftrightarrow (\psi^{\mathcal{I}})^{\alpha}\big]\,.
\end{aligned}$$

Consequently, ψ is bounded over $\mathcal{I}(\mathcal{C})$ if and only if $\psi^{\mathcal{I}}$ is bounded over $\mathcal{C}$. □

Corollary 10.6. *Let $\mathcal{C}$ be a class of τ-structures and ψ an* MSO*-formula. If* $\mathrm{BDD}^1(\mathrm{MSO}, \mathcal{C})$ *is decidable then so is* $\mathrm{BDD}^1(\mathrm{MSO}, \mathcal{C}_\psi)$ *where*

$$\mathcal{C}_\psi := \{\, \mathfrak{A} \in \mathcal{C} \mid \mathfrak{A} \models \psi \,\}\,.$$

Proof. We can use the interpretation $\mathcal{I} = \langle \chi, \delta, \varepsilon, (\varphi_R) \rangle$ with

$$\begin{aligned}
\chi &:= \psi\,, & \varepsilon(x,y) &:= x = y\,, \\
\delta(x) &:= x = x\,, & \varphi_R(\bar{x}) &:= R\bar{x}\,.
\end{aligned}$$

□

For the application to boundedness below we will need the following interpretation results. First, let us consider classes of trees. The proof of the following lemma is straightforward. For (a) and (b), we use the usual first-child/next-sibling encoding of a tree, while for (c) we use a marking of the root, which can be used to recover the orientation of the edges since we can express reachability in MSO.

Lemma 10.7. *There exist* MSO*-interpretations mapping*

(1) *the class $\mathcal{T}_3$ of all ternary trees to the class $\mathcal{T}_{\aleph_0}$ of all countable trees;*
(2) *the class of all finite ternary trees to the class of all finite trees;*
(3) *the class of all undirected trees to the class of all directed trees.* □

Next, we study structures of bounded tree-width.

Definition 10.8. Let τ be a relational vocabulary and $\mathfrak{A}$ a τ-structure. A *tree-decomposition* of $\mathfrak{A}$ is a 2^A-labelled directed tree $D = (T, E, \lambda)$ satisfying the following conditions:

- $\bigcup_{t \in T} \lambda(t) = A$.
- For all relation symbols $R \in \tau$ of arity n and all tuples $(a_1, \dots, a_n) \in R^{\mathfrak{A}}$, there is some $t \in T$ such that $a_1, \dots, a_n \in \lambda(t)$.
- For all $a \in A$, the set $\{\, t \in T \mid a \in \lambda(t) \,\}$ is connected in (T, E).

The *width* of D is $\max_{t \in T} |\lambda(t)| - 1$. The *tree-width* of $\mathfrak{A}$ is the minimum width of a tree decomposition of $\mathfrak{A}$.

Lemma 10.9. *For every $k < \omega$ and all relational vocabularies τ, there exists an* MSO*-interpretation mapping the class of all trees to the class $\mathcal{W}_k[\tau]$ of all relational τ-structures of tree-width at most k, and the class of all finite trees to the class of all finite structures from $\mathcal{W}_k[\tau]$.*

Proof. For finite trees, such an interpretation was first given by Courcelle and Engelfriet [10], but using a slightly different notion of an interpretation. We give a detailed proof, since the precise version needed here does not appear in the literature.

We start by explaining how we can encode a structure $\mathfrak{A} \in \mathcal{W}_k[\tau]$ into a tree. Then we will construct an interpretation $\mathcal{I}$ performing the inverse translation. Let $\mathfrak{A}$ be a τ-structure of tree-width at most k and let $D = (T, E, \lambda)$ be a tree decomposition of $\mathfrak{A}$ of width at most k. It is no restriction to assume that no $\lambda(v)$ is empty and that there is some injective function $\iota : A \to T$ with $a \in \lambda(\iota(a))$, for all a. For every $v \in T$, we fix an enumeration $c_0^v, \dots, c_{\ell_v}^v$ of $\lambda(v)$ where $0 \leq \ell_v \leq k$ and $a = c_0^{\iota(a)}$ for all $a \in A$. We obtain a tree structure by turning (T, E) into an undirected tree expanded by the following monadic relations:

- a unary predicate R containing only the root;
- unary relations $E_{i,j}$, for $0 \leq i, j \leq k$, containing those $v \in T$ such that v has a parent u and $c_i^v = c_j^u$;
- unary predicates $P_{\mathfrak{C}}$, for every τ-structure $\mathfrak{C}$ with universe $C = \{0, \dots, \ell\}$, for some $0 \leq \ell \leq k$, where $P_{\mathfrak{C}}$ contains those $v \in T$ such that $\mathfrak{C}$ is isomorphic to the substructure of $\mathfrak{A}$ induced by $\lambda(v)$ via the isomorphism $i \mapsto c_i^v$.

The interpretation $\mathcal{I} = \langle \chi, \delta(x), \varepsilon(x, y), (\varphi_R(\bar{x}))_{R \in \tau} \rangle$ that reverses this encoding is defined as follows. The formula χ states that

- R is a singleton, unless the universe is empty;
- for all $0 \leq i, i', j, j' \leq k$ with $i \neq i'$ and $j \neq j'$, $E_{i,j}$ is disjoint from $E_{i,j'}$ and from $E_{i',j}$;
- the $P_{\mathfrak{C}}$ are disjoint;
- if $v \in P_{\mathfrak{C}}$ and $0 \leq i, j \leq k$ are such that $i \geq |C|$, then there is no $u \in T$ with $(v, u) \in E_{i,j}$ or $(u, v) \in E_{j,i}$.

For all $0 \leq i, j \leq k$, there is an MSO-formula $\varepsilon_{i,j}(x, y)$ which defines the set of pairs (v, u) such that $c_i^v = c_j^u$. We set

$$\delta(x) := x = x \quad \text{and} \quad \varepsilon := \varepsilon_{0,0}\,.$$

Finally, for each relation symbol $R \in \tau$ of arity r, we define

$$\varphi_R(x_1, \dots, x_r) := \bigvee_{0 \leq i_1 \leq k} \cdots \bigvee_{0 \leq i_r \leq k} \bigvee_{\substack{\mathfrak{C} \\ (i_1, \dots, i_r) \in R^{\mathfrak{C}}}} \exists y \Big[\bigwedge_{1 \leq j \leq r} \varepsilon_{0,i_j}(x_j, y) \wedge P_{\mathfrak{C}} y \Big]\,.$$

□

11. Decidability results for boundedness

Using the reduction techniques of the previous sections we obtain a wealth of decidability results. We start with $\mathrm{BDD}^1(\mathrm{MSO}, \mathcal{T})$ and $\mathrm{BDD}^1(\mathrm{MSO}, \mathcal{W}_k)$.

Proposition 11.1. *The monadic boundedness problem for* MSO *over the class of all trees is decidable.*

Proof. By Proposition 9.8, an MSO-formula is bounded over the class of all trees if, and only if, it is bounded over the class of all countable trees. Hence, it is sufficient to prove that $\mathrm{BDD}^1(\mathrm{MSO}, \mathcal{T}_{\aleph_0})$ is decidable. By Lemma 10.7, there exists an MSO-interpretation $\mathcal{I}$ mapping $\mathcal{T}_3$ to $\mathcal{T}_{\aleph_0}$. Hence, the decidability of $\mathrm{BDD}^1(\mathrm{MSO}, \mathcal{T}_{\aleph_0})$ follows by Proposition 10.5 and Theorem 7.6. □

Theorem 11.2. *For every $k < \omega$ and all relational vocabularies τ, $\mathrm{BDD}^1(\mathrm{MSO}, \mathcal{W}_k[\tau])$, the monadic boundedness problem for* MSO *over the class $\mathcal{W}_k[\tau]$ of all relational τ-structures of tree-width at most k is decidable.*

Proof. With the help of Lemma 10.9 and Proposition 10.5 we can reduce $\mathrm{BDD}^1(\mathrm{MSO}, \mathcal{W}_k[\tau])$ to $\mathrm{BDD}^1(\mathrm{MSO}, \mathcal{T})$. The latter is decidable by Proposition 11.1. $\square$

Corollary 11.3. $\mathrm{BDD}^1(\mathrm{FO}_+(\exists^*))$, $\mathrm{BDD}^1(\mathrm{FO}_-(\forall^*))$, *and* $\mathrm{BDD}^1(\mathrm{ML})$ *are decidable.*

Proof. For each of these logics, Observation 9.3 provides a transfer result to (finite) structures of bounded tree-width. $\square$

Using similar techniques as above, one can extend Theorem 11.2 to the extension of MSO by counting quantifiers, to guarded second-order logic GSO, and to simultaneous fixed points. Instead of replacing MSO by a stronger logic, one can also replace tree-width by clique-width.

We only give a sketch of the proof. Let us denote by MSO + C and $\mathrm{GSO}^* + \mathrm{C}$ the extension of the respective logic by predicates of the form $|X| < \aleph_0$ and $|X| \equiv k \pmod{m}$, where X is a second-order variable and $k, m < \omega$. A *simultaneous* fixed point is defined by a system of formulae $\varphi_0(\bar{X}, \bar{x}_0), \dots, \varphi_{n-1}(\bar{X}, \bar{x}_{n-1})$ with first-order variables $\bar{x}_i$ and n second-order variables $X_0, \dots, X_{n-1}$.

Theorem 11.4. *For every $k < \omega$, the boundedness problem for simultaneous* $(\mathrm{GSO}^* + \mathrm{C})$-*fixed points over the class of all relational structures of tree-width at most k is decidable.*

Sketch. Since structures of tree-width at most k are sparse, we can find, for every $(\mathrm{GSO}^* + \mathrm{C})$-formula, an equivalent $(\mathrm{MSO} + \mathrm{C})$-formula (see [9, 3]). Therefore, the boundedness problem reduces to the boundedness of simultaneous $(\mathrm{MSO} + \mathrm{C})$-fixed points on that class. Using the interpretation argument from above, we can reduce it further to the boundedness for simultaneous $(\mathrm{MSO} + \mathrm{C})$-fixed points on the class of all ternary trees. On ternary trees, MSO + C collapses to MSO. Therefore, we only need to decide boundedness for simultaneous MSO-fixed points. Finally, using again an interpretation argument we can replace a simultaneous fixed point by an ordinary one (by making several copies of each vertex of the tree, one for each component of the simultaneous fixed point). $\square$

Corollary 11.5. *The following problems are decidable:* $\mathrm{BDD}(\mathrm{GF})$, $\mathrm{BDD}(\mu\mathrm{GF})$, $\mathrm{BDD}^1(\mathrm{L}_\mu)$, $\mathrm{BDD}(\mathrm{GGSO}, \mathcal{W}_k)$, *and* $\mathrm{BDD}^1(\mathrm{GSO}, \mathcal{W}_k)$.

Proof. By Observation 9.3, GF and μGF have the bounded-tree-width property for BDD. Hence, BDD(GF) and $\mathrm{BDD}(\mu\mathrm{GF})$ reduce to $\mathrm{BDD}(\mathrm{GF}, \mathcal{W}_k)$ and $\mathrm{BDD}(\mu\mathrm{GF}, \mathcal{W}_k)$, respectively, which in turn are subsumed by $\mathrm{BDD}(\mathrm{GGSO}, \mathcal{W}_k)$. According to Proposition 10.2, $\mathrm{BDD}(\mathrm{GGSO}, \mathcal{W}_k)$ reduces to $\mathrm{BDD}(\mathrm{GGSO}^*, \mathcal{W}_k)$ which is decidable by Theorem 11.4.

For $\mathrm{BDD}^1(\mathrm{GSO}, \mathcal{W}_k)$ note that, singletons being always guarded, every GSO-formula $\varphi(X, x)$ with a single free first-order variable x belongs to GSO^*. Hence, $\mathrm{BDD}^1(\mathrm{GSO}, \mathcal{W}_k)$ reduces to $\mathrm{BDD}^1(\mathrm{GSO}^*, \mathcal{W}_k)$ and the claim follows again from Theorem 11.4. $\square$

Part III. Complexity Results

12. Complexity

In connection with our decision procedures we have not been specific about the algorithmic complexities involved. The fact that we have to deal with X-positive n-types as basic data has a major impact on all upper bounds that can be derived from our approach. Space $\exp^n(\Theta(|\tau|))$ is necessary to even store such a type ($\exp^n$ denotes the n-fold application of the exponentiation operation, that is, a tower of height n). Overall it is straightforward to check that, on input φ, our decision procedure runs in time $\exp^{\mathrm{qr}(\varphi)+O(1)}(|\varphi|)$.

We now provide a corresponding lower bound, even for monadic boundedness for first-order logic over just finite trees. Note that, for most natural fragments of MSO, one can obtain a lower bound from the complexity of the satisfiability problem of the fragment. For instance, $\mathrm{BDD}^1(\mathrm{ML})$ is PSPACE-hard since satisfiability for ML is PSPACE-complete. For first-order logic over finite words *with order,* as well as over finite trees *without order,* we can similarly derive lower bounds from corresponding bounds for the satisfiability problem.

Theorem 12.1. (a) *The boundedness problem* $\mathrm{BDD}^1(\mathrm{FO}, \mathcal{P})$ *for first-order logic over the class of finite words with order, is complete for* $\mathrm{DSPACE}(\exp^{\mathrm{poly}(n)}(1))$.

(b) *The boundedness problem* $\mathrm{BDD}^1(\mathrm{FO}, \mathcal{T}_{\mathrm{fin}})$ *for first-order logic over the class of all finite trees is hard for* $\mathrm{DTIME}(\exp^{\mathrm{poly}(n)}(1))$. □

Part (a) follows from the corresponding result for $\mathrm{SAT}(\mathrm{FO}, \mathcal{P})$; see [26] for a proof and exposition. Part (b) is a consequence of the following complexity bound for $\mathrm{SAT}(\mathrm{FO}, \mathcal{T}_{\mathrm{fin}})$; although this is based on standard techniques, we include a proof since this complexity bound does not seem to appear in the literature.

Proposition 12.2. $\mathrm{SAT}(\mathrm{FO}, \mathcal{T}_{\mathrm{fin}})$ *is hard for* $\mathrm{DTIME}(\exp^{\mathrm{poly}(n)}(1))$ *under polynomial time reductions.*

Proof. We show that $\mathrm{SAT}(\mathrm{FO}, \mathcal{T}_{\mathrm{fin}})$ is hard for $\mathrm{NTIME}(\exp^{\mathrm{poly}(n)}(1))$, which is the same as $\mathrm{DTIME}(\exp^{\mathrm{poly}(n)}(1))$. We use the following tiling problem, which is complete for $\mathrm{NTIME}(\exp^{\mathrm{poly}(n)}(1))$ (see [19] for an overview): given a set D of tiles, two relations $H, V \subseteq D \times D$, and a natural number n (in unary encoding), determine whether there exists a tiling of the $(\exp^n(1) \times \exp^n(1))$-grid, i.e., a function $\tau : \exp^n(1) \times \exp^n(1) \to D$ such that

$$(\tau(x,y), \tau(x+1,y)) \in H \quad \text{and} \quad (\tau(x,y), \tau(x,y+1)) \in V\,, \quad \text{for all } x, y\,.$$

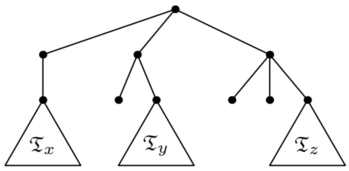
For the reduction, we set $N := \exp^n(1)$. One can show that the problem remains complete for $\mathrm{NTIME}(\exp^{\mathrm{poly}(n)}(1))$ even if we require for convenience that there are at most N tiles. Thus, we can represent tiles by numbers less than N. We construct a formula ψ that is satisfied by some finite tree if, and only if, there exists a tiling of the $(N \times N)$-grid.

We use an encoding of numbers by directed trees introduced in [13] (see also [11]) where numbers from $\{0, \dots, N-1\}$ are encoded by trees of height at most n. The encoding is such that there are first-order formulae $\varphi_N(x)$, $\varphi_{\min}(x)$, $\varphi_{\max}(x)$, $\varphi_=(x,y)$ and $\varphi_{\mathrm{suc}}(x,y)$, which can be constructed in time polynomial in n, with the property that

- a vertex v in a tree $\mathfrak{T}$ satisfies $\varphi_N(x)$ if the subtree rooted at v encodes a number from $\{0, \dots, N-1\}$;

- a vertex v satisfies $\varphi_{\min}(x)$ or $\varphi_{\max}(x)$ if the subtree rooted at v encodes the number 0 or $N-1$, respectively;
- the formulae $\varphi_=(x,y)$ and $\varphi_{\text{suc}}(x,y)$ similarly define equality and the successor relation for numbers encoded in the subtrees rooted at x and y.

We use this encoding to represent triples of numbers as follows. The triple (x,y,z) is encoded by a tree of the form



where $\mathfrak{T}_x$, $\mathfrak{T}_y$, and $\mathfrak{T}_z$ are the trees representing x, y, and z, respectively. Then, a tiling can by represented by a set of triples (x,y,z), where x and y are coordinates and z is the tile at position (x,y). The respective set of trees is turned into a single tree by making all these triples children of a new root.

To axiomatise the representation of a valid tiling, we use a formula ψ based on the formulae φ_N, $\varphi_{\min}$, $\varphi_{\max}$, φ_{suc}, and $\varphi_=$ from above. The formula ψ expresses the following:

- All children of the root encode triples of numbers.
- There is some triple (x,y,z) with $x=y=0$.
- Each triple has a neighbour to the right, unless the x-coordinate already is $N-1$. The tiles of the triple and its neighbour match.
- Similarly, there is a neighbour above.
- Each position occurs at most once.

Such a formula ψ is constructible in time polynomial in n and the size of the tile set. Clearly, directed tree models of ψ correspond to valid tilings of an $(N\times N)$-grid. Hence, ψ is satisfiable by such a tree if, and only if, such a tiling exists.

To work inside the class $\mathcal{T}_{\text{fin}}$, we need to replace the directed trees by undirected ones. Observe that every model of ψ is a tree of height at most $n+3$. Hence, we can uniquely mark the root by attaching a path of length $n+4$ to it. It is easy to modify ψ to work with such undirected trees instead. $\square$

Corollary 12.3. *The following boundedness problems are* $\text{DTIME}\big(\exp^{\text{poly}(n)}(1)\big)$*-complete:*

(1) $\text{BDD}^1(\text{FO},\mathcal{T}_{\text{fin}})$ *where* $\mathcal{T}_{\text{fin}}$ *is the class of all finite trees.*
(2) $\text{BDD}^1(\text{MSO},\mathcal{T})$ *where* $\mathcal{T}$ *is the class of all trees.*
(3) *Boundedness for simultaneous* (GSO^*+C)*-fixed points over the class of structures of bounded tree-width.*

Proof. (1) follows from Proposition 12.2. As (2) reduces to (3), for which we already have a trivial non-elementary upper bound, it is sufficient to provide a lower bound for $\text{BDD}^1(\text{MSO},\mathcal{T})$. As we have seen, $\text{BDD}^1(\text{FO},\mathcal{T}_{\text{fin}})$ reduces to $\text{BDD}^1(\text{MSO},\mathcal{T}_3)$, which in turn reduces to $\text{BDD}^1(\text{MSO},\mathcal{T})$. Hence, the lower bound follows from (1). $\square$

This lower bound shows that, for many cases, our algorithm is best possible. Of course, there are important fragments of MSO to which the lower bound is not applicable. For instance, the following upper bounds are known from the literature:

Theorem 12.4.

(1) $\mathrm{BDD}^1(\mathrm{ML})$ *is in* Exptime [24].
(2) $\mathrm{BDD}^1(\mathrm{FO}_+(\exists^*))$ *is in* 2-Exptime [8]. □

Since it is not the main concern of this article, we leave the exact complexity of $\mathrm{BDD}^1(\mathrm{ML})$, $\mathrm{BDD}^1(\mathrm{L}_\mu)$, $\mathrm{BDD}^1(\mathrm{FO}_+(\exists^*))$, $\mathrm{BDD}^1(\mathrm{FO}_-(\forall^*))$, BDD(GF), and BDD($\mu$GF) open.

References

[1] Hainal Andréka, Johan van Benthem, and Istvan Németi. Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic*, 27:217–274, 1998.

[2] Jon Barwise and Yiannis Moschovakis. Global inductive definability. *The Journal of Symbolic Logic*, 43(3):521–534, 1978.

[3] Achim Blumensath. Guarded Second-Order Logic, Spanning Trees, and Network Flows. *Logical Methods in Computer Science*, 6, 2010.

[4] Egon Börger, Erich Grädel, and Yuri Gurevich. *The Classical Decision Problem*. Springer-Verlag, 1997.

[5] Thomas Colcombet. Regular cost functions over words. unpublished, 2009.

[6] Thomas Colcombet. *Fonctions régulières de coût*. Habilitation Thesis, Université Paris 7, 2013.

[7] Thomas Colcombet and Christof Löding. The nesting-depth of disjunctive μ-calculus for tree languages and the limitedness problem. In *Proc. 17th EACSL Annual Conference on Computer Science Logic, CSL 2008*, 2008.

[8] Stavros Cosmadakis, Haim Gaifman, Paris Kanellakis, and Moshe Vardi. Decidable optimization problems for database logic programs. In *Proceedings 20th Annual ACM Symposium on Theory of Computing, STOC 1988*, pages 477–490, 1988.

[9] Bruno Courcelle. The monadic second-order logic of graphs XIV: Uniformly sparse graphs and edge set quantifications. *Theoretical Computer Science*, 299:1–36, 2003.

[10] Bruno Courcelle and Joost Engelfriet. A Logical Characterization of the Sets of Hypergraphs Defined by Hyperedge Replacement Grammars. *Math. System Theory*, 28:515–552, 1995.

[11] Anuj Dawar, Martin Grohe, Stephan Kreutzer, and Nicole Schweikardt. Model theory makes formulas large. In *ICALP 2007, Proc. 34th International Colloquium on Automata, Languages and Programming*, number 4596 in LNCS, pages 913–924, 2007.

[12] Heinz-Dieter Ebbinghaus and Jörg Flum. *Finite Model Theory*. Springer Verlag, 1995.

[13] Jörg Flum and Martin Grohe. *Parametrized Complexity Theory*. Springer Verlag, 2006.

[14] Haim Gaifman, Harry Mairson, Yehoshua Sagiv, and Moshe Vardi. Undecidable optimization problems for database logic problems. *Journal of the ACM*, 40:683–713, 1993.

[15] Valentin Goranko and Martin Otto. Model theory of modal logic. In Patrick Blackburn, Johann van Benthem, and Frank Wolter, editors, *Handbook of Modal Logic*, pages 249–329. Elsevier, 2007.

[16] Erich Grädel. On the restraining power of guards. *JSL*, 64:1719–1742, 1999.

[17] Erich Grädel, Colin Hirsch, and Martin Otto. Back and forth between guarded and modal logics. *ACM Transactions on Computational Logic*, 3:418–463, 2002.

[18] Erich Grädel and Igor Walukiewicz. Guarded Fixed Point Logic. In *Proceedings of 14th IEEE Symposium on Logic in Computer Science LICS '99, Trento*, pages 45–54, 1999.

[19] David Harel. Recurring Dominoes: Making the Highly Undecidable Highly Understandable. *Annals of Discrete Mathematics*, 24:51–72, 1985.

[20] Gerd Hillebrand, Paris Kanellakis, Harry Mairson, and Moshe Vardi. Undecidable boundedness problems for datalog programs. *The Journal of Logic Programming*, 25:163–190, 1995.

[21] David Janin and Igor Walukiewicz. On the expressive completeness of the propositional μ-calculus w.r.t. monadic second-order logic. In *Proceedings CONCUR '96*, pages 263–277, 1996.

[22] Phokion Kolaitis and Martin Otto. On the boundedness problem for fragments of first-order logic. unpublished draft.

[23] Stephan Kreutzer, Martin Otto, and Nicole Schweikardt. Boundedness of monadic FO over acyclic structures. In *ICALP 2007, Proc. 34th International Colloquium on Automata, Languages and Programming*, number 4596 in LNCS, pages 571–582, 2007.

[24] Martin Otto. Eliminating recursion in the μ-calculus. In *STACS 1999, Proc. 16th Annual Symposium on Theoretical Aspects of Computer Science*, number 1563 in LNCS, pages 531–540, 1999.

[25] Martin Otto. The boundedness problem for monadic universal first-order logic. In *Proc. 21th IEEE Symposium on Logic in Computer Science, LICS 2006*, pages 37–46, 2006.
[26] Klaus Reinhardt. The complexity of translating logic to finite automata. In E. Grädel, W. Thomas, and T. Wilke, editors, *Automata, Logic and Infinite Games*, LNCS 2500, pages 231–238. Springer Verlag, 2002.
[27] Johan van Benthem. *Modal Logic and Classical Logic*. Bibliopolis, 1983.
[28] Miachael Vanden Boom. *Weak Cost Automata over Infinite Trees.* Ph. D. Thesis, University Oxford, 2012.
[29] Igor Walukiewicz. Monadic second-order logic on tree-like structures. *Theoretical Computer Science*, 275(1–2):311–346, 2002.