Boundedness of monadic second-order formulae over finite words

A. Blumensath[1], M. Otto[1], and M. Weyer[2]

[1] Fachbereich Mathematik, Technische Universität Darmstadt
[2] Lehrstuhl Logik in der Informatik, Humboldt-Universität zu Berlin

Abstract. We prove that the boundedness problem for monadic second-order logic over the class of all finite words is decidable.

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. Fixed-point operators introduce a measure of relational recursion and typically boost expressiveness in the direction of more dynamic and less local properties, by iteration and recursion based on the expressiveness that is locally or statically available in the underlying fragment, say of first-order logic FO. We here exclusively consider monadic least fixed points, based on formulae $\varphi(X, x)$ that are monotone (positive) in the monadic recursion variable X. 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 least stationary point of the monotone iteration sequence of stages $\varphi^\alpha(\mathfrak{A})$ starting from $\varphi^0(\mathfrak{A}) := \emptyset$. 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 (a pure expressiveness issue); (b) one may also be interested to know whether the given fixed-point iteration terminates within a uniformly finitely bounded number of iterations (an algorithmic issue, concerning the dynamics of the fixed-point recursion rather than its result).

The boundedness problem $\mathsf{Bdd}(\mathcal{F}, \mathcal{C})$ for a class of formulae $\mathcal{F}$ and a class of structures $\mathcal{C}$ concerns question (b): to decide, for $\varphi \in \mathcal{F}$, 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 [1], 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, e.g., [6]. Notable exceptions are monadic boundedness for positive existential formulae (datalog) [3], for modal formulae [9], and for (a restricted class of) universal formulae without equality [10].

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 [9], 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 [10].

Motivated by this observation, [7] 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. [7] shows that boundedness is decidable for *all* monadic fixed points in FO over the class of all acyclic relational structures. Technically [7] expands on modal and locality based proof ideas and reductions to the MSO theory of trees from [9, 10] 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, and – maybe the most important novel feature – in 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 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, for the purposes of the present treatment, the underlying structures are restricted to just finite word structures. We expect that this restriction can be somewhat relaxed, though. Work in progress based on automata theoretic results recently obtained by Colcombet and Löding [2] shows that our approach generalises from finite words to the case of finite trees. This extension of the present results will, via MSO interpretability in trees, then reach up at least to the finite model-theory version of the following conjecture, which has been implicit as a potential keystone to this alternative approach to boundedness:

Conjecture 1. The boundedness problem for monadic second-order logic over the class of all trees (and hence over any MSO-definable class of finite tree width) is decidable.

2 Preliminaries

We assume that the reader is familiar with basic notions of logic (see, e.g., [4] for details). Throughout the paper we assume that all vocabularies are finite and that they contain only relation symbols and constant symbols, but no function symbols. We regard free variables as constant symbols.

Let φ and ψ be formulae over a vocabulary τ containing a unary relation symbol X and a constant symbol x. As usual, the formula $\varphi[c/x]$ is obtained from φ by replacing all free occurrences of x by c. The formula $\varphi[\psi(x)/X]$ is obtained from φ by replacing all free occurrences of X, say Xc with constant symbol c, by $\psi[c/x]$. For $\alpha < \omega$, we define the formula φ^α inductively as follows:

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

Note that the vocabulary of φ^α is $\tau \setminus \{X\}$. Suppose that $\mathfrak{A}$ is a structure of vocabulary $\tau \setminus \{X, x\}$. If φ is positive in X, then φ^α defines the α-th stage of the least fixed-point induction of φ on $\mathfrak{A}$. We denote this set by $\varphi^\alpha(\mathfrak{A})$. The corresponding fixed point is $\varphi^\infty(\mathfrak{A})$.

Definition 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 $\mathcal{C}$.*

Lemma 1. *Let L be a logic and $\mathcal{C}$ a class of structures such that equivalence of L-formulae over $\mathcal{C}$ is decidable. The boundedness problem for L over $\mathcal{C}$ is decidable if and only if there is a computable function $f : L \to \omega$ such that, if a formula $\varphi \in L$ is bounded over $\mathcal{C}$, then it is bounded by $f(\varphi)$ over $\mathcal{C}$.*

In this paper we consider the class of all finite words over some alphabet Σ. We encode such words as structures in the usual way. Let τ_Σ be the signature consisting of a binary relation $\leq$ and unary relations P_c, for every $c \in \Sigma$. We represent a finite word $w = a_0 \dots a_{n-1} \in \Sigma^*$ as the structure whose universe $[n] := \{0, \dots, n-1\}$ consists of all positions in w and where $\leq$ is interpreted by the usual order on integers, and P_c is interpreted by the set $\{ i \in [n] \mid a_i = c \}$ of all positions carrying the letter c.

We denote the *concatenation* of two words $\mathfrak{A}$ and $\mathfrak{B}$ by $\mathfrak{A} + \mathfrak{B}$. For a structure $\mathfrak{A}$ and a set $U \subseteq A$, we denote by $\mathfrak{A}_U$ the substructure induced by U. (If $\mathfrak{A}$ contains constants with value outside of U then we drop them from the vocabulary when forming $\mathfrak{A}_U$.)

We will reduce the boundedness problem to a corresponding problem for automata. A *distance automaton* is a tuple $\mathcal{A} = (\Sigma, Q, \Delta_0, \Delta_1, I, F)$, where $\mathcal{A}' = (\Sigma, Q, \Delta, I, F)$ for $\Delta = \Delta_0 \dot\cup \Delta_1$ is a finite nondeterministic automaton in the usual sense with alphabet Σ, state space Q, transition relation $\Delta \subseteq Q \times \Sigma \times Q$, set of initial states $I \subseteq Q$, and set of final states $F \subseteq Q$. The language $L(\mathcal{A})$ of $\mathcal{A}$ is the language of $\mathcal{A}'$ in the usual sense, and for $w \in L(\mathcal{A})$, the distance $d_{\mathcal{A}}(w)$ is the minimal number of transitions from Δ_1, the minimum ranging over all accepting runs of $\mathcal{A}'$ on w. For $w \notin L(\mathcal{A})$, we set $d_{\mathcal{A}}(w) := \infty$. As usual, we set $d_{\mathcal{A}}(L) := \{ d_{\mathcal{A}}(w) \mid w \in L \}$, for sets $L \subseteq \Sigma^*$. This definition is a slightly modified version of the one in [5].

Theorem 1 (Hashiguchi [5, 8]). *Let $\mathcal{A}$ be a distance automaton with state space Q. If $d_{\mathcal{A}}(L(\mathcal{A}))$ is bounded, then it is bounded by $2^{4|Q|^3}$:*

$$\sup d_{\mathcal{A}}(L(\mathcal{A})) < \infty \quad \textit{implies} \quad \sup d_{\mathcal{A}}(L(\mathcal{A})) \leq 2^{4|Q|^3} .$$

3 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. 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. $\mathrm{MSO}^n_X[\tau]$ is finite up to logical equivalence, and we will silently assume that all formulae are canonised in some way. For example, for $\Phi \subseteq \mathrm{MSO}^n_X[\tau]$ the conjunction $\bigwedge \Phi$ is always a formula from $\mathrm{MSO}^n_X[\tau]$, and it will even happen that $\bigwedge \Phi \in \Phi$. The following result carries over from $\mathrm{MSO}^n[\tau]$ to $\mathrm{MSO}^n_X[\tau]$.

Fact 1. *There exists a computable function $f : \omega \to \omega$ such that, up to logical equivalence, we have*

$$|\mathrm{MSO}^n[\tau]| \leq f\big(n + |\tau| + \mathrm{ar}(\tau)\big) .$$

Definition 2. *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.

Lemma 2. *Let $\mathfrak{A}$ be a structure and $P \subseteq P' \subseteq A$. Then*

$$\mathrm{tp}^n_X(\mathfrak{A}, P) \subseteq \mathrm{tp}^n_X(\mathfrak{A}, P') ,$$

where X is interpreted by P and P', respectively.

Fact 2. *Let* $\mathfrak{B}$ *be the* τ*-reduct of* $\mathfrak{A}$*. Then* $\mathrm{tp}^n_X(\mathfrak{B}) = \mathrm{tp}^n_X(\mathfrak{A}) \cap \mathrm{MSO}^n_X[\tau]$.

Lemma 3. *For every* $n < \omega$*, there is a binary function* $\oplus_n$ *such that*

$$\mathrm{tp}^n_X(\mathfrak{A} + \mathfrak{B}) = \mathrm{tp}^n_X(\mathfrak{A}) \oplus_n \mathrm{tp}^n_X(\mathfrak{B})\,, \quad \text{for all words } \mathfrak{A} \text{ and } \mathfrak{B}\,.$$

Furthermore, $\oplus_n$ *is monotone:*

$$s \subseteq s' \quad \text{and} \quad t \subseteq t' \quad \text{implies} \quad s \oplus_n t \subseteq s' \oplus_n t'\,.$$

Note that, being a homomorphic image of word concatenation $+$, the operation $\oplus_n$ is associative.

4 The main theorem

Let us temporarily fix a formula $\varphi \in \mathrm{MSO}^n_X[\tau]$ with vocabulary $\tau = \{x, X, \leq, P_a, P_b, \dots\}$ belonging to a word structure with one constant symbol x and one additional unary predicate X. Let

$$\pi : \mathrm{Tp}^n_X[\tau] \to \mathrm{Tp}^n_X[\tau \setminus \{x\}]$$

be the canonical projection defined by $\pi(t) := t \cap \mathrm{MSO}^n_X[\tau \setminus \{x\}]$.

Note that, by our assumption on τ, there are exactly two X-positive n-types of one-letter τ-words with letter a: the one not containing Xx and the one which does contain Xx. We will at times denote these by, respectively, 0_a and 1_a. Frequently, we will omit the index a if we do not want to specify the letter.

Given a word structure $\mathfrak{A}$ of vocabulary $\tau \setminus \{X, x\}$, we consider the fixed-point induction of φ. For every $\alpha < \omega$ and every position p of $\mathfrak{A}$ we consider the type $\mathrm{tp}(\mathfrak{A}, \varphi^\alpha(\mathfrak{A}), p)$. We annotate $\mathfrak{A}$ with all these types. At each position p we write down the list of these types for all stages α. These annotations can be used to determine the fixed-point rank of all elements of $\mathfrak{A}$. A position p 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 position p each change between two consecutive layers is caused by some change at some other position 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 distance automaton that recognises (approximations of) such annotations. Furthermore, the distance computed by the automaton coincides with the longest path of changes in the annotation. It follows that the automaton is bounded if and only if the fixed-point induction is bounded. Consequently, we can solve the boundedness problem for φ with the help of Theorem 1.

Let us start by precisely defining the annotations we use. A local stage annotation at a position above a fixed letter of $\mathfrak{A}$ captures the flow of information that is relevant for stage updates in the fixed-point induction at this letter and at some stage.

Definition 3. (a) *A* local stage annotation *is a* 6*-tuple*

$$\gamma = \begin{pmatrix} {}^{<}t \; t^{\wedge} \; t^{>} \\ {}_{>}t \; {}_{\wedge}t \; t_{<} \end{pmatrix}$$

of types where

- ${}_{\wedge}t,\ t^{\wedge} \in \{0_a, 1_a\}$ *with* ${}_{\wedge}t \subseteq t^{\wedge}$*, for some letter* $a \in \Sigma$*,*
- ${}_{>}t,\ {}^{<}t,\ t^{>},\ t_{<} \in \mathrm{Tp}^n_X[\tau \setminus \{x\}]$,
- ${}^{<}t = \pi({}_{\wedge}t \oplus t_{<})$ *and* $t^{>} = \pi({}_{>}t \oplus {}_{\wedge}t)$,
- $Xx \in t^{\wedge}$ *iff* $\varphi \in {}_{>}t \oplus {}_{\wedge}t \oplus t_{<}$.

We say that γ *is an annotation of* a*, for the letter* a *in the first clause.*

(b) *Let* $\mathfrak{A}$ *be the word structure corresponding to* $a_0 \dots a_{\ell-1} \in \Sigma^*$*. For* $\alpha < \omega$*, we denote the expansion of* $\mathfrak{A}$ *by the* α*-th stage of* φ *by* $\mathfrak{A}^\alpha := (\mathfrak{A}, \varphi^\alpha(\mathfrak{A}))$*.*

The annotated word $\mathrm{An}(\mathfrak{A})$ *is a word* $b_0 \dots b_{\ell-1}$ *where the* p*-th letter* b_p *is the sequence of local stage annotations of* a_p *obtained by the removal of duplicates from the sequence* $(\gamma^\alpha)_{\alpha<\omega}$ *with*

$$\gamma^\alpha := \begin{pmatrix} \mathrm{tp}^n_X(\mathfrak{A}^\alpha_{[p,\ell)}) \; \mathrm{tp}^n_X(\mathfrak{A}^{\alpha+1}_{\{p\}}, p) \; \mathrm{tp}^n_X(\mathfrak{A}^\alpha_{[0,p]}) \\ \mathrm{tp}^n_X(\mathfrak{A}^\alpha_{[0,p)}) \; \mathrm{tp}^n_X(\mathfrak{A}^\alpha_{\{p\}}, p) \; \mathrm{tp}^n_X(\mathfrak{A}^\alpha_{(p,\ell)}) \end{pmatrix} .$$

Here, $\mathfrak{A}^\alpha_U$ *denotes* $(\mathfrak{A}^\alpha)_U$*, not* $(\mathfrak{A}_U)^\alpha$*.*

The components of an annotation γ are called *incoming from the left, outgoing to the left,* and so on. They are denoted by ${}_{>}\gamma, {}^{<}\gamma, \dots$. We also speak of the ${}_{>}\bullet$-component of γ, etc.

Example 1. Consider the formula

$$\varphi(X, x) := \forall y[y < x \rightarrow Xy] \vee \forall y[y > x \rightarrow Xy] .$$

Figure 1 shows (the first 4 elements of) the real annotation of a word of length at least 9. Here,

- λ denotes the type of the empty word,
- 0 denotes any type not containing the formula $\exists y Xy$,
- 1 denotes any type containing the formula $\forall y Xy$, and
- 01 denotes any type containing $\exists y Xy$, but not $\forall y Xy$.

Below we will construct an automaton that, given a word $\mathfrak{A}$ guesses potential annotations for $\mathfrak{A}$ and computes bounds on the length of the fixed-point induction of φ on $\mathfrak{A}$. Unfortunately, the real annotations $\mathrm{An}(\mathfrak{A})$ cannot be recognised by automata. For instance, in the above example the real annotations of words of even length are of the form ux^ny^nv where y^nv is the 'mirror image' of ux^n. This language is not regular.

So we have to work with approximations. Let us see what such approximations look like.

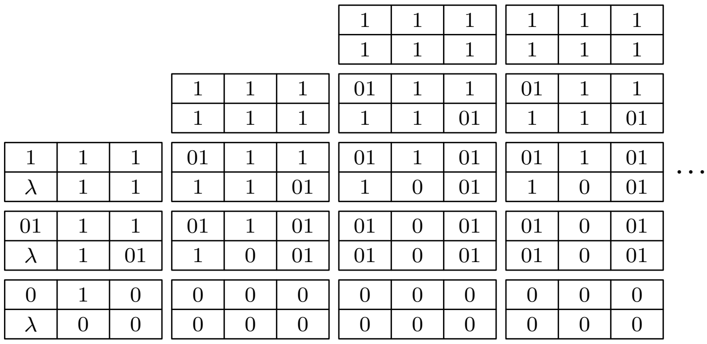


Fig. 1. Annotation for $\varphi(X, x) := \forall y[y < x \rightarrow Xy] \vee \forall y[y > x \rightarrow Xy]$

Definition 4. (a) *We extend the order $\subseteq$ on X-positive n-types to local stage annotations by requiring that $\subseteq$ holds component-wise. A* history *(at a) is a strictly increasing sequence $h = (h^0 \subsetneq \cdots \subsetneq h^m)$ of local stage annotations (at a) such that*

- ${}_\wedge(h^0) = 0_a$,
- ${}_\wedge(h^{i+1}) = (h^i)^\wedge$, *for all $i < m$, and*
- $(h^m)^\wedge = 1_a$ *implies* ${}_\wedge(h^m) = 1_a$.

Let Σ_τ denote the set of all histories with $a \in \Sigma$. An annotated word *is a word over Σ_τ.*

(b) *We say that an annotated word is* consistent, *if it satisfies the following conditions.*

(1) *If h_2 is the immediate successor of h_1, then the projections of h_1 to the components $\bullet^>$ and $\bullet_<$ coincide*[3] *with the projections of h_2 to the components ${}_>\bullet$ and ${}^<\bullet$, respectively.*
(2) *For the first letter: the ${}_>\bullet$ components in its history are all equal to $\mathrm{tp}^n_X(\lambda)$, where λ is the empty word.*
(3) *Similarly, for the last letter: the $\bullet_<$ components in its history are all equal to $\mathrm{tp}^n_X(\lambda)$.*

Clearly, $\mathrm{An}(\mathfrak{A})$ is a consistent annotated word. Furthermore, consistency of annotated words can be checked by an automaton since all conditions are strictly local. The main part of our work will consist in computing bounds on the real fixed-point rank of an element from such a word.

For an annotated word $\mathfrak{A}$, we index the individual type annotations by triples (p, i, j) where p is a position in $\mathfrak{A}$, i is an index for the history at position p, and j specifies the component in the local stage annotation. We denote the type specified in this way by $t_{p,i,j}$, or by $({}_>t)_{p,i}$, $({}^<t)_{p,i}$ for a concrete component $j = {}_>\bullet, {}^<\bullet$, etc.

[3] This is coincidence as a set, i.e., with duplicates removed.

When considering the annotated word encoding the fixed-point induction of φ, the indices of particular interest are those at which type changes occur. The fixed point is reached as soon as no such changes occur anymore.

Definition 5. *An index $I = (p, i, j)$ is* relevant *if either*

$$i > 0 \text{ and } t_{p,i,j} \neq t_{p,i-1,j}\,, \quad \text{or} \quad i = 0\,,\ j = \bullet^{\wedge}\,,\ \text{and } Xx \in t_{p,i,j}\,.$$

In the latter case, we call I initially relevant.

During the fixed-point induction changes at one index trigger changes at other indices in the next stage. The following definition formalises this dependency. We introduce three notions of dependency between indices. We have *direct dependencies,* where a change at one index immediately leads to a change at another one, and we have what we call *lower* and *upper dependencies,* intuitively associated with the temporal sequence of events. However, due to the lack of synchronisation between levels of adjacent histories (which in turn comes from the deletion of duplicates in each history), this temporal intuition is not directly available for dependencies linking adjacent histories. Some of the real stage dependencies can only be reconstructed globally, which will eventually give us the required bounds on ranks.

Definition 6. *Let $I = (p, i, j)$ and $I' = (p', i', j')$ be two relevant indices. We say that I* directly depends *on I' if I is not initially relevant and one of the following cases occurs:*

$$\begin{array}{llll}
j = {}_{>}\bullet\,, & j' = \bullet^{>}\,, & p' = p - 1\,, & \text{and} \quad t_I = t_{I'}\,;\\
j = \bullet_{<}\,, & j' = {}^{<}\bullet\,, & p' = p + 1\,, & \text{and} \quad t_I = t_{I'}\,;\\
j = {}_{\wedge}\bullet\,, & j' = \bullet^{\wedge}\,, & p' = p\,, & \text{and} \quad i' = i - 1\,;\\
j = \bullet^{\wedge}\,, & j' \in \{{}_{>}\bullet, {}_{\wedge}\bullet, \bullet_{<}\}\,, & p' = p\,, & \text{and} \quad i' = i\,;\\
j = {}^{<}\bullet\,, & j' \in \{\bullet_{<}, {}_{\wedge}\bullet\}\,, & p' = p\,, & \text{and} \quad i' = i\,;\\
j = \bullet^{>}\,, & j' \in \{{}_{>}\bullet, {}_{\wedge}\bullet\}\,, & p' = p\,, & \text{and} \quad i' = i\,.
\end{array}$$

A direct dependency of some index $(p, i, {}_{\wedge}\bullet)$ on $(p, i-1, \bullet^{\wedge})$ is called a jump.

Relaxing the equality requirement $i' = i$ to either $i' \leq i$ or to $i' \geq i$ in each of the last three clauses (thus also allowing upward or downward steps within the same history in those cases), we obtain dependencies from below *or* from above.

Note that the last three forms of direct dependencies go from outgoing to incoming indices within the same local annotation.

Furthermore, I directly depends on I' if and only if it depends on I' both from below and from above.

Also note that the first two clauses of (direct) dependency are the only dependencies between distinct (namely adjacent) histories. In these there is no condition on i, i', corresponding to the lack of synchronisation discussed above.

Finally note that in the case of a jump, i.e., a (direct) dependency of $(p, i, {}_{\wedge}\bullet)$ on $(p, i-1, \bullet^{\wedge})$, we have

$$({}_{\wedge}t)_{p,i} = (t^{\wedge})_{p,i-1} = 1 \quad \text{and} \quad ({}_{\wedge}t)_{p,i-1} = 0\,.$$

In particular, at every position p there can be at most one jump.

Lemma 4. *Let I be a relevant index in a consistent annotated word. Either I is initially relevant, or there is some relevant index on which I depends directly.*

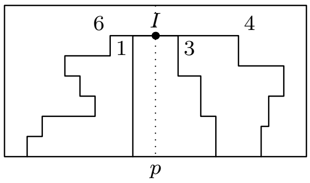
We can form a digraph consisting of all relevant indices where there is an edge from I to I' if I depends on I' from below. We call this digraph the *lower dependency graph*. Similarly, we can define the *upper dependency graph* by using dependencies from above.

Lemma 5. *The digraph of lower dependencies in a consistent annotated word is acyclic.*

A path in the dependency digraph is called *grounded* if it ends in an initially relevant index. The *rank* of a path is the number of jumps it contains.

Due to acyclicity and finiteness, all maximal paths in the lower dependency graph are grounded. The same is true also in the direct dependency graph, since every relevant index is either initial or it directly depends on some other relevant index by Lemma 4. For the upper dependency graph, which may have cycles, we can only say that every maximal cycle-free path must be grounded (and there are always such, namely in particular paths w.r.t. direct dependencies).

Let I be a relevant index and $\alpha < \omega$. We say that α is *a lower rank* of I if in the lower dependency graph there is some grounded path from I of rank α. Similarly, we define *upper ranks* of I as the ranks of grounded cycle-free paths in the upper dependency graph. Note that I can have several different lower and upper ranks, but at least one of each kind (due to the existence of grounded paths w.r.t. direct dependencies, Lemma 4).



We now fix a consistent annotated word $\mathfrak{A}$ over the underlying Σ-word $\mathfrak{B}$. Let ℓ be their length. As above, we write $\mathfrak{B}^\alpha := (\mathfrak{B}, \varphi^\alpha(\mathfrak{B}))$ for the expansion of $\mathfrak{B}$ by the α-th stage of the fixed-point induction.

We say that α *satisfies* an outgoing index $I = (p, i, j)$ if

$$\begin{aligned} & j = {}^{<}\bullet \quad \text{and} \quad \mathfrak{B}^\alpha_{[p,\ell)} \models t_I\,, \\ \text{or} \quad & j = \bullet^{\wedge} \quad \text{and} \quad (\mathfrak{B}^{\alpha+1}_{\{p\}}, p) \models t_I\,, \\ \text{or} \quad & j = \bullet^{>} \quad \text{and} \quad \mathfrak{B}^\alpha_{[0,p]} \models t_I\,. \end{aligned}$$

Note that $\mathfrak{B}^\alpha_{[p,\ell)} \models t_I$ just means that

$$t_I \subseteq \mathrm{tp}^n_X(\mathfrak{B}^\alpha_{[p,\ell)})\,.$$

Reverting the inclusion, we say that $I = (p, i, {}^{<}\bullet)$ *confines* α if

$$\mathrm{tp}^n_X(\mathfrak{B}^\alpha_{[p,\ell)}) \subseteq t_I \,.$$

For the other outgoing cases, we define confinement analogously.

The next lemma relates the real fixed-point induction of φ on $\mathfrak{B}$ to the given annotation $\mathfrak{A}$, through confinement. In particular, the top level of the annotation confines all stages of the real fixed point.

Lemma 6. *Let p be a position and m_p the length of the history at position p.*

(a) $\mathrm{tp}^n_X(\mathfrak{B}^0_{[0,p)}) = ({}_{>}t)_{p,0}$ *and* $\mathrm{tp}^n_X(\mathfrak{B}^0_{(p,\ell)}) = (t_{<})_{p,0}$.
(b) *For every $\alpha < \omega$, we have*

$$\begin{aligned}\mathrm{tp}^n_X(\mathfrak{B}^\alpha_{[0,p)}) &\subseteq ({}_{>}t)_{p,m_p}\,,\\ \mathrm{tp}^n_X(\mathfrak{B}^\alpha_{\{p\}}, p) &\subseteq ({}_{\wedge}t)_{p,m_p}\,,\\ \mathrm{tp}^n_X(\mathfrak{B}^\alpha_{(p,\ell)}) &\subseteq (t_{<})_{p,m_p}\,.\end{aligned}$$

We can use the preceding lemma to show that the lower and upper ranks provide bounds for the real rank of an element.

Lemma 7. *Let $I = (p, i, j)$ be a relevant outgoing index and $\alpha < \omega$.*

(a) *If $\alpha \geq \alpha'$ for all lower ranks α' of I, then α satisfies I.*
(b) *If I is not initially relevant and all upper ranks of I are larger than α, then $(p, i-1, j)$ confines α.*

We call a position p *active* if there is some i such that $(p, i, \bullet^{\wedge})$ is relevant; in this case, the corresponding i is unique. We may thus define the set of *upper ranks* of an active position p as the set of upper ranks of the relevant index of the form $(p, i, \bullet^{\wedge})$ at p. Recall that an upper rank of a relevant index is any rank of a grounded cycle-free upper dependency path.

Lemma 8. *Let p be a position.*

(a) *If $p \in \varphi^\infty(\mathfrak{B})$, then p is active.*
(b) *If p is active, then $p \in \varphi^\infty(\mathfrak{B})$.*
(c) *If $p \in \varphi^\alpha(\mathfrak{B})$ (and hence p is active by* (a)*), then some upper rank of p is at most α.*

A *proposal* is a pair $(\mathfrak{A}, p)$ where $\mathfrak{A}$ is a consistent annotated word and p is an active position in $\mathfrak{A}$. In order to treat proposals as words over some alphabet one can extend annotated letters with a mark for the special position p.

Lemma 9. *There exists a computable function $g : \omega \to \omega$ such that, for every formula φ, we can effectively construct a distance automaton $\mathcal{A}$ with at most $g(|\varphi|)$ states such that*

(a) *$L(\mathcal{A})$ is the set of proposals;*

(b) *if* $(\mathfrak{A}, p)$ *is a proposal then* $d_{\mathcal{A}}(\mathfrak{A}, p)$ *is the minimum over all upper ranks of* p.

Next, let us consider annotated words that arise from the actual fixed-point induction. Recall that $\mathrm{An}(\mathfrak{B})$ is the word whose p-th letter is the history (h^i) at position p. The removal of duplicates in the definition of a history induces a non-decreasing mapping $i_{\mathfrak{B},p} : \omega \to \omega$ from stages to history entries such that, for example, ${}_{>}(h^{i_{\mathfrak{B},p}(\alpha)}) = \mathrm{tp}_X^\alpha(\mathfrak{B}_{[0,p)})$. For $I = (p, i, j)$, we set

$$\alpha_{\mathfrak{B}}(I) := \alpha_{\mathfrak{B},p}(i) := \min \{ \alpha < \omega \mid i = i_{\mathfrak{B},p}(\alpha) \} .$$

Lemma 10. *Let* $\mathfrak{B}$ *be a word and let* I *be a relevant index in* $\mathrm{An}(\mathfrak{B})$*. Then each upper rank of* I *is bounded from below by* $\alpha_{\mathfrak{B}}(I)$*.*

Theorem 2. *The boundedness problem for* MSO *over the class of all finite words is decidable.*

Proof. Let $\varphi \in \mathrm{MSO}$ be positive in X and let g be the function from Lemma 9. We exploit Lemma 1 and claim that, over finite words, if φ is bounded then it is bounded by $N := 2^{4g(|\varphi|)^3} + 1$.

Assume that φ is bounded over finite words, say by N'. For every proposal $(\mathfrak{A}, p)$, it follows from Lemma 8 (b), that $p \in \varphi^\infty(\mathfrak{A})$. Hence, $p \in \varphi^{N'}(\mathfrak{A})$. Lemma 8 (c) then implies that some rank of p is at most N'. Let $\mathcal{A}$ be the distance automaton from Lemma 9. Then we have $d_{\mathcal{A}}(L(\mathcal{A})) \le N' < \infty$. Therefore, Theorem 1 implies that $d_{\mathcal{A}}(L(\mathcal{A})) \le N - 1$. Consequently, for all proposals $(\mathfrak{A}, p)$, some rank of p is at most $N - 1$. In particular, this holds if $\mathfrak{A} = \mathrm{An}(\mathfrak{B})$, for some word structure $\mathfrak{B}$. By Lemma 10, p enters the fixed-point induction not later than stage N. As $\mathfrak{B}$ and p were arbitrary, it follows that φ is bounded over words by N. ☐

5 Extensions

Having obtained the decidability for the boundedness problem over the class of all finite words we can use model theoretic interpretations to obtain further decidability results.

Theorem 3. *For all* k*, the boundedness problem for* MSO *over the class of all finite structures of path width at most* k *is decidable.*

Example 2. Let C_n be the class of all unranked trees (T, E, S) of height at most n where E is the successor relation and S is the next sibling relation. This class has path width at most $2n$. By the theorem, it follows that the boundedness problem for monadic second-order formulae over C_n is decidable.

Using similar techniques, one can extend the theorem to MSO-axiomatisable subclasses, to guarded second-order logic GSO, and to simultaneous fixed points. If we could show that the boundedness problem is also decidable for the class of all (finite) trees, then it would follow in the same way that the problem is decidable for every GSO-axiomatisable class of (finite) structures of bounded tree width.

References

1. J. Barwise and Y. N. Moschovakis, *Global inductive definability*, The Journal of Symbolic Logic, 43 (1978), pp. 521–534.
2. T. Colcombet and C. 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.
3. S. S. Cosmadakis, H. Gaifman, P. C. Kanellakis, and M. Y. Vardi, *Decidable optimization problems for database logic programs*, in Proc. 20th Annual ACM Symposium on Theory of Computing, STOC 1988, 1988, pp. 477–490.
4. H.-D. Ebbinghaus and J. Flum, *Finite Model Theory*, Springer Verlag, 1995.
5. K. Hashiguchi, *Improved limitedness theorems on finite automata with distance functions*, Theoretical Computer Science, 72 (1990), pp. 72–38.
6. G. Hillebrand, P. Kanellakis, H. Mairson, and M. Vardi, *Undecidable boundedness problems for datalog programs*, The Journal of Logic Programming, 25 (1995), pp. 163–190.
7. S. Kreutzer, M. Otto, and N. Schweikardt, *Boundedness of monadic FO over acyclic structures*, in Automata, Languages and Programming, Proc. 34th Int. Colloquium, ICALP 2007, no. 4596 in LNCS, 2007, pp. 571–582.
8. H. Leung and V. Podolskiy, *The limitedness problem on distance automata: Hashiguchi's method revisited*, Theoretical Computer Science, 310 (2004), pp. 147–158.
9. M. Otto, *Eliminating recursion in the μ-calculus*, in STACS 1999, Proc. 16th Annual Symposium on Theoretical Aspects of Computer Science, no. 1563 in LNCS, 1999, pp. 531–540.
10. ———, *The boundedness problem for monadic universal first-order logic*, in Proc. 21th IEEE Symposium on Logic in Computer Science, LICS 2006, 2006, pp. 37–46.

A Appendix

Proof (Proof of Lemma 1). (⇒) We need to show how to compute a bound $f(\varphi)$ from φ, under the condition that we know that φ is bounded over $\mathcal{C}$. This is done by simply checking, for $\alpha = 0, 1, \dots$, whether φ^α and $\varphi^{\alpha+1}$ are equivalent over $\mathcal{C}$. This process must stop after finitely many steps with the desired bound $f(\varphi)$.

(⇐) We can check boundedness by checking whether $\varphi^{f(\varphi)}$ and $\varphi^{f(\varphi)+1}$ are equivalent over $\mathcal{C}$. □

Proof (Proof of Lemma 2). Positive formulae are monotone, that is, for all $\varphi \in \mathrm{MSO}^n_X[\tau]$, we have that $(\mathfrak{A}, P) \models \varphi$ implies $(\mathfrak{A}, P') \models \varphi$. □

Proof (Proof of Lemma 3). We proceed by induction on n, simultaneously for all vocabularies.

For $n = 0$, we can write each formula $\varphi \in \mathrm{MSO}^0_X[\tau]$ as a positive boolean combination of literals. To compute $\mathrm{tp}^0_X(\mathfrak{A} + \mathfrak{B})$ it is therefore sufficient to determine which literals it contains. Most of these are already determined by $\mathrm{tp}^0_X(\mathfrak{A})$ or $\mathrm{tp}^0_X(\mathfrak{B})$. The only exceptions are atoms of the form $c = d$, $d = c$, $c \leq d$, and $d \leq c$ (and their negations) where $c \in A$ and $d \in B$. In these cases we always have

$$c \neq d,\ d \neq c,\ c \leq d,\ \neg(d \leq c) \in \mathrm{tp}^0_X(\mathfrak{A} + \mathfrak{B})\,.$$

For monotonicity, consider structures $\mathfrak{A}'$ and $\mathfrak{B}'$ with $\mathrm{tp}^n_X(\mathfrak{A}) \subseteq \mathrm{tp}^n_X(\mathfrak{A}')$ and $\mathrm{tp}^n_X(\mathfrak{B}) \subseteq \mathrm{tp}^n_X(\mathfrak{B}')$. We have to show that $\varphi \in \mathrm{tp}^n_X(\mathfrak{A} + \mathfrak{B})$ implies $\varphi \in \mathrm{tp}^n_X(\mathfrak{A}' + \mathfrak{B}')$. Since the claim is preserved under positive boolean combinations of formulae φ, we can assume that φ is a literal. Literals formed of the four exceptional atoms above are included or excluded without regard to the structures involved. By symmetry, we may therefore assume that φ is a literal that is determined by $\mathfrak{A}$. In this case we have

$$\begin{aligned} \varphi \in \mathrm{tp}^n_X(\mathfrak{A} + \mathfrak{B}) \quad &\text{iff} \quad \varphi \in \mathrm{tp}^n_X(\mathfrak{A}) \\ \text{and}\, \varphi \in \mathrm{tp}^n_X(\mathfrak{A}' + \mathfrak{B}') \quad &\text{iff} \quad \varphi \in \mathrm{tp}^n_X(\mathfrak{A}')\,. \end{aligned}$$

Since $\mathrm{tp}^n_X(\mathfrak{A}) \subseteq \mathrm{tp}^n_X(\mathfrak{A}')$ the claim follows.

For the inductive step, let $\varphi \in \mathrm{MSO}^{n+1}_X[\tau]$. Then φ is a positive boolean combination of formulae of quantifier rank at most n, and of formulae of the form $\exists y\psi$, $\forall y\psi$, $\exists Y\psi$, and $\forall Y\psi$ with $\psi \in \mathrm{MSO}^n_X[\tau]$. As above, it suffices to define $\oplus_{n+1}$ and to prove its monotonicity for such basic formulae. For formulae in $\mathrm{MSO}^n_X[\tau]$, we can apply the induction hypothesis. Hence, we only need to consider basic formulae of quantifier rank $n+1$. Such formulae talk about expansions of $\mathfrak{A} + \mathfrak{B}$ by either a new constant $c \in A \cup B$, or by a unary predicate $C \subseteq A \cup B$. For the first case, note that

$$(\mathfrak{A} + \mathfrak{B}, c) = \begin{cases} (\mathfrak{A}, c) + \mathfrak{B} & \text{if } c \in A\,, \\ \mathfrak{A} + (\mathfrak{B}, c) & \text{if } c \in B\,. \end{cases}$$

In the second case we have

$$(\mathfrak{A}+\mathfrak{B}, C) = (\mathfrak{A}, C\cap A) + (\mathfrak{B}, C \cap B)\,.$$

For $\mathfrak{D} \in \{\mathfrak{A}, \mathfrak{B}\}$, let us set

$$T_1(\mathfrak{D}) := \{\, \mathrm{tp}^n_X(\mathfrak{D}, c) \mid c \in D\,\}\,,$$
$$T_2(\mathfrak{D}) := \{\, \mathrm{tp}^n_X(\mathfrak{D}, C) \mid C \subseteq D\,\}$$

Let us start with the formula $\exists y\psi$. By induction hypothesis, we have

$$\begin{aligned}
&\exists y\psi \in \mathrm{tp}^{n+1}_X(\mathfrak{A}+\mathfrak{B})\\
\text{iff}\quad &\psi \in \mathrm{tp}^n_X((\mathfrak{A},c)+\mathfrak{B})\,, \quad \text{for some } c \in A\,,\\
\text{or }\ &\psi \in \mathrm{tp}^n_X(\mathfrak{A}+(\mathfrak{B},c))\,, \quad \text{for some } c \in B\,,\\
\text{iff}\quad &\psi \in \mathrm{tp}^n_X(\mathfrak{A},c) \oplus_n \mathrm{tp}^n_X(\mathfrak{B})\,, \quad \text{for some } c \in A\,,\\
\text{or }\ &\psi \in \mathrm{tp}^n_X(\mathfrak{A}) \oplus_n \mathrm{tp}^n_X(\mathfrak{B},c)\,, \quad \text{for some } c \in B\,,\\
\text{iff}\quad &\psi \in t \oplus_n \mathrm{tp}^n_X(\mathfrak{B})\,, \quad \text{for some } t \in T_1(\mathfrak{A})\,,\\
\text{or }\ &\psi \in \mathrm{tp}^n_X(\mathfrak{A}) \oplus_n t\,, \quad \text{for some } t \in T_1(\mathfrak{B})\,.
\end{aligned}$$

We claim that

(1) $\psi \in t \oplus_n \mathrm{tp}^n_X(\mathfrak{B})$, for some $t \in T_1(\mathfrak{A})$,

is equivalent to

(2) $\psi \in t \oplus_n \mathrm{tp}^n_X(\mathfrak{B})$, for some $t \in T'_1(\mathfrak{A})$, where

$$T'_1(\mathfrak{A}) := \bigl\{\, t \in \mathrm{Tp}^n_X[\tau \cup \{y\}] \bigm| \exists y \textstyle\bigwedge t \in \mathrm{tp}^{n+1}_X(\mathfrak{A})\,\bigr\}\,.$$

Clearly $T_1(\mathfrak{A}) \subseteq T'_1(\mathfrak{A})$ and (1) $\Rightarrow$ (2). But note that the inclusion may be strict, as $t \in T'_1(\mathfrak{A})$ need not be a (full) positive type of any element of $\mathfrak{A}$.

For (2) $\Rightarrow$ (1) suppose that t is a witness for (2) and let c be an element with $(\mathfrak{A}, c) \models \bigwedge t$. Setting $t' := \mathrm{tp}^n_X(\mathfrak{A}, c)$ we have $t \subseteq t'$. By induction hypothesis we know that $\oplus_n$ is monotone. Hence, $\psi \in t \oplus_n \mathrm{tp}^n_X(\mathfrak{B})$ implies that $\psi \in t' \oplus_n \mathrm{tp}^n_X(\mathfrak{B})$. As $t' \in T_1(\mathfrak{A})$, (1) follows.

So the given conditions are equivalent, and an analogous statement holds with $\mathfrak{A}$ and $\mathfrak{B}$ interchanged. Hence, we have

$$\begin{aligned}
&\exists y\psi \in \mathrm{tp}^{n+1}_X(\mathfrak{A}+\mathfrak{B})\\
\text{iff}\quad &\psi \in t \oplus_n \mathrm{tp}^n_X(\mathfrak{B})\,, \quad \text{for some } t \in \mathrm{Tp}^n_X[\tau] \text{ such that } \exists y \textstyle\bigwedge t \in \mathrm{tp}^{n+1}_X(\mathfrak{A})\\
\text{or }\ &\psi \in \mathrm{tp}^n_X(\mathfrak{A}) \oplus_n t\,, \quad \text{for some } t \in \mathrm{Tp}^n_X[\tau] \text{ such that } \exists y \textstyle\bigwedge t \in \mathrm{tp}^{n+1}_X(\mathfrak{B})\,.
\end{aligned}$$

This last statement clearly depends only on $\mathrm{tp}^{n+1}_X(\mathfrak{A})$ and $\mathrm{tp}^{n+1}_X(\mathfrak{B})$, and it does so in a monotone way.

Let us consider $\forall Y \psi$ next. We call a pair (S, S') of subsets of $\mathrm{Tp}^n_X[\tau \cup \{Y\}]$ *good,* if

$$\begin{aligned}
&\mathfrak{A} \models \forall Y \bigvee \{ \bigwedge t \mid t \in S \}\,, \\
&\mathfrak{B} \models \forall Y \bigvee \{ \bigwedge t' \mid t' \in S' \}\,, \\
\text{and } &\psi \in t \oplus_n t'\,, \quad \text{for all } t \in S \text{ and } t' \in S'\,.
\end{aligned}$$

Similarly to the previous case one can show that

$$\begin{aligned}
&\forall Y \psi \in \mathrm{tp}^{n+1}_X(\mathfrak{A} + \mathfrak{B}) \\
\text{iff} \quad &\psi \in t \oplus_n t'\,, \quad \text{for all } t \in T_2(\mathfrak{A}),\ t' \in T_2(\mathfrak{B})\,. \qquad (*)
\end{aligned}$$

Let us compare $(*)$ to the existence of a good pair. Clearly, $(*)$ implies that $(T_2(\mathfrak{A}), T_2(\mathfrak{B}))$ is good. Conversely, suppose that (S, S') is good, and let $t \in T_2(\mathfrak{A})$ and $t' \in T_2(\mathfrak{B})$ be types as in $(*)$. We have to show that $\psi \in t \oplus_n t'$. Let $P \subseteq A$ be such that $t = \mathrm{tp}^n_X(\mathfrak{A}, P)$ and let $s \in S$ be a type with $(\mathfrak{A}, P) \models \bigwedge s$. Then $\bigwedge s \in t$, so $s \subseteq t$. Analogously, we obtain some $s' \in S'$ such that $s' \subseteq t'$. Since (S, S') is good, it follows that $\psi \in s \oplus_n s'$. By monotonicity of $\oplus_n$, we therefore have $\psi \in t \oplus_n t'$.

The claim follows since the existence of a good pair is clearly determined by $\mathrm{tp}^{n+1}_X(\mathfrak{A})$ and $\mathrm{tp}^{n+1}_X(\mathfrak{B})$, and since this dependence is monotone.

The remaining cases $\forall y \psi$ and $\exists Y \psi$ use similar techniques. □

Proof (Proof of Lemma 4). Suppose that $I = (p, i, j)$ is not initially relevant. Hence, we have $i > 0$ and $t_{p,i,j} \neq t_{p,i-1,j}$.

If $j = {}_{\wedge}\bullet$ then $({}_{\wedge}t)_{p,i} = (t^{\wedge})_{p,i-1}$ implies that the index $I' := (p, i-1, \bullet^{\wedge})$ also is relevant. Hence, I directly depends on I'.

If $j = {}_{>}\bullet$, then p is not the leftmost position, because otherwise $({}_{>}t)_{p,i}$ and $({}_{>}t)_{p,i-1}$ both are the unique type of the empty word, in contradiction to $({}_{>}t)_{p,i} \neq ({}_{>}t)_{p,i-1}$. So $p-1$ exists. Since the word is consistent, there must be some i' such that $(t^{>})_{p-1,i'} = ({}_{>}t)_{p,i}$ and $(t^{>})_{p-1,i'-1} = ({}_{>}t)_{p,i-1}$. Hence, $I' := (p-1, i', \bullet^{>})$ is relevant and I directly depends on I'.

The case that $j = \bullet_{<}$ is similar. So we are left with the outgoing cases. If $j = \bullet^{\wedge}$ then we have $(t^{\wedge})_{p,i} = 1$ and $(t^{\wedge})_{p,i-1} = 0$. Consequently,

$$\varphi \in ({}_{>}t)_{p,i} \oplus ({}_{\wedge}t)_{p,i} \oplus (t_{<})_{p,i} \quad \text{and} \quad \varphi \notin ({}_{>}t)_{p,i-1} \oplus ({}_{\wedge}t)_{p,i-1} \oplus (t_{<})_{p,i-1}\,.$$

Hence, at least one of

$$({}_{>}t)_{p,i} \neq ({}_{>}t)_{p,i-1}\,, \quad ({}_{\wedge}t)_{p,i} \neq ({}_{\wedge}t)_{p,i-1}\,, \quad \text{or} \quad (t_{<})_{p,i} \neq (t_{<})_{p,i-1}$$

must hold. For the sake of argument, let us assume the first inequation. Then $I' := (p, i, {}_{>}\bullet)$ is relevant and I directly depends on I'. The other inequations can be handled similarly.

If $j = \bullet^{>}$, then

$$\pi(({}_{>}t)_{p,i} \oplus ({}_{\wedge}t)_{p,i}) = (t^{>})_{p,i} \neq (t^{>})_{p,i-1} = \pi(({}_{>}t)_{p,i-1} \oplus ({}_{\wedge}t)_{p,i-1})\,.$$

Hence, at least one of

$$({}_{>}t)_{p,i} \neq ({}_{>}t)_{p,i-1} \quad \text{or} \quad ({}_{\wedge}t)_{p,i} \neq ({}_{\wedge}t)_{p,i-1}$$

must hold. Therefore, we have a direct dependence of I on $(p, i, {}_{>}\bullet)$ or on $(p, i, {}_{\wedge}\bullet)$.

The last case, of $j = {}^{<}\bullet$, is analogous. □

Proof (Proof of Lemma 5). We first prove an auxiliary claim: whenever $(I_k)_{k\leq m}$ is a sequence of relevant indices $I_k = (p_k, i_k, j_k)$, where each I_k depends on I_{k+1} from below and we have

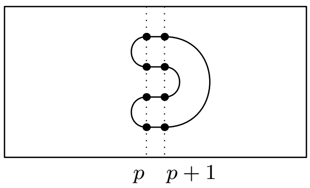
$$p_0 = p_m\,, \quad j_0 = {}_{>}\bullet\,, \quad j_m = {}^{<}\bullet\,, \quad \text{and} \quad p_k < p_0\,, \quad \text{for all } 0 < k < m\,,$$

then it follows that $i_m < i_0$.

The proof is by induction on p_0 from left to right. If p_0 is the leftmost position there is nothing to do since no relevant index can be of the form $(p_0, i, {}_{>}\bullet)$. For the inductive step, note that, by definition of dependence, we have $j_1 = \bullet^{>}$, $j_{m-1} = \bullet_{<}$, and $p_1 = p_{m-1} = p_0 - 1$. Let us consider the set K of all k such that $p_k = p_0 - 1$. We have just seen that 1 and $m-1$ belong to K. The set K consists of subsequences that are interspersed with parts of the sequence of lower dependencies where $p_k < p_0 - 1$. For one such part with endpoints $k_1 < k_2$ in K, we can use the induction hypothesis to obtain $i_{k_2} < i_{k_1}$.

Let us consider a subsequence inside of K with endpoints k_1 and k_2. This subsequence consists of jumps and lower dependencies of outgoing on incoming indices. Both cannot increase the i-component of the indices. Hence, $i_{k_2} \leq i_{k_1}$. If a jump occurs we even have $i_{k_2} < i_{k_1}$. Since the full sequence must contain at least one jump in K or visit positions left of $p-1$, it follows that $i_{m-1} < i_1$. Consistency of the annotation then implies that $i_m < i_0$.

By a symmetric argument we obtain a corresponding claim for an analysis that looks to the right of some position $p_0 = p_m$ that is visited twice.



From these two claims, we can prove the lemma as follows. Consider some cycle in the lower dependency graph. Clearly, it cannot stay at only one position. So, it must span at least two adjacent positions p and $p+1$. Cutting the cycle into (a positive number of) sequences left of $p+1$ with endpoints $(p+1, i, {}_{>}\bullet)$ and $(p+1, i', {}^{<}\bullet)$, and into sequences right of p with endpoints $(p, i, \bullet_{<})$ and $(p, i', \bullet^{>})$, we can apply the claim to derive a contradiction. □

Proof (Proof of Lemma 6). (a) follows from Lemma 3 by induction on p (from left to right for the first equation, and from right to left for the second one) since

$$({}_{>}t)_{p,0} = \pi(({}_{>}t)_{p-1,0} \oplus ({}_{\wedge}t)_{p,0}) ,$$
$$(t_{<})_{p,0} = \pi(({}_{\wedge}t)_{p,0} \oplus (t_{<})_{p+1,0}) ,$$

and $({}_{\wedge}t)_{p,0} = 0 = \mathrm{tp}^n_X(\mathfrak{B}^0_{\{p\}}, p)$.

(b) We proceed by induction on α, simultaneously for all p. For fixed α, the first and third inclusion follow from the second one by a similar induction as in (a), additionally using monotonicity of $\oplus$ and π. So it suffices to show that $\mathrm{tp}^n_X(\mathfrak{B}^\alpha_{\{p\}}, p) \subseteq ({}_{\wedge}t)_{p,m_p}$.

For $\alpha = 0$, this follows from $\mathrm{tp}^n_X(\mathfrak{B}^0_{\{p\}}, p) = 0$. Suppose that $\alpha > 0$. By Lemma 3 and the induction hypothesis, we have

$$\mathrm{tp}^n_X(\mathfrak{B}^{\alpha-1}, p) \subseteq t := ({}_{>}t)_{p,m_p} \oplus ({}_{\wedge}t)_{p,m_p} \oplus (t_{<})_{p,m_p} .$$

To show that $\mathrm{tp}^n_X(\mathfrak{B}^\alpha_{\{p\}}, p) \subseteq ({}_{\wedge}t)_{p,m_p}$ we have to prove that $p \in \varphi^\alpha(\mathfrak{B})$ implies $({}_{\wedge}t)_{p,m_p} = 1$. Hence, suppose that $(\mathfrak{B}^{\alpha-1}, p) \models \varphi$. Then $\varphi \in t$ and it follows, by definition of an annotation, that $(t^{\wedge})_{p,m_p} = 1$. By definition of a history, this implies that $({}_{\wedge}t)_{p,m_p} = 1$, as desired. □

Proof (Proof of Lemma 7). The proof is by induction on α. For fixed α, we distinguish the cases $j = {}^{<}\bullet$, $j = \bullet^{\wedge}$, and $j = \bullet^{>}$.

For $j = \bullet^{>}$, we proceed by induction on p from left to right. So, assume that both claims hold for all $p' < p$. For (a), assume that $\alpha \geq \alpha'$, for all lower ranks α' of I. By the definition of an annotation, we have $t_I = \pi(t_1 \oplus t_2)$, where $t_1 := ({}_{>}t)_{p,i}$ and $t_2 := ({}_{\wedge}t)_{p,i}$. Let i_1 be minimal such that $t_1 = t_{I_1}$ for $I_1 := (p, i_1, {}_{>}\bullet)$. We consider two cases.

If $i_1 = 0$ then we have $t_1 = \mathrm{tp}^n_X(\mathfrak{B}^0_{[0,p)}) \subseteq \mathrm{tp}^n_X(\mathfrak{B}^\alpha_{[0,p)})$ by Lemma 6 and monotonicity.

Otherwise, I_1 is relevant and I depends on I_1 from below. In particular, p is not the leftmost position. In turn, I_1 depends on $I'_1 := (p-1, i'_1, \bullet^{>})$ from below for some i'_1 and $t_{I'_1} = t_{I_1} = t_1$. Then, each lower rank of I'_1 is also a lower rank of I. It follows that $\alpha \geq \alpha'$ for all lower ranks α' of I'_1. By the induction hypothesis on p, we may conclude that $t_1 = t_{I'_1} \subseteq \mathrm{tp}^n_X(\mathfrak{B}^\alpha_{[0,p-1]}) = \mathrm{tp}^n_X(\mathfrak{B}^\alpha_{[0,p)})$.

So, in both cases we have

$$t_1 \subseteq \mathrm{tp}^n_X(\mathfrak{B}^\alpha_{[0,p)}) .$$

Next, we prove that $t_2 \subseteq \mathrm{tp}^n_X(\mathfrak{B}^\alpha_{\{p\}}, p)$. If $t_2 = 0$, this holds since 0 is the minimal type. Otherwise, let i_2 be minimal such that $t_2 = t_{I_2}$ where $I_2 := (p, i_2, {}_{\wedge}\bullet)$. Then I_2 is relevant, I depends on I_2 from below, and I_2 depends on $I'_2 := (p, i_2 - 1, \bullet^{\wedge})$ from below by a jump. Hence, for every lower rank α' of I'_2, $\alpha' + 1$ is a lower rank of I. Thus, $\alpha' \leq \alpha - 1$. By the induction hypothesis on α, it follows that

$$t_2 = t_{I_2} = t_{I'_2} \subseteq \mathrm{tp}^n_X(\mathfrak{B}^{\alpha-1+1}_{\{p\}}, p) = \mathrm{tp}^n_X(\mathfrak{B}^\alpha_{\{p\}}, p) .$$

Since $t_1 \subseteq \mathrm{tp}^n_X(\mathfrak{B}^\alpha_{[0,p)})$ and $t_2 \subseteq \mathrm{tp}^n_X(\mathfrak{B}^\alpha_{\{p\}}, p)$, it follows by Lemma 3 that $t_I \subseteq \mathrm{tp}^n_X(\mathfrak{B}^\alpha_{[0,p]})$. Consequently, α satisfies I.

Next, let us consider (b), still for $j = \bullet^{>}$. The proof is dual to the previous one. We assume that $\alpha < \alpha'$ for all upper ranks α' of $I = (p, i, \bullet^{>})$. Set $I^- := (p, i-1, \bullet^{>})$ and analogously for other indices. We now have the equality $t_{I^-} = \pi(t_1 \oplus t_2)$ with $t_1 = ({}_{>}t)_{p,i-1}$ and $t_2 = ({}_{\wedge}t)_{p,i-1}$. This time, choose i_1 and i_2 maximal, such that $t_1 = t_{I_1^-}$ and $t_2 = t_{I_2^-}$ with I_1 and I_2 as above. In the same way we proved that $t_1 \subseteq \mathrm{tp}^n_X(\mathfrak{B}^\alpha_{[0,p)})$ we can show that $\mathrm{tp}^n_X(\mathfrak{B}^\alpha_{[0,p)}) \subseteq t_1$. It remains to prove that $\mathrm{tp}^n_X(\mathfrak{B}^\alpha_{\{p\}}, p) \subseteq t_2$. The case $t_2 = 1$ is trivial. For $t_2 = 0$, if I_2 is relevant, the proof is again dual to the corresponding proof above. Otherwise, we have $i_2 = m_p$ where m_p is the maximal annotation in the history at the position p. Then Lemma 6 implies $\mathrm{tp}^n_X(\mathfrak{B}^\alpha_{\{p\}}, p) \subseteq t_2$.

This concludes the case of indices outgoing to the right.

The case of $j = {}^{<}\bullet$ is symmetric.

It remains to consider the case that $j = \bullet^{\wedge}$. So let $I = (p, i, \bullet^{\wedge})$ be relevant. Then $(t^{\wedge})_{p,i} = 1$ and $({}_{\wedge}t)_{p,i} = 0$ (this holds both if I is initially relevant and if it is non-initially relevant). For (a), assume that $\alpha \geq \alpha'$ for all lower ranks α' of I. As $t_I = 1$, we have $\varphi \in t_1 \oplus 0 \oplus t_2$ where $t_1 := ({}_{>}t)_{p,i}$ and $t_2 := (t_{<})_{p,i}$. Let i_1 be minimal such that $t_1 = t_{I_1}$ for $I_1 = (p, i_1, {}_{>}\bullet)$. If $i_1 = 0$ then Lemma 6 and monotonicity imply that $t_1 \subseteq \mathrm{tp}^n_X(\mathfrak{B}^\alpha_{[0,p)})$. If $i_1 > 0$ then we have $i > 0$, I is not initially relevant, I_1 is relevant, and I depends on I_1 from below. As I_1 is relevant, p is not the leftmost position. Hence, there is some i_1', such that $t_1 = t_{I_1'}$ for $I_1' = (p-1, i_1', {}^{<}\bullet)$. In the same way as in the case of $j = \bullet^{>}$, we conclude that $t_1 \subseteq \mathrm{tp}^n_X(\mathfrak{B}^\alpha_{[0,p)})$. Symmetrically, we obtain that $t_2 \subseteq \mathrm{tp}^n_X(\mathfrak{B}^\alpha_{(p,\ell)})$. The fact that $0 \subseteq \mathrm{tp}^n_X(\mathfrak{B}^\alpha_{\{p\}}, p)$ follows from 0 being the minimal type. Consequently, Lemma 3 implies that $\varphi \in t_1 \oplus 0 \oplus t_2 \subseteq \mathrm{tp}^n_X(\mathfrak{B}^\alpha, p)$. Hence, $Xx \in \mathrm{tp}^n_X(\mathfrak{B}^{\alpha+1}_{\{p\}}, p)$ and $t_I = 1 = \mathrm{tp}^n_X(\mathfrak{B}^{\alpha+1}_{\{p\}}, p)$.

For (b), we can assume that I is non-initially relevant. Then we have $i > 0$, $(t^{\wedge})_{p,i-1} = 0$, and $({}_{\wedge}t)_{p,i-1} = 0$. It follows that $\varphi \notin t_1 \oplus 0 \oplus t_2$ where $t_1 := ({}_{>}t)_{p,i-1}$ and $t_2 := (t_{<})_{p,i-1}$. From here we can proceed in a manner dual to the above. □

Proof (Proof of Lemma 8). (a) follows from Lemma 6 (b).

(b) Let i be such that $I := (p, i, \bullet^{\wedge})$ is relevant. Then we have $t_I = 1$. From the finiteness of B and the fact that lower dependency paths are acyclic, it follows that I has only a finite number of lower ranks. Let α be a bound on them. From Lemma 7 (a), we conclude that $p \in \varphi^{\alpha+1}(\mathfrak{B})$.

(c) Let us assume that all upper ranks of I exceed α. Using Lemma 7 (b) it follows that $p \notin \varphi^\alpha(\mathfrak{B})$. Contradiction. □

Proof (Proof of Lemma 9). To simplify the exposition we describe $\mathcal{A}$ informally. The automaton nondeterministically guesses a grounded cycle-free upper dependency path from the relevant index $I = (p, i, \bullet^{\wedge})$ and calculates the number of jumps on this path.

For a given φ, let τ be the vocabulary of φ and let $N := f(|\varphi|)$, where f is the function from Fact 1. Then $|\mathrm{MSO}^n_X[\tau]| \leq N$, which implies that $|\mathrm{Tp}^n_X[\tau]| \leq 2^N$. Each history can be encoded as a set of 6-tuples of types. Consequently, there are at most $2^{2^{6N}}$ histories, and the input alphabet of our automaton has twice this size containing an additional bit marking the position of the proposal.

Let us describe the behaviour of the automaton and the amount of memory it needs. The automaton has to check that the input word is a proposal, i.e., a consistent annotated word with exactly one marked position. The existence of exactly one mark can be checked easily requiring only a single bit of memory. Consistency can be also be checked by an automaton since it is a local condition: each pair of consecutive letters must satisfy a certain coherence condition. This can be done by remembering the last letter and comparing it with the current one, thus requiring 2^{6N} bits of memory.

The second task of our automaton is to compute the minimum of the upper ranks of the marked position. To do so it guesses a grounded, cycle-free path in the upper dependency graph to the marked position, and it counts the number of jumps the path contains. This can be done by first guessing a labelling of all indices that tells us whether an index belongs to our path and which index is the next one along the path. Since the length of a history is bounded it follows that every path can cross a given position of the word only a bounded number of times. Therefore, the guessed labelling contains only a finite amount of information per input letter. Again the consistency of the labelling for consecutive input letters is a local condition and can be checked by an automaton. Furthermore, the automaton can count the number of jumps by putting those transitions into Δ_1 that witness a jump on the path. (As there is at most one jump per position, a weight of 1 suffices.) We put the other transitions into Δ_0. To compute the required memory, note that we can use 2^{2N} bits to mark those pairs of types whose indices belong to the path. In addition, we need to remember which of these pairs are connected by the part of the path already seen to the left. This is a binary relation that we can store in 2^{4N} bits.

Summing the memory bounds from above, we see that $\mathcal{A}$ can make do with at most $2^{1+2^{2N}+2^{4N}+2^{6N}}$ states. This gives the desired function g. □

Proof (Proof of Lemma 10). The proof is by induction on the length of grounded cycle-free paths in the upper dependency digraph. We show that, along each such path $I_n, I_{n-1}, \ldots, I_0$ (where I_0 is initial), the rank γ_i of the path $I_i, \ldots, I_0$ bounds $\alpha_{\mathfrak{B}}(I_i)$.

For $n = 0$, I_0 is initially relevant and $\alpha_{\mathfrak{B}}(I)_0 = 0$, and the claim is trivial.

For the inductive step we assume $\gamma_i \geq \alpha_{\mathfrak{B}}(I_i)$ for all $i < n$ and need to show that also $\gamma_n \geq \alpha_{\mathfrak{B}}(I_n)$. Let $I_n := (p, i, j) = I$ and $I_{n-1} := (p', i', j') = I'$.

If I is outgoing then $p' = p$ and $i' \geq i$. As $\alpha_{\mathfrak{B},p} = \alpha_{\mathfrak{B},p'}$ is monotone, we conclude that $\alpha_{\mathfrak{B}}(I) \leq \alpha_{\mathfrak{B}}(I')$. Since $\gamma_n = \gamma_{n-1}$, the claim follows.

Next, let us consider the case that $j = {}_{>}\bullet$ and $j' = \bullet^{>}$. Since this is not a jump, the ranks of the paths are equal. It is therefore sufficient to show that $\alpha_{\mathfrak{B}}(I) \leq \alpha_{\mathfrak{B}}(I')$. In fact we even show that $\alpha_{\mathfrak{B}}(I) = \alpha_{\mathfrak{B}}(I')$. For this we use the fact that I and I' are relevant. By the former, we know that $\alpha_{\mathfrak{B}}(I)$ is the unique α such that $t_I = \mathrm{tp}^n_X(\mathfrak{B}^\alpha_{[0,p)})$ and such that $\mathrm{tp}^n_X(\mathfrak{B}^\alpha_{[0,p)}) \neq \mathrm{tp}^n_X(\mathfrak{B}^{\alpha-1}_{[0,p)})$. The same holds for $\alpha_{\mathfrak{B}}(I')$ and $t_{I'} = t_I$. The case that I is incoming from the right is symmetric.

Finally, suppose that $j = {}_\wedge\bullet$ and $j' = \bullet^\wedge$. This is a jump and the rank increases by 1. Hence, it suffices to show that $\alpha = \alpha' + 1$ where $\alpha := \alpha_{\mathfrak{B}}(I)$ and $\alpha' := \alpha_{\mathfrak{B}}(I')$. As I is relevant and $i' = i-1$, we have $({}_\wedge t)_{p,i} = 1$ and $({}_\wedge t)_{p,i'} = 0$. By the definition of a history, it follows that $(t^\wedge)_{p,i'} = 1$. This implies that

$$\mathrm{tp}^n_X(\mathfrak{B}^\alpha_{\{p\}}, p) = 1\,, \quad \mathrm{tp}^n_X(\mathfrak{B}^{\alpha'}_{\{p\}}, p) = 0\,, \quad \text{and} \quad \mathrm{tp}^n_X(\mathfrak{B}^{\alpha'+1}_{\{p\}}, p) = 1\,.$$

Minimality of α further gives $\mathrm{tp}^n_X(\mathfrak{B}^{\alpha-1}_{\{p\}}, p) = {}_\wedge(t_{p,i-1}) = 0$. Hence, α is the stage at which p enters the fixed point. Consequently, $\alpha = \alpha' + 1$. □

Proof (Proof of Theorem 3). For ease of presentation, we restrict the proof to finite digraphs of path width less than k. Set $\Sigma := \mathcal{P}([k]) \times [k+1]$. We can encode every graph $\mathfrak{G}$ of path width less than k as a Σ-word as follows. Suppose that $(U_i)_{i \leq n}$ is a path decomposition of $\mathfrak{G}$ with $|U_i| = k$, for $i < n$, and $m := |U_n| \leq k$. Such a path decomposition always exists. For every $i \leq n$, we fix a bijection $\mu_i : U_i \to [|U_i|]$. We encode $\mathfrak{G}$ as a word $a_0 \dots a_{kn+m}$ of length $kn + m = \sum_{i \leq n} |U_i|$ where the $(ik + l)$-th letter represents the element v of U_i with $\mu_i(v) = l$. By construction, such an element always exists, and the set of word positions can be thought of as a disjoint union of all U_i. The letters have to encode two pieces of information. We have to record when two elements $u \in U_i$ and $v \in U_{i+1}$ represent the same element of $\mathfrak{G}$ and we have to record whether two elements $u, v \in U_i$ are adjacent in $\mathfrak{G}$. Suppose that $\mu_i(u) = l$. We choose the letter $a_{ik+l} = (X, y) \in \Sigma$ such that

$$X := \{\, \mu_i(v) \mid (u, v) \in E \,\} \quad \text{and} \quad y := \begin{cases} \mu_{i+1}(u) & \text{if } u \in U_{i+1}\,, \\ k & \text{otherwise}\,. \end{cases}$$

Observe that, conversely, every Σ-word $\mathfrak{A}$ encodes in this way a path decomposition of some graph $\mathfrak{G}$ of path width less than k.

Furthermore, given such a Σ-word there exists an MSO-formula $\psi(x, y)$ stating that the elements represented by position x and by position y coincide, and there is an MSO-formula $\vartheta(x, y)$ stating that the elements represented by, respectively, x and y are adjacent.

Let φ be a formula of vocabulary $\{E, X, x\}$. Replacing in φ all equality and edge atoms by ψ and ϑ, respectively, we obtain a formula φ' such that, for all graphs $\mathfrak{G}$ encoded by a word $\mathfrak{A}$, we have

$$(\mathfrak{G}, P, v) \models \varphi \quad \text{iff} \quad (\mathfrak{A}, Q, i) \models \varphi'\,, \quad \text{where } i \text{ and } Q \text{ represent } v \text{ and } P.$$

It follows that $(\varphi')^\alpha(\mathfrak{A})$ represents $\varphi^\alpha(\mathfrak{G})$, for all α. (In particular, every set $(\varphi')^\alpha(\mathfrak{A})$ is closed under the equivalence relation given by ψ.) Consequently, φ is bounded if and only if φ' is.

Thus, the boundedness question for the class of all finite digraphs can be reduced to the boundedness question for the class of all finite words over Σ. □