Decidable (Ac)counting with Parikh and Muller: Adding Presburger Arithmetic to Monadic Second-Order Logic over Tree-Interpretable Structures

Luisa Herrmann ✉ ⌂ ⓘ
Computational Logic Group, TU Dresden, Germany
Center for Scalable Data Analytics and Artificial Intelligence Dresden/Leipzig, Germany

Vincent Peth ✉ ⓘ
Département d'informatique de l'ÉNS, École normale supérieure, CNRS, PSL University, France

Sebastian Rudolph ✉ ⌂ ⓘ
Computational Logic Group, TU Dresden, Germany
Center for Scalable Data Analytics and Artificial Intelligence Dresden/Leipzig, Germany

─── **Abstract** ───

We propose $\omega\text{MSO}\bowtie\text{BAPA}$, an expressive logic for describing countable structures, which subsumes and transcends both Counting Monadic Second-Order Logic (CMSO) and Boolean Algebra with Presburger Arithmetic (BAPA). We show that satisfiability of $\omega\text{MSO}\bowtie\text{BAPA}$ is decidable over the class of labeled infinite binary trees, whereas it becomes undecidable even for a rather mild relaxations. The decidability result is established by an elaborate multi-step transformation into a particular normal form, followed by the deployment of *Parikh-Muller Tree Automata*, a novel kind of automaton for infinite labeled binary trees, integrating and generalizing both Muller and Parikh automata while still exhibiting a decidable (in fact PSpace-complete) emptiness problem. By means of MSO-interpretations, we lift the decidability result to all tree-interpretable classes of structures, including the classes of finite/countable structures of bounded treewidth/cliquewidth/partitionwidth. We generalize the result further by showing that decidability is even preserved when coupling width-restricted $\omega\text{MSO}\bowtie\text{BAPA}$ with width-unrestricted two-variable logic with advanced counting. A final showcase demonstrates how our results can be leveraged to harvest decidability results for expressive μ-calculi extended by global Presburger constraints.

1 Introduction

Monadic second-order logic (MSO) is a popular, expressive, yet computationally reasonably well-behaved logical formalism to deal with various classes of finite or countable structures. It allows for expressing "mildly recursive" structural properties like connectedness or reachability, which go beyond first-order logic yet meet crucial modeling demands in verification, database theory, knowledge representation, and other fields of computational logic. The well-understood link between MSO and automata theory has been very fertile in theory and practice.

Unfortunately, MSO's native capabilities to express cardinality relationships are very limited; they are essentially restricted to fixed thresholds (e.g. "there are at least 10 leaves"). *Counting MSO* [18, 17], denoted CMSO, extends MSO by modulo counting and a finiteness test over sets (e.g. "there is an even number of nodes"), which increases expressiveness in general, while over finite and infinite words or trees, CMSO can be simulated in plain MSO.

In contrast, enriching MSO with *cardinality constraints* [41, 42] (as in "all nodes have as many incoming as outgoing edges") increases the expressivity drastically, but causes satisfiability to become undecidable even over finite words. Decidability (over finite words, trees, or graphs of bounded treewidth [43]) can be recovered when confining set variables occurring in cardinality constraints to those existentially quantified in front (MSO$^{\exists \mathrm{Card}}$). One way to show this is through *Parikh automata* extending finite automata by adding finitely many counters and exploiting the relationship of Presburger arithmetic and semilinear sets [31].

Very recent work [37, 33, 35] extended Parikh word automata to infinite words and investigated the impact of various acceptance conditions, but left a logical characterization as open question. As with the original Parikh automata, one central motivation behind these works is to provide automata-based approaches for specifying and verifying systems beyond regular languages. The study of ω-Parikh automata is motivated by reactive systems, whose behaviors are typically represented by infinite words. Then, the plethora of branching-time approaches in verification should call for a further generalization to ω-tree-automata. Yet, to our knowledge, Parikh automata have not been studied in the context of infinite trees so far.

Another, orthogonal logical approach for describing sets and their cardinalities, motivated by tasks from program analysis and verification, combines the first-order theory of *Boolean algebras* (BA) with *Presburger arithmetic* (PA), resulting in the theory of *BAPA* [45, 46]. As opposed to computationally benign extensions of MSO, BAPA provides stronger support for arithmetic (so one can talk about "all selections with the same number of blue and red nodes" or even "all selections with a share of $70\% - 80\%$ red nodes", modeling statistical information). BAPA usually assumes a finite universe, but can be extended to the countable setting [47]; satisfiability is decidable in either case. However, very regrettably, BAPA lacks non-unary relations, which is outright fatal when it comes to expressing structural properties.

Combining both worlds, we introduce ωMSO$\bowtie$BAPA [ˈoːmzoʊˌbapa], a logic for countable structures, which extends CMSO by BAPA's set operations and Presburger statements, strictly contains MSO$^{\exists \mathrm{Card}}$, and allows for sophisticated structural-arithmetic statements (Section 3). We warrant computational manageability by gently controlling the usage of variables, noting that satisfiability turns undecidable otherwise (Section 4). Exhibiting an elaborate transformation (Section 5), we prove that ωMSO$\bowtie$BAPA formulae over trees can be brought into a very restricted *tree normal form* (TNF). We then provide a characterization showing that the sets of ω-trees satisfying TNF formulae coincide with the sets of trees recognized by *Parikh-Muller Tree Automata* (PMTA), a novel automata model designed by us – and the first-ever automaton model on infinite trees capable of testing Parikh conditions (Section 6). PMTA generalize both Muller and Parikh automata and their emptiness is decidable. The decidability of ωMSO$\bowtie$BAPA over the class of labeled infinite binary trees thereby obtained is then lifted to all *tree-interpretable classes*, including vast and practically relevant classes of finite or countable structures that are bounded in terms of certain width measures (Section 7). Such width-bounded ωMSO$\bowtie$BAPA can be decidably coupled with width-unbounded two-variable logics with advanced counting (Section 8). We demonstrate how to leverage our results to gain decidability results for statistics-enhanced formalisms of the μ-calculus family, which subsumes branching-time logics such as CTL* (Section 9).

2 Preliminaries

As usual, for any $n \in \mathbb{N}$, let $[n] \coloneqq \{1, \ldots, n\}$. In order to **count to infinity**, we use $\mathbb{N}$ extended by (countable) infinity ∞, with arithmetics lifted in the usual way; in particular, $\infty + n = \infty + \infty = (n+1) \cdot \infty = \infty$ and $0 \cdot \infty = 0$ as well as $n \leq \infty$ and $\infty \leq \infty$. For

countable sets A, let $|A|$ denote the element of $\mathbb{N} \cup \{\infty\}$ that corresponds to their cardinality.

To define **countable structures**, assume the following countable, pairwise disjoint sets: a set $\mathbf{C}$ of *(individual) constants*, denoted by $\mathtt{a}, \mathtt{b}, \mathtt{c}, ...$, and, for every $n \in \mathbb{N}$, a set $\mathbf{P}_n$ of *n-ary predicates*, denoted by $\mathtt{P}, \mathtt{R}, \mathtt{Q}, ...$. The set of all predicates will be denoted by $\mathbf{P} := \bigcup_{i \in \mathbb{N}} \mathbf{P}_n$, and we let $\mathrm{ar} : \mathbf{P} \to \mathbb{N}$ such that $\mathrm{ar}(\mathtt{Q}) = n$ iff $\mathtt{Q} \in \mathbf{P}_n$. A *(relational) signature* $\mathbb{S}$ is a union $\mathbb{S}_\mathbf{C} \cup \mathbb{S}_\mathbf{P}$ of finite subsets of $\mathbf{C}$ and $\mathbf{P}$, respectively. An $\mathbb{S}$-*structure* is a pair $\mathfrak{A} = (A, \cdot^\mathfrak{A})$, where A is a countable, nonempty set, called the *domain* of $\mathfrak{A}$ and $\cdot^\mathfrak{A}$ is a function that maps every $\mathtt{a} \in \mathbb{S}_\mathbf{C}$ to a domain element $\mathtt{a}^\mathfrak{A} \in A$, and every $\mathtt{Q} \in \mathbb{S}_\mathbf{P}$ to an $\mathrm{ar}(\mathtt{Q})$-ary relation $\mathtt{Q}^\mathfrak{A} \subseteq A^{\mathrm{ar}(\mathtt{Q})}$.

We define **infinite trees** starting from a finite, non-empty set Σ, called *alphabet*. A *(full) infinite binary tree* (often simply called a *tree*) labeled by some alphabet Σ is a mapping $\xi \colon \{0,1\}^* \to \Sigma$. We denote the set of all trees labeled by Σ by T_Σ^ω. A *finite tree* is a mapping $\xi \colon X \to \Sigma$ where X is a finite, prefix-closed subset of $\{0,1\}^*$. The set of all finite trees over Σ will be denoted by T_Σ. We sometimes refer to the domain X of ξ by $\mathrm{pos}(\xi)$, whose elements we call *positions* or *nodes* of ξ. Given a tree $\xi \in T_\Sigma^\omega$ and a finite, prefix-closed set $X \subseteq \{0,1\}^*$, we denote by $\xi_{|X}$ the finite tree in T_Σ that has X as domain and coincides with ξ on X.

An *(infinite) path* π is an infinite sequence $\pi = \pi_1 \pi_2 \dots$ of positions from $\{0,1\}^*$ such that $\pi_1 = \varepsilon$ and, for each $i \geq 1$, $\pi_{i+1} \in (\pi_i \cdot \{0,1\})$. Given a tree $\xi \in T_\Sigma^\omega$ and a path π, we denote by $\xi(\pi)$ the infinite word $\xi(\pi_1)\xi(\pi_2)\dots$ obtained by concatenating the labels of ξ along π. We denote by $\inf(\xi(\pi))$ the set of all labels occurring infinitely often in $\xi(\pi)$.

We will also find it convenient to represent trees over some given alphabet $\Sigma = \{a_1, \dots, a_n\}$ as structures over the signature $\mathbb{S} = \mathbb{S}_\mathbf{P} = \{\succ_0, \succ_1, \mathtt{P}_{a_1}, \dots, \mathtt{P}_{a_n}\}$: Thereby, a tree $\xi \in T_\Sigma^\omega$ will be represented by the structure $\mathfrak{A}_\xi$ with $A_\xi = \{0,1\}^*$, where $\succ_0^{\mathfrak{A}_\xi} = \{(w, w0) \mid w \in \{0,1\}^*\}$ and $\succ_1^{\mathfrak{A}_\xi} = \{(w, w1) \mid w \in \{0,1\}^*\}$ while $\mathtt{P}_{a_i}^{\mathfrak{A}_\xi} = \{u \in \{0,1\}^* \mid \xi(u) = a_i\}$ for each $i \in [n]$. When there is no danger of confusion, we will simply write ξ instead of $\mathfrak{A}_\xi$.

3 Syntax and Semantics of ωMSO⋈BAPA

We now introduce the logic ωMSO⋈BAPA. The underlying "design principles" for this logical formalism are to have a language that syntactically subsumes and tightly integrates CMSO and BAPA, while still exhibiting favorable computational properties, even over countably infinite structures. To this end, we will first define the language ωMSO·BAPA and then obtain ωMSO⋈BAPA by imposing some syntactic restrictions on the usage of variables.

▶ **Definition 1** (Syntax of ωMSO·BAPA). *Given a signature $\mathbb{S} = \mathbb{S}_\mathbf{C} \cup \mathbb{S}_\mathbf{P}$, together with three countable and pairwise disjoint sets $\mathbf{V}_{\mathrm{ind}}$ of individual variables (denoted $x, y, z, ...$), $\mathbf{V}_{\mathrm{set}}$ of set variables (denoted $X, Y, Z, ...$), and $\mathbf{V}_{\mathrm{num}}$ of number variables (denoted $\mathbb{k}, \ell, m, n...$), we define the following sets of expressions by mutual induction:*

- *the set $\mathbf{I}$ of individual terms:* $\quad \iota ::= \mathtt{a} \mid x$
- *the set $\mathbf{S}$ of set terms ($\mathtt{P}$ being a unary predicate):* $\quad S ::= \{\mathtt{a}\} \mid \mathtt{P} \mid X \mid S^c \mid S_1 \cap S_2 \mid S_1 \cup S_2$
- *the set $\mathbf{N}$ of number terms:*[1] $\quad t ::= \mathtt{n} \mid \infty \mid \mathbb{k} \mid \#S \mid \mathtt{m}\, t \mid t_1 + t_2$
  (with $n \in \mathbb{N}$ and $m \in \mathbb{N} \setminus \{0\}$; we use typewriter font to indicate that we mean an explicit representation of a constant natural number n or m rather than the symbol "n" or "m")
- *the set $\mathbf{F}$ of (unrestricted) formulae:*

$$\varphi ::= \mathtt{Q}(\iota_1, \dots, \iota_n) \mid S(\iota) \mid t \leq_{\mathtt{fin}} t' \mid t \leq t' \mid \#S \equiv_{\mathtt{n}} \mathtt{m} \mid \mathrm{Fin}(S) \mid \mathbf{true} \mid \mathbf{false} \mid$$
$$\neg\varphi \mid \varphi \wedge \varphi' \mid \varphi \vee \varphi' \mid \exists x.\varphi \mid \forall x.\varphi \mid \exists X.\varphi \mid \forall X.\varphi \mid \exists \mathbb{k}.\varphi \mid \forall \mathbb{k}.\varphi$$

---

[1] We will consider number terms obtainable from each other through basic transformations (reordering, factoring, summarizing, rules for ∞) as syntactically equal, allowing us to focus on simplified expressions.

The first six types of atomic formulae will be referred to as predicate atoms, set atoms, classical Presburger atoms, modern Presburger atoms, modulo atoms, *and* finiteness atoms, *respectively. We use* Presburger atoms *and write $t \leq_{(\mathtt{fin})} t'$ to jointly refer to the classical and modern variants. A Presburger atom $t \leq_{(\mathtt{fin})} t'$ is called* simple, *if it contains at most one occurrence of a term of the shape $\#S$ and no occurrences of number variables.*

Note that, for notational homogeneity, we choose to write $X(\iota)$ instead of $\iota \in X$. Where convenient, we will also make use of the Boolean connectives $\Rightarrow$ and $\Leftrightarrow$ as abbreviations with the usual meaning. While the original syntax of ωMSO·BAPA does not provide an explicit equality predicate, both individual and set equality can be expressed (see further below).

▶ **Definition 2** (Semantics of ωMSO·BAPA). *A* variable assignment *(for a structure $\mathfrak{A}$) is a function ν that maps*
- *every individual variable $x \in \mathbf{V}_{\mathrm{ind}}$ to a domain element $\nu(x) \in A$,*
- *every set variable $X \in \mathbf{V}_{\mathrm{set}}$ to a subset $\nu(X) \subseteq A$ of the domain, and*
- *every number variable $\Bbbk \in \mathbf{V}_{\mathrm{num}}$ to a number $\nu(\Bbbk) \in \mathbb{N} \cup \{\infty\}$.*

We write $\nu_{x \mapsto a}$, $\nu_{X \mapsto A'}$, and $\nu_{\Bbbk \mapsto n}$ to denote ν updated in the way indicated in the subscript.

   Given an interpretation $\mathfrak{A}$ and a variable assignment ν, we let the function $\cdot^{\mathfrak{A},\nu}$ map
- *$\mathbf{I}$ to A by letting $\mathtt{a}^{\mathfrak{A},\nu} = \mathtt{a}^{\mathfrak{A}}$ and $x^{\mathfrak{A},\nu} = \nu(x)$,*
- *$\mathbf{S}$ to 2^A by letting*

$$
\begin{array}{lll}
\{\mathtt{a}\}^{\mathfrak{A},\nu} = \{\mathtt{a}^{\mathfrak{A},\nu}\} & X^{\mathfrak{A},\nu} = \nu(X) & (S_1 \cap S_2)^{\mathfrak{A},\nu} = S_1^{\mathfrak{A},\nu} \cap S_2^{\mathfrak{A},\nu} \\
\mathtt{P}^{\mathfrak{A},\nu} = \mathtt{P}^{\mathfrak{A}} & (S^c)^{\mathfrak{A},\nu} = A \setminus S^{\mathfrak{A},\nu} & (S_1 \cup S_2)^{\mathfrak{A},\nu} = S_1^{\mathfrak{A},\nu} \cup S_2^{\mathfrak{A},\nu}
\end{array}
$$

- *$\mathbf{N}$ to $\mathbb{N} \cup \{\infty\}$ by letting*

$$
\begin{array}{lll}
\mathtt{n}^{\mathfrak{A},\nu} = n & \Bbbk^{\mathfrak{A},\nu} = \nu(\Bbbk) & (\mathtt{n}\,t)^{\mathfrak{A},\nu} = n \cdot t^{\mathfrak{A},\nu} \\
\infty^{\mathfrak{A},\nu} = \infty & (\#S)^{\mathfrak{A},\nu} = |S^{\mathfrak{A},\nu}| & (t_1 + t_2)^{\mathfrak{A},\nu} = t_1^{\mathfrak{A},\nu} + t_2^{\mathfrak{A},\nu}
\end{array}
$$

Finally we define satisfaction of formulae from $\mathbf{F}$ as follows: $\mathfrak{A}, \nu$ satisfies

$$
\begin{array}{llll}
\mathtt{Q}(\iota_1,...,\iota_n) & \textit{iff} & ((\iota_1)^{\mathfrak{A},\nu},...,(\iota_n)^{\mathfrak{A},\nu}) \in \mathtt{Q}^{\mathfrak{A}} & \varphi_1 \wedge \varphi_2 \;\; \textit{iff} \;\; \mathfrak{A},\nu \models \varphi_1 \;\textit{and}\; \mathfrak{A},\nu \models \varphi_2 \\
S(\iota) & \textit{iff} & \iota^{\mathfrak{A},\nu} \in S^{\mathfrak{A},\nu} & \varphi_1 \vee \varphi_2 \;\; \textit{iff} \;\; \mathfrak{A},\nu \models \varphi_1 \;\textit{or}\; \mathfrak{A},\nu \models \varphi_2 \\
t_1 \leq t_2 & \textit{iff} & t_1^{\mathfrak{A},\nu} \leq t_2^{\mathfrak{A},\nu} & \exists x.\varphi \;\; \textit{iff} \;\; \mathfrak{A},\nu_{x \mapsto a} \models \varphi \;\textit{for some}\; a \in A \\
t_1 \leq_{\mathtt{fin}} t_2 & \textit{iff} & t_1^{\mathfrak{A},\nu} \leq t_2^{\mathfrak{A},\nu} < \infty & \forall x.\varphi \;\; \textit{iff} \;\; \mathfrak{A},\nu_{x \mapsto a} \models \varphi \;\textit{for all}\; a \in A \\
\#S \equiv_{\mathtt{n}} \mathtt{m} & \textit{iff} & (\#S)^{\mathfrak{A},\nu} = m \mod n & \exists X.\varphi \;\; \textit{iff} \;\; \mathfrak{A},\nu_{X \mapsto A'} \models \varphi \;\textit{for some}\; A' \subseteq A \\
& & \textit{and}\; (\#S)^{\mathfrak{A},\nu} < \infty & \forall X.\varphi \;\; \textit{iff} \;\; \mathfrak{A},\nu_{X \mapsto A'} \models \varphi \;\textit{for all}\; A' \subseteq A \\
\mathrm{Fin}(S) & \textit{iff} & |S^{\mathfrak{A},\nu}| < \infty & \exists \Bbbk.\varphi \;\; \textit{iff} \;\; \mathfrak{A},\nu_{\Bbbk \mapsto n} \models \varphi \;\textit{for some}\; n \in \mathbb{N} \cup \{\infty\} \\
\neg\varphi & \textit{iff} & \mathfrak{A},\nu \not\models \varphi & \forall \Bbbk.\varphi \;\; \textit{iff} \;\; \mathfrak{A},\nu_{\Bbbk \mapsto n} \models \varphi \;\textit{for all}\; n \in \mathbb{N} \cup \{\infty\}
\end{array}
$$

Plus, we always let $\mathfrak{A}, \nu \models \mathbf{true}$ and $\mathfrak{A}, \nu \not\models \mathbf{false}$. For a formula φ, its free variables (denoted $\mathrm{free}(\varphi)$) are defined as usual; φ is a sentence *if $\mathrm{free}(\varphi) = \emptyset$. For sentences, ν does not influence satisfaction, which allows us to write $\mathfrak{A} \models \varphi$ and call $\mathfrak{A}$ a* model *of φ in case $\mathfrak{A}, \nu \models \varphi$ holds for any ν. We call φ* satisfiable *if it has a model.*

▶ **Definition 3** (Syntax of ωMSO⋈BAPA). *From now on, we will make the following assumption (which is easily obtainable via renaming): In every formula, all quantifications use different variable names and these are disjoint from the names of free variables. Given an ωMSO·BAPA formula φ satisfying this assumption, we analyze its constituents as follows:*
- *A (set or individual) variable is called* assertive, *if it is free, or it is existentially quantified and the quantification does not occur inside the scope of a negation or of a universal (set, individual, or number) quantifier.*

- *The set of* delicate *individual and set variables is the smallest set of (non-assertive) variables satisfying the following:*
  - *Every non-assertive set variable occurring in a non-simple Presburger atom is delicate.*
  - *If some atom contains a delicate (individual or set) variable, then all of this atom's non-assertive (individual or set) variables are delicate.*

Then, φ is an ωMSO⋈BAPA formula iff each of its predicate atoms $\mathtt{Q}(\cdots)$ contains at most one delicate variable (possibly in multiple occurrences).

It is easy to see that, despite the above restrictions, ωMSO⋈BAPA entirely encompasses CMSO and MSO$^{\exists\mathrm{Card}}$ (no delicate variables) as well as BAPA (no predicates of arity >1). For convenience and better readability, we will make use of the following abbreviations.

$$
\begin{aligned}
x = y &:= \forall Z. Z(x) \Leftrightarrow Z(y) & \exists x \in S. \varphi &:= \exists x. S(x) \wedge \varphi \\
S \neq \emptyset &:= \exists z. S(z) & \forall x \in S. \varphi &:= \forall x. S(x) \Rightarrow \varphi \\
S \subseteq S' &:= \forall z. S(z) \Rightarrow S'(z) & t = t' &:= (t \leq t') \wedge (t' \leq t) \\
S = S' &:= (S \subseteq S') \wedge (S' \subseteq S) & t =_{\mathtt{fin}} t' &:= (t \leq_{\mathtt{fin}} t') \wedge (t' \leq_{\mathtt{fin}} t)
\end{aligned}
$$

An analysis of these abbreviations reveals that ωMSO⋈BAPA allows for the variables x, y and set variables in S, S' in these abbreviations to be delicate. We will also employ shortcuts specific to the signature $\{\succ_0, \succ_1, \mathtt{P}_a \mid a \in \Sigma\}$ for Σ-labeled trees. Contrary to above, in these shortcuts, x, y, X, Y must not be delicate to warrant inclusion in ωMSO⋈BAPA (Obs. †).

$$
\begin{aligned}
X(x.i) &:= \exists y. x \succ_i y \wedge X(y) & x \succ y &:= (x \succ_0 y) \vee (x \succ_1 y) \\
\varphi_{\uparrow\mathrm{clsd}}(X) &:= \forall z. X(z.0) \vee X(z.1) \Rightarrow X(z) & \varphi_{\mathrm{root}}(x) &:= \neg\exists z.(z \succ x) \\
x \succ^* y &:= \forall Z. Z(y) \wedge \varphi_{\uparrow\mathrm{clsd}}(Z) \Rightarrow Z(x) & x \succ^+ y &:= (x \succ^* y) \wedge (x \neq y) \\
\varphi_{\downarrow}(x, X) &:= \forall z. \big(X(z) \Leftrightarrow x \succ^* z\big) \\
\varphi_{\mathrm{path}}(X) &:= X \neq \emptyset \wedge \varphi_{\uparrow\mathrm{clsd}}(X) \wedge \forall z \in X. \big(X(z.0) \Leftrightarrow \neg X(z.1)\big) \\
\varphi_{\inf}(X) &:= \exists Z. \varphi_{\mathrm{path}}(Z) \wedge \forall z \in Z. \exists z' \in X.(z \succ^+ z') \\
\varphi_{\inf}^{\cap}(X, Y) &:= \exists Z. Z \subseteq X \wedge Z \subseteq Y \wedge \varphi_{\inf}(Z)
\end{aligned}
$$

▶ **Example 4.** We use ωMSO⋈BAPA to specify the class of all labeled infinite binary trees over the alphabet $\Sigma = \{\mathrm{blue}, \mathrm{red}, \mathrm{green}, \mathrm{yellow}, \mathrm{black}\}$ satisfying the following property:

"There is a path X and some node x on X such that the following hold:

1. *For every infinite selection Y of blue nodes from the x-descendants on the path X, there is a selection Y' of red nodes from the whole tree, such that*
   a. *Y and Y' contain the same number of nodes with infinitely many green descendants,*
   b. *Y contains twice as many nodes as Y' having less than 10 yellow descendants.*
2. *For every finite selection Z of blue x-descendants, the total number of nodes lying on paths from x to nodes of Z is even."*

$$
\begin{aligned}
&\exists X. \exists x. \varphi_{\mathrm{path}}(X) \wedge X(x) \wedge \exists V_0. \varphi_{\downarrow}(x, V_0) \wedge \\
&\quad \Big(\exists V_1. \big(\forall v_1. V_1(v_1) \Leftrightarrow \exists V_{\downarrow}^{v_1}. \varphi_{\downarrow}(v_1, V_{\downarrow}^{v_1}) \wedge \neg\mathrm{Fin}(V_{\downarrow}^{v_1} \cap \mathtt{P}_{\mathrm{green}})\big) \wedge \\
&\qquad \exists V_2. \big(\forall v_2. V_2(v_2) \Leftrightarrow \exists V_{\downarrow}^{v_2}. \varphi_{\downarrow}(v_2, V_{\downarrow}^{v_2}) \wedge \#(V_{\downarrow}^{v_2} \cap \mathtt{P}_{\mathrm{yellow}}) \leq 10\big) \wedge \\
&\qquad \Big(\forall Y. \big(\neg\mathrm{Fin}(Y) \wedge Y \subseteq X \cap V_0 \cap \mathtt{P}_{\mathrm{blue}}\big) \Rightarrow \\
&\qquad\quad \exists Y'. Y' \subseteq \mathtt{P}_{\mathrm{red}} \wedge \#(Y \cap V_1) = \#(Y' \cap V_1) \wedge \#(Y \cap V_2) = 2\,\#(Y' \cap V_2)\big)\Big) \Big) \wedge \\
&\quad \Big(\forall Z. \big(\mathrm{Fin}(Z) \wedge Z \subseteq V_0 \cap \mathtt{P}_{\mathrm{blue}}\big) \Rightarrow \\
&\qquad \exists V_3. \big(\forall v_3. V_3(v_3) \Leftrightarrow (x \succ^+ v_3 \wedge \exists z \in Z. v_3 \succ^* z')\big) \wedge \#V_3 \equiv_2 0 \Big)
\end{aligned}
$$

Therein, we use set variables capturing all descendants of x (V_0); all nodes with infinitely many green descendants (V_1); all nodes with less than 10 yellow descendants (V_2); and all nodes between x and elements of Z (V_3). Analysing the variables yields that X, x, V_0, V_1, and V_2 are assertive, while Y and Y' are delicate due to their occurrence in the non-simple Presburger atoms in the fifth line. Delicacy is not inherited further, thus no two delicate variables occur in any predicate atom. Therefore the formula is indeed in ωMSO⋈BAPA. Note that it is crucial that V_1 and V_2 are defined "prematurely" outside the scope of $\forall Y$, so they become assertive and thus their occurrence in the (non-simple) Presburger atoms does not turn them delicate. This technique of "encapsulating" unary descriptions into assertive set variables unveils significant additional expressiveness of ωMSO⋈BAPA. See also Section 10 for a discussion on a handier syntax for this.

4 Mildly Extending ωMSO⋈BAPA Leads to Undecidability

Just slightly relaxing the syntax of ωMSO⋈BAPA allows us to express Hilbert's 10th Problem.

▶ **Definition 5** (Positive Diophantine Equation). *A* positive Diophantine equation $\mathcal{D}$ *is a tuple* $(NV, M, (n_w)_{w \in M}, (m_w)_{w \in M})$ *where NV is a non-empty, ordered set $\{\varkappa_1, \ldots, \varkappa_{\Bbbk}\}$ of number variables; M (the variable products or* monomials*) is a finite and non-empty, prefix-closed set of sorted variable sequences, i.e.,*

$$M \subseteq \{\underbrace{\varkappa_1 \ldots \varkappa_1}_{i_1} \ldots \underbrace{\varkappa_k \ldots \varkappa_k}_{i_k} \mid i_1, \ldots, i_k \in \mathbb{N}\};$$
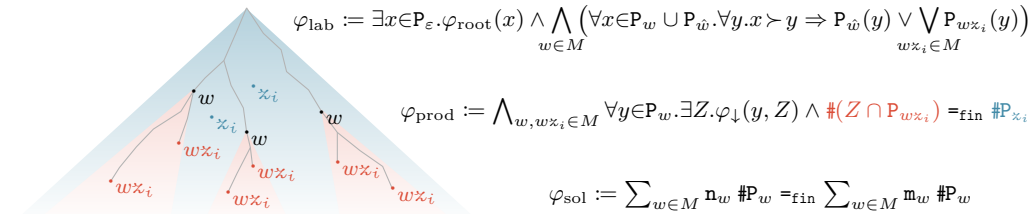
and all n_w and m_w are from $\mathbb{N}$ and encode the monomial coefficients on either side of the equation. A positive Diophantine equation is solvable *if it admits a solution, where a* solution *for $\mathcal{D} = (NV, M, (n_w)_{w \in M}, (m_w)_{w \in M})$ is a variable assignment $\nu : NV \to \mathbb{N}$ satisfying*

$$\sum_{w = \varkappa_1^{i_1} \ldots \varkappa_k^{i_k} \in M} n_w \cdot \nu(\varkappa_1)^{i_1} \cdot \ldots \cdot \nu(\varkappa_k)^{i_k} = \sum_{w = \varkappa_1^{i_1} \ldots \varkappa_k^{i_k} \in M} m_w \cdot \nu(\varkappa_1)^{i_1} \cdot \ldots \cdot \nu(\varkappa_k)^{i_k}.$$

Solvability of positive Diophantine equations is undecidable, which is a straightforward consequence of the undecidability of arbitrary Diophantine equations over integers [49].

We will show that for any $\mathcal{D}$, we can compute an ωMSO·BAPA sentence $\varphi_\mathcal{D}$ whose satisfiability over labeled trees coincides with solvability of $\mathcal{D}$, despite $\varphi_\mathcal{D}$ being only "minimally outside" ωMSO⋈BAPA — also contrasting the fact that sentences of this shape still warrant decidable satisfiability over finite words [40, Thm. 8.13].

As detailed in Figure 1, we let $\varphi_\mathcal{D} := \varphi_{\text{lab}} \wedge \varphi_{\text{prod}} \wedge \varphi_{\text{sol}}$ characterize trees labeled by w and $\hat{w}$, for $w \in M$, such that each model ξ of $\varphi_\mathcal{D}$ corresponds to a solution ν of $\mathcal{D}$ as follows: for each $\varkappa \in NV$, the number of nodes in ξ labeled with $\varkappa$ (i.e., $\#\mathsf{P}_\varkappa$) equals the number that ν assigns to $\varkappa$. Likewise, for each variable product $w\varkappa_i \in M$, we ensure that

$$\varphi_{\text{lab}} := \exists x \in \mathsf{P}_\varepsilon . \varphi_{\text{root}}(x) \wedge \bigwedge_{w \in M} \Big(\forall x \in \mathsf{P}_w \cup \mathsf{P}_{\hat{w}} . \forall y . x \succ y \Rightarrow \mathsf{P}_{\hat{w}}(y) \vee \bigvee_{w\varkappa_i \in M} \mathsf{P}_{w\varkappa_i}(y) \Big)$$

$$\varphi_{\text{prod}} := \bigwedge_{w, w\varkappa_i \in M} \forall y \in \mathsf{P}_w . \exists Z . \varphi_\downarrow(y, Z) \wedge \#(Z \cap \mathsf{P}_{w\varkappa_i}) =_{\text{fin}} \#\mathsf{P}_{\varkappa_i}$$

$$\varphi_{\text{sol}} := \sum_{w \in M} \mathtt{n}_w \ \#\mathsf{P}_w =_{\text{fin}} \sum_{w \in M} \mathtt{m}_w \ \#\mathsf{P}_w$$

Figure 1 Illustration of the intended model structure and definition of $\varphi_\mathcal{D} := \varphi_{\text{lab}} \wedge \varphi_{\text{prod}} \wedge \varphi_{\text{sol}}$

$\#P_{w\varkappa_i} = \#P_w \cdot \#P_{\varkappa_i}$. To this end, we stipulate via φ_{lab} that for any w, all w-labeled nodes are pairwise $\succ^*$-incomparable, and every $w\varkappa$-labeled node has exactly one w-labeled ancestor (using the label $\hat{w}$ for "padding" between w and $w\varkappa_i$), and we enforce via φ_{prod} that for any $w, w\varkappa_i \in M$, each subtree rooted in a w-labeled node contains precisely as many $w\varkappa_i$-labeled nodes as there are $\varkappa_i$-labeled nodes in the whole tree. Finally, under the conditions enforced by φ_{lab} and φ_{prod}, φ_{sol} implements that the model indeed encodes a solution of the given $\mathcal{D}$.

While the first conjunct is pure MSO and the third conjunct is a variable-free Presburger atom, the second conjunct is not in $\omega\mathrm{MSO}\bowtie\mathrm{BAPA}$: $\exists Z$ occurs inside the scope of $\forall y$, thus Z is not assertive. Yet, as discussed in Section 3 (Obs. †), this is at odds with Z occurring in $\varphi_\downarrow(y, Z)$.

▶ **Proposition 6.** *For any positive Diophantine equation $\mathcal{D}$, satisfaction of $\varphi_{\mathcal{D}}$ over (finite or infinite) labeled trees coincides with solvability of $\mathcal{D}$. Consequently, satisfiability of the class of $\omega MSO \cdot BAPA$ sentences of the shape $\varphi_{\mathcal{D}}$ is undecidable.*

5   Transformation into Normal Form

Toward establishing our decidability result, we show that $\omega\mathrm{MSO}\bowtie\mathrm{BAPA}$ formulae can be transformed into a specific, very restricted normal form. To this end, we use a variety of techniques, mostly known from the literature, but with some adjustments to our setting; thus, due to space, we will restrict ourselves to a high-level description and examples. The normalization procedure is subdivided into two phases: The first phase, establishing the *general normal form* (GNF), is valid independently of the underlying class of structures. The second phase, yielding the *tree normal form* (TNF), is specific to the class of labeled trees.

Given an $\omega\mathrm{MSO}\bowtie\mathrm{BAPA}$ formula, substitute complex set expressions in modulo and finiteness atoms by new set variables (e.g. $\mathrm{Fin}(P \cap X)$ becomes $\exists Y.(Y = P \cap X) \wedge \mathrm{Fin}(Y)$), remove set operations from set atoms (e.g. turning $(P^c \cap X)(y)$ into $\neg P(y) \wedge X(y)$), and rewrite all simple Presburger atoms into plain MSO (e.g. $2\#P \leq 3$ becomes $\forall xy.P(x) \wedge P(y) \Rightarrow x{=}y$). Then, *skolemize* all assertive variables (e.g. $\exists x.\exists X.\forall y.R(x,y) \Rightarrow X(y)$ becomes $\forall y.R(c_x, y) \Rightarrow P_X(y)$). Next *"presburgerize"* all non-Presburger atoms containing (only) delicate variables (e.g. replacing $\#X \equiv_3 1$ with $\exists \Bbbk.\#X =_{\mathrm{fin}} 3\Bbbk + 1$), which may require to turn delicate individual into set variables (e.g. $\forall y.P(y) \Rightarrow X(y)$ becomes $\forall Y.(\#Y = 1) \wedge 1 \leq \#(P \cap Y) \Rightarrow 1 \leq \#(X \cap Y)$). The resulting formula exhibits a clear separation of variable usage: Presburger atoms use delicate and number variables, all other atoms use non-delicate variables. In a subsequent step, we *"disentangle"* the quantifiers, such that the scopes of quantified number or delicate variables are strictly separated from those of non-delicate variables.[2]

We next apply *"vennification"*: a technique known from BAPA. In essence, we introduce new number variables to count the number of elements contained in every *Venn region*, that is, every possible combination of set (non-)memberships (with this, $\#(P \cup X) \leq \#P^c$ becomes $\Bbbk_{P \cap X} + \Bbbk_{P^c \cap X} + \Bbbk_{P \cap X^c} \leq \Bbbk_{P^c \cap X} + \Bbbk_{P^c \cap X^c}$). This allows us to remove all delicate set variables from our formula. We are now in the setting where we can apply the well-known *quantifier elimination* for Presburger Arithmetic over the "purely arithmetic" subformulae (which may produce new modulo atoms) – since the latter is classically defined for $\mathbb{N}$ instead of $\mathbb{N} \cup \{\infty\}$, we require a pre-processing step implementing a vast case-distinction as to which of the Venn regions are infinite. As a consequence, we obtain a formula free of number variables, with all Presburger atoms being classic and outside any quantifier scope.

---

[2] While this transformation is not very complicated technically, it may incur nonelementary blowup.

Finally, we *"de-skolemize"*: all constants and unary predicates introduced via the initial skolemization, but also by the intermediate transformation steps, are projected away from the signature, re-interpreting them as existentially quantified individual and set variables. We thus recover "proper" equivalence with the initial formula. Last, we bring the formula in disjunctive normal form and pull the trailing existential individual quantifiers inside.

▶ **Definition 7** (General Normal Form). *A* Parikh constraint *is a classical Presburger atom without number variables and where all occurring set terms are set variables. An ωMSO⋈BAPA formula is in* general normal form *(GNF), if it is of the shape*

$$\exists X_1. \cdots \exists X_n. \bigvee_{i=1}^{k} \big(\varphi_i \wedge \bigwedge_{j=1}^{l_i} \chi_{i,j} \big),$$

where the φ_i are CMSO formulae,[3] whereas the $\chi_{i,j}$ are (unnegated) Parikh constraints.

▶ **Theorem 8.** *For every ωMSO⋈BAPA formula φ, it is possible to compute an equivalent formula φ' in general normal form.*

We now focus on the case of labeled trees. Very similar to the case of CMSO, under this assumption, we can equivalently transform the GNF formula into one without occurrences of modulo and finiteness atoms. We rewrite $\#X \equiv_n \mathtt{m}$ into the formula

$$\mathrm{Fin}(X) \wedge \exists X_0 ... \exists X_{n-1}. \Big(\exists x. \big(\varphi_{\mathrm{root}}(x) \wedge \bigwedge_{\substack{0 \le i < n \\ i \ne m}} \neg X_i(x) \big) \wedge \forall x. \big((\exists y \in X. x \succ^* y) \vee X_0(x) \big) \wedge$$
$$\bigwedge_{i,j \in \{0,...,n-1\}} \forall z. \big(X_i(z.0) \wedge X_j(z.1) \Rightarrow (\neg X(z) \Rightarrow X_{i \oplus j}(z)) \wedge (X(z) \Rightarrow X_{i \oplus j \oplus 1}(z)) \big) \Big),$$

where $\oplus$ denotes addition modulo n. Finally, we replace all occurrences of $\mathrm{Fin}(X)$ by $\varphi_{\mathrm{fin}}(X)$, as defined in Section 3. Thus, when employing ωMSO⋈BAPA to describe labeled trees, we can confine ourselves to an even more restrictive normal form.

▶ **Definition 9** (Tree Normal Form). *An ωMSO⋈BAPA formula is in* tree normal form *(TNF), if it is of the shape*

$$\exists X_1. \cdots \exists X_n. \bigvee_{i=1}^{k} \big(\varphi_i \wedge \bigwedge_{j=1}^{l_i} \chi_{i,j} \big),$$

where the φ_i are plain MSO formulae and the $\chi_{i,j}$ are (unnegated) Parikh constraints.

▶ **Theorem 10.** *For every ωMSO⋈BAPA formula φ, it is possible to compute a formula φ' in tree normal form that is equivalent to φ over all labeled infinite binary trees.*

6   Parikh-Muller Tree Automata

In this section, we introduce a novel type of automata, combining and generalizing Parikh tree automata and Muller tree automata. We prove that the tree languages recognized by this automaton type coincide with those definable by TNF formulae. Moreover, we show that the emptiness problem of this automaton model is decidable. In combination, this yields us decidable satisfiability of ωMSO⋈BAPA over labeled infinite binary trees.

Variable-adorned Trees, Semilinear Sets, and Extended Parikh Maps

Given a finite set $\mathbf{V} \subseteq (\mathbf{V}_{\mathrm{ind}} \cup \mathbf{V}_{\mathrm{set}})$, we denote by $\Phi_{\mathbf{V}}$ the set of all variable assignments of variables from $\mathbf{V}$ to elements/subsets of $\{0,1\}^*$. The *set of $\mathbf{V}$-models* of a formula φ is the set

---

[3] Recall that CMSO is MSO with modulo and finiteness atoms over set variables.

$\mathcal{L}_{\mathbf{V}}(\varphi) \coloneqq \{(\xi, \nu) \mid \xi \in T_{\Sigma}^{\omega}, \nu \in \Phi_{\mathbf{V}}, \xi, \nu \models \varphi\}$ and by $\mathcal{L}(\varphi)$ we mean $\mathcal{L}_{\mathrm{free}(\varphi)}(\varphi)$. To represent **V**-models, it is convenient to encode variable assignments $\nu \in \Phi_{\mathbf{V}}$ into the alphabet. For this, we let $\Sigma_{\mathbf{V}} = \Sigma \times 2^{\mathbf{V}}$ be a new alphabet and identify $\Sigma_{\emptyset}$ with Σ. We say that a tree $\xi \in T_{\Sigma_{\mathbf{V}}}^{\omega}$ is *valid* (i.e., it encodes a variable assignment) if for each individual variable x in **V** there is exactly one position in ξ where x occurs. As there is a bijection between $T_{\Sigma}^{\omega} \times \Phi_{\mathbf{V}}$ and the set of all valid trees in $T_{\Sigma_{\mathbf{V}}}^{\omega}$, we use these two views interchangeably.

A set $C \subseteq \mathbb{N}^s$, $s \geq 1$, is *linear* if it is of the form $C = \{\vec{v}_0 + \sum_{i \in [l]} m_i \vec{v}_i \mid m_1, \ldots, m_l \in \mathbb{N}\}$ for some $l \in \mathbb{N}$ and vectors $\vec{v}_0, \ldots, \vec{v}_l \in \mathbb{N}^s$. Any finite union of linear sets is called *semilinear*.

Given two vectors $\vec{v} = (v_1, \ldots, v_s) \in \mathbb{N}^s$ and $\vec{v}' = (v_1', \ldots, v_{s'}') \in \mathbb{N}^{s'}$, we define their *concatenation* $\vec{v} \cdot \vec{v}'$ as the vector $(v_1, \ldots, v_s, v_1', \ldots, v_{s'}') \in \mathbb{N}^{s+s'}$. This definition is lifted to sets by letting $C \cdot C' = \{\vec{v} \cdot \vec{v}' \mid \vec{v} \in C, \ \vec{v}' \in C'\} \subseteq \mathbb{N}^{s+s'}$ for $C \subseteq \mathbb{N}^s, C' \subseteq \mathbb{N}^{s'}$.

▶ **Lemma 11** ([30, 31]). *The family of semilinear sets of $\mathbb{N}^s$ coincides with the family of Presburger sets of $\mathbb{N}^s$ (i.e., sets of the form $\{(x_1, \ldots, x_s) \mid \varphi(x_1, \ldots, x_s)\}$ for a Presburger formula φ). Semilinear sets are closed under union, intersection, complement, and concatenation.*

Given an alphabet Σ and some finite $D \subseteq \mathbb{N}^s$ for $s \geq 1$, our automaton model works with symbols from $\Sigma \times D$. Thus we use the *projections* $\cdot_{\Sigma} : \Sigma \times D \to \Sigma$ with $(a, d)_{\Sigma} = a$ and $\cdot_D : \Sigma \times D \to D$ with $(a, d)_D = d$, which we will also apply to finite and infinite trees, resulting in a pointwise substitution of labels. Moreover, the *extended Parikh map* $\Psi : T_{\Sigma \times D} \to \mathbb{N}^s$ is defined for each finite, non-empty tree $\xi \in T_{\Sigma \times D}$ by $\Psi(\xi) = \sum_{i \in \mathrm{pos}(\xi)} (\xi(i))_D$.

Automaton Model

We now formally introduce our notion of a *Parikh-Muller Tree Automaton (PMTA)*, which recognizes infinite trees employing a Muller acceptance condition while also testing some finite initial tree part for an arithmetic property related to Parikh's commutative image [50]. This is implemented by utilizing a finite number of global counters, which are "blindly" increased throughout the run, but are read off only once a posteriori – when it is verified whether the tuple of the final counter values belongs to a given semilinear set.

▶ **Definition 12** (Parikh-Muller Tree Automaton). *Let Σ be an alphabet, let $s \in \mathbb{N} \setminus \{0\}$, let $D \subseteq \mathbb{N}^s$ be finite, and denote $(\Sigma \times D) \cup \Sigma$ by Ξ. A PMTA (of* dimension s*) is a tuple $\mathcal{A} = (Q, \Xi, q_I, \Delta, \mathcal{F}, C)$ where $Q = Q_P \cup Q_{\omega} \cup \{q_I\}$ is a finite set of* states *with Q_P, Q_{ω} disjoint and q_I being the* initial state*, $\Delta = \Delta_P \cup \Delta_{\omega}$ is the* transition relation *with*

$$\Delta_P \subseteq (Q_P \cup \{q_I\}) \times (\Sigma \times D) \times Q \times Q \quad and \quad \Delta_{\omega} \subseteq (Q_{\omega} \cup \{q_I\}) \times \Sigma \times Q_{\omega} \times Q_{\omega},$$

$\mathcal{F} \subseteq 2^{Q_{\omega}}$ is a set of final state sets*, and $C \subseteq \mathbb{N}^s$ is a semilinear set named* final constraint*.*

▶ **Definition 13** (Semantics of PMTA). *A* run *of $\mathcal{A}$ on a tree $\zeta \in T_{\Xi}^{\omega}$ is a tree $\kappa_{\zeta} \in T_Q^{\omega}$ whose root is labeled with q_I and which respects Δ jointly with ζ. By definition of Δ, if a run exists, then $\zeta^{-1}(\Sigma \times D)$ is prefix-closed; we denote $\zeta_{|\zeta^{-1}(\Sigma \times D)}$ by ζ_{cnt}. A run κ_{ζ} is accepting if*
1. *for each path π, we have $\inf(\kappa_{\zeta}(\pi)) \in \mathcal{F}$, and*
2. *if $\mathrm{pos}(\zeta_{\mathrm{cnt}}) \neq \emptyset$, then $\Psi(\zeta_{\mathrm{cnt}}) \in C$.*
Note that, by the first condition, κ_{ζ} being accepting implies finiteness of ζ_{cnt} and, thus, well-definedness of the sum in $\Psi(\zeta_{\mathrm{cnt}})$. The set of all accepting runs of $\mathcal{A}$ on ζ will be denoted by $\mathrm{Run}_{\mathcal{A}}(\zeta)$. Then, the tree language *of $\mathcal{A}$, denoted by $\mathcal{L}(\mathcal{A})$, is the set*

$$\mathcal{L}(\mathcal{A}) \coloneqq \{\xi \in T_{\Sigma}^{\omega} \mid \exists \zeta \in T_{\Xi}^{\omega} \text{ with } \mathrm{Run}_{\mathcal{A}}(\zeta) \neq \emptyset \text{ and } (\zeta)_{\Sigma} = \xi\}.$$

We highlight that, by choosing $\Delta_P = \emptyset$, we reobtain the well-known concept of a *Muller tree automaton (MTA)*. In this case, we can drop Q_P, D, Δ_P, and C from $\mathcal{A}$'s specification without affecting its semantics. Thus, we define an MTA $\mathcal{A}$ by the tuple $(Q_\omega, \Sigma, q_I, \Delta_\omega, \mathcal{F})$.

For alphabets Σ, Γ, a *relabeling* (from Σ to Γ) is a mapping $\tau \colon \Sigma \to \mathcal{P}(\Gamma)$. We extend it to a mapping $\tau \colon T_\Sigma^\omega \to \mathcal{P}(T_\Gamma^\omega)$ by letting $\xi' \in \tau(\xi)$ if and only if for each position $\varrho \in \{0,1\}^*$, we have $\xi'(\varrho) \in \tau(\xi(\varrho))$. Note that the reverse τ^{-1} of a relabeling τ is again a relabeling.

▶ **Proposition 14.** *The set of tree languages recognized by Parikh-Muller tree automata is closed under union, intersection, and relabeling.*

Proof (sketch). As the proof techniques are rather standard and some of them were already presented in earlier work [37], we only sketch the main ideas here. Let $\mathcal{A}_1$ and $\mathcal{A}_2$ be PMTA.

For the *union*, we construct a PMTA that starts in a fresh initial state. From there, it can either enter the transitions of $\mathcal{A}_1$ or of $\mathcal{A}_2$; we keep apart the final constraints of $\mathcal{A}_1$ and $\mathcal{A}_2$ by using one additional dimension. The *intersection* PMTA is constructed as the Cartesian product of $\mathcal{A}_1$ and $\mathcal{A}_2$; it uses the concatenation of final constraints of both given PMTA and, as $\mathcal{A}_1$ and $\mathcal{A}_2$ might not "arithmetically test" the same initial tree part, it can nondeterministically freeze parts of its counters on different paths. *Relabeling* is trivial. ◀

Correspondence of PMTA and ωMSO⋈BAPA

We now provide a logical characterization of PMTAs, by showing that a tree language is recognized by a PMTA precisely if it is the set of tree models of some ωMSO⋈BAPA sentence. The "only if" part is established by Proposition 15 and the "if" part by Proposition 17.

▶ **Proposition 15.** *For any PMTA $\mathcal{A}$, there is an ωMSO⋈BAPA sentence φ with $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\varphi)$.*

Proof. Given a PMTA $\mathcal{A} = (Q, \Xi, q_I, \Delta, \mathcal{F}, C)$, we adopt (and slightly simplify) the idea from [42, Thm. 10] of how to encode counter values and the semilinear set C, and combine it with the usual construction to define the behavior of an MTA by means of an MSO formula: The existence of a run is defined by a sequence of existential set quantifiers representing the states of $\mathcal{A}$; one additional universal set quantifier ranging over paths is used to encode the Muller acceptance condition. Furthermore, we (outermost) existentially quantify over "counter contributions" using set quantifiers $Z_1^0, \dots, Z_1^K, \dots, Z_s^0, \dots, Z_s^K$ (with s being the number of counters and K the greatest counter increment occurring in $\mathcal{A}$'s transitions) – the presence of a variable $Z_i^{d_i}$ at a position indicates that d_i has to be added to the ith counter to simulate the extended Parikh map. Then we enforce satisfaction of the final constraint C by adding the conjunct φ_C defined as follows: By definition of C, there are $k, l \in \mathbb{N} \setminus \{0\}$ and linear polynomials $p_1, \dots, p_k \colon \mathbb{N}^l \to \mathbb{N}^s$ such that C is the union of the images of $p_1, \dots, p_k$. Assume $p_g(m_1, \dots, m_l) = \vec{v}_0 + m_1 \vec{v}_1 + \dots + m_l \vec{v}_l$ with $\vec{v}_j = (v_{j,1}, \dots, v_{j,s})$. Then, using number variables $m_1, \dots, m_l$, we encode p_g by

$$\varphi_{p_g} \coloneqq \exists m_1 \dots \exists m_l. \bigwedge_{i=1}^s \left(\sum_{d=0}^K \mathtt{d\#} Z_i^d =_{\mathtt{fin}} \mathtt{v}_{0,i} + \mathtt{v}_{1,i} m_1 + \dots + \mathtt{v}_{l,i} m_l \right),$$

and let $\varphi_C \coloneqq \left(\bigwedge_{i=1}^s \bigwedge_{d=0}^K \forall x. \neg Z_i^d(x) \right) \vee \varphi_{p_1} \vee \dots \vee \varphi_{p_k}$. This finishes the construction of the overall sentence specifying $\mathcal{L}(\mathcal{A})$, which can be easily shown to be in ωMSO⋈BAPA. ◀

The other direction is proved by an induction on the structure of TNF formulae involving the closure properties of PMTA. The last piece that needs to be shown for this is the recognizability of the models of a Parikh constraint.

▶ **Lemma 16.** *For each Parikh constraint χ there is a PMTA $\mathcal{A}$ with $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\chi)$.*

Proof. We assume w.l.o.g. that χ is of the form $\mathtt{c} + \sum_{i\in[r]} \mathtt{c_i} \# X_i \leqslant_{\mathtt{fin}} \mathtt{d} + \sum_{j\in[k]} \mathtt{d_j} \# Y_j$ where all X_i are pairwise distinct, and all Y_j likewise. Given a subset $\theta \subseteq \mathrm{free}(\chi)$, we denote by $|\theta|_X$ the number $\sum_{X_i\in\theta} c_i$ (and similar for $|\theta|_Y$). Then, assuming $\xi(\varrho) = (\sigma_\varrho^\xi, \theta_\varrho^\xi)$, we get

$$\mathcal{L}(\chi) = \{\xi \in T^\omega_{\Sigma_{\mathrm{free}(\chi)}} \mid c + \textstyle\sum_{\varrho\in\mathrm{pos}(\xi)} |\theta_\varrho^\xi|_X \le d + \sum_{\varrho\in\mathrm{pos}(\xi)} |\theta_\varrho^\xi|_Y < \infty\}$$

and, by the condition $< \infty$, both sums can add up only finitely many non-zero elements. Therefore, $\xi \in \mathcal{L}(\chi)$ holds exactly if there is a non-empty, finite, prefix-closed $Z \subset \{0,1\}^*$ that comprises all positions holding variable assignments and for which $\xi|_Z$ satisfies χ. This condition can be verified by a PMTA defined in the following.

Let $D = \{(i,j) \mid 0 \le i \le \sum_{l\in[r]} c_l, 0 \le j \le \sum_{l\in[k]} d_l\}$. We construct the PMTA $\mathcal{A} = (\{q_I, q_f\}, \Xi, q_I, \Delta, \{\{q_f\}\}, C)$ with $\Xi = (\Sigma_{\mathrm{free}(\chi)} \times D) \cup \Sigma_{\mathrm{free}(\chi)}$, $\Delta = \Delta_P \cup \Delta_\omega$ where

- $\Delta_P = \{(q_I, ((\sigma,\theta), (|\theta|_X, |\theta|_Y)), q', q') \mid (\sigma,\theta) \in \Sigma_{\mathrm{free}(\chi)}, q' \in \{q_I, q_f\}\}$ and
- $\Delta_\omega = \{(q_f, (\sigma, \emptyset), q_f, q_f) \mid \sigma \in \Sigma\}$

and $C = \{(z_1, z_2) \mid \mathtt{c} + z_1 \leqslant_{\mathtt{fin}} \mathtt{d} + z_2\}$.[4] Then, one can easily show that $\mathcal{L}(\chi) = \mathcal{L}(\mathcal{A})$. ◀

▶ **Proposition 17.** *For every* $\omega\mathrm{MSO}{\bowtie}\mathrm{BAPA}$ *formula* φ *there is a PMTA* $\mathcal{A}$ *with* $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\varphi)$.

Proof. Let φ be an $\omega\mathrm{MSO}{\bowtie}\mathrm{BAPA}$ formula. By Theorem 10, we can assume that φ is in tree normal form, i.e., of the form $\exists X_1. \cdots \exists X_n. \bigvee_{i=1}^k (\varphi_i \wedge \bigwedge_{j=1}^{l_i} \chi_{i,j})$, where φ_i are plain MSO sentences and the $\chi_{i,j}$ are (unnegated) Parikh constraints. The proof of the statement is an induction on the (now restricted) structure of φ using the well-known recognizability of MSO sentences [52], Lemma 16, and Proposition 14. ◀

The characterization obtained through Proposition 15 and Proposition 17 also provides an answer to the open problem posed by the authors in [35, 34] to find a logical characterization for their *reachability-regular Parikh automata* (RRPA) on words: in the usual way, our tree automata can simulate word automata (by embedding words in particular trees) and it is not too hard to see that the word version of PMTA is expressively equivalent to RRPA (details can be found in the appendix). Finally, by a routine inspection of the corresponding proofs we easily observe that our logical characterization also applies to the word setting.

Deciding Emptiness of Parikh-Muller Tree Automata

Our proof of decidability (and complexity) of the emptiness problem of PMTA rests on the respective results for the two components it combines, MTA and PTA. Thus, let us first recall the definition of Parikh tree automata [41, 40], slightly adjusted to our setting.

▶ **Definition 18** (Parikh tree automaton [42]). *Let* Σ *be an alphabet, let* $s \ge 1$, *and let* $D \subseteq \mathbb{N}^s$ *be finite. A* Parikh tree automaton *(PTA) is a tuple* $\mathcal{A} = (Q, \Sigma \times D, \delta, q_I, F, C)$ *where* Q *is a finite set of states,* $\delta \subseteq Q \times (\Sigma \times D) \times Q \times Q$ *is the* transition relation, q_I *is the* initial state, $F \subseteq Q$ *is a set of* final states, *and* $C \subseteq \mathbb{N}^s$ *is a semilinear set.*[5] *Given a finite tree* $\xi \in T_{\Sigma\times D}$, *a* run *of* $\mathcal{A}$ *on* ξ *is a tree* $\kappa_\xi \in T_Q$ *with* $\mathrm{pos}(\kappa_\xi) = \{\varepsilon\} \cup \{ui \mid u \in \mathrm{pos}(\xi), i \in \{0,1\}\}$ *and* $\kappa(\varepsilon) = q_I$ *that respects the transition relation of* $\mathcal{A}$. *The run* κ_ξ *is said to be* accepting *if* $\Psi(\xi) \in C$ *and* $\kappa_\xi(u) \in F$ *for each leaf* $u \in \mathrm{pos}(\kappa_\xi) \setminus \mathrm{pos}(\xi)$; *we denote the set of all accepting runs of* $\mathcal{A}$ *on* ξ *by* $\mathrm{Run}_\mathcal{A}(\xi)$. *Finally, the tree language of* $\mathcal{A}$, *denoted* $\mathcal{L}(\mathcal{A})$, *is the set*

$$\mathcal{L}(\mathcal{A}) := \{\xi \in T_\Sigma \mid \exists \xi' \in T_{\Sigma\times D} \text{ with } \mathrm{Run}_\mathcal{A}(\xi') \ne \emptyset \text{ and } (\xi')_\Sigma = \xi\}.$$

---

[4] Note that by Lemma 11 we can use this description for a semilinear set.

[5] We note that the PTAs defined in [42] were total, i.e., δ is a function of type $Q \times (\Sigma \times D) \to \mathcal{P}(Q \times Q)$. Each PTA as defined here can be made total by using an additional sink state.

It was shown in [40] that non-emptiness is decidable for PTA. The exact complexity can be obtained by adopting [28, Proposition III.2.] to the tree setting. This ultimately enables us to establish the desired result for our automaton model.

▶ **Proposition 19** (based on [40, 28]). *Given a PTA $\mathcal{A}$, deciding $\mathcal{L}(\mathcal{A}) \neq \emptyset$ is NP-complete.*

▶ **Theorem 20.** *Given a PMTA $\mathcal{A}$, deciding $\mathcal{L}(\mathcal{A}) \neq \emptyset$ is PSPACE-complete.*

Proof (sketch). Let $\mathcal{A} = (Q, \Xi, q_I, \Delta, \mathcal{F}, C)$ be a PMTA with $Q = Q_P \cup Q_\omega \cup \{q_I\}$, $\Xi = (\Sigma \times D) \cup \Sigma$, and $\Delta = \Delta_P \cup \Delta_\omega$. We observe that each tree in the language of $\mathcal{A}$ can be seen as some finite tree over $\Sigma \times D$ (on which the Parikh constraint is tested), having infinite trees from T_Σ attached to all its leafs. This allows us to reduce PMTA non-emptiness testing to deciding non-emptiness of Muller tree automata and Parikh tree automata. To this end, consider

- the Muller tree automaton $\mathcal{A}_{q_I} = (Q_\omega \cup \{q_I\}, \Sigma, q_I, \Delta_\omega, \mathcal{F})$,
- the Muller tree automata $\mathcal{A}_q = (Q_\omega, \Sigma, q, \Delta_\omega, \mathcal{F})$ for all $q \in Q_\omega$, and
- the Parikh tree automaton $\mathcal{A}_P = (Q, \Sigma \times D, q_I, \Delta_P, F_P, C)$ with $F_P = \{q \in Q_\omega \mid \mathcal{L}(\mathcal{A}_q) \neq \emptyset\}$.

As deciding $\mathcal{L}(\mathcal{A}_q) \neq \emptyset$ is PSPACE-complete [52, 39], $\mathcal{A}_P$ can be constructed in PSPACE and, by Proposition 19, its non-emptiness can be decided in NP. Thus, the overall PSPACE complexity follows from the observation that $\mathcal{L}(\mathcal{A}) \neq \emptyset$ iff $\mathcal{L}(\mathcal{A}_{q_I}) \neq \emptyset$ or $\mathcal{L}(\mathcal{A}_P) \neq \emptyset$. ◀

▶ **Corollary 21.** *Satisfiability of ωMSO⋈BAPA over labeled infinite binary trees is decidable.*

7 Decidability over Tree-Interpretable Classes of Structures

Finally, we lift the obtained decidability result for labeled trees to much more general classes of structures, leveraging the well-known technique of *MSO-interpretations* (also referred to as MSO-transductions or MSO-definable functions in the literature [1, 19, 25, 20, 22]).

▶ **Definition 22** (MSO-Interpretation). *Given two signatures $\mathbb{S}$ and $\mathbb{S}'$, an MSO-interpretation is a sequence $\mathcal{I} = (\varphi_{\mathrm{Dom}}(x), (\varphi_{\mathsf{c}}(x))_{\mathsf{c} \in \mathbb{S}_{\mathbf{C}}}, (\varphi_{\mathsf{Q}}(x_1, \ldots, x_{ar(\mathsf{Q})}))_{\mathsf{Q} \in \mathbb{S}_{\mathbf{P}}})$ of MSO-formulae over $\mathbb{S}'$ (with free variables as indicated). We identify $\mathcal{I}$ with the partial function satisfying $\mathcal{I}(\mathfrak{A}) = \mathfrak{B}$ for an $\mathbb{S}'$-structure $\mathfrak{A}$ and an $\mathbb{S}$-structure $\mathfrak{B}$ if $\{a \in A \mid \mathfrak{A}, \{x \mapsto a\} \models \varphi_{\mathrm{Dom}}(x)\} = B$ as well as $\{a \in B \mid \mathfrak{A}, \{x \mapsto a\} \models \varphi_{\mathsf{c}}(x)\} = \{\mathsf{c}^{\mathfrak{B}}\}$ for every $\mathsf{c} \in \mathbb{S}_{\mathbf{C}}$, and, for every $\mathsf{Q} \in \mathbb{S}_{\mathbf{P}}$, we have $\mathsf{Q}^{\mathfrak{B}} = \{(a_1, \ldots, a_{ar(\mathsf{Q})}) \in B^{ar(\mathsf{Q})} \mid \mathfrak{A}, \{x_i \mapsto a_i\}_{1 \leq i \leq ar(\mathsf{Q})} \models \varphi_{\mathsf{Q}}(x_1, \ldots, x_{ar(\mathsf{Q})})\}$. For a class $\mathcal{S}$ of $\mathbb{S}'$-structures, let $\mathcal{I}(\mathcal{S}) \coloneqq \{\mathfrak{B} \mid \mathcal{I}(\mathfrak{A}) = \mathfrak{B}, \mathfrak{A} \in \mathcal{S}\}$. A class $\mathcal{T}$ of $\mathbb{S}$-structures is tree-interpretable, if it coincides with $\mathcal{I}(T_\Sigma^\omega)$ for some Σ and corresponding MSO-interpretation $\mathcal{I}$.*

The key insight for our result is that the well-known rewritability of MSO formulae under MSO-interpretations can be lifted to ωMSO⋈BAPA without much effort.

▶ **Lemma 23.** *Let $\mathcal{I}$ be an MSO-interpretation. Then, for every ωMSO⋈BAPA sentence φ over $\mathbb{S}$ one can compute an ωMSO⋈BAPA sentence $\varphi^{\mathcal{I}}$ over $\mathbb{S}'$ satisfying $\mathfrak{A} \models \varphi^{\mathcal{I}} \Longleftrightarrow \mathfrak{B} \models \varphi$ for every $\mathbb{S}'$-structure $\mathfrak{A}$ and $\mathbb{S}$-structure $\mathfrak{B}$ with $\mathcal{I}(\mathfrak{A}) \cong \mathfrak{B}$.*

This insight can be used to show that decidability is propagated through MSO-interpretations, and thus can be guaranteed for all tree-interpretable classes, thanks to Corollary 21.

▶ **Theorem 24.** *Let $\mathcal{S}$ be a class of structures over which satisfiability of ωMSO⋈BAPA is decidable, let $\mathcal{I}$ be an MSO-interpretation. Then satisfiability of ωMSO⋈BAPA over $\mathcal{I}(\mathcal{S})$ is decidable as well. In particular, ωMSO⋈BAPA is decidable over any tree-interpretable class.*

This result allows us, in one go, to harvest several decidability results, as tree-interpretability is able to capture classes of (finite or countable) structures whose treewidth [53], cliquewidth [27, 22, 21, 36], or partitionwidth [10, 11, 26] is bounded by some value $k \in \mathbb{N}$.

▶ **Corollary 25.** *Given a signature* $\mathbb{S}$, *satisfiability of* ωMSO⋈BAPA *is decidable over the classes of finite or countable* $\mathbb{S}$-*structures of bounded treewidth, cliquewidth, and partitionwidth.*

8 Incorporating Two-Variable-Logics without Width Restrictions

Corollary 25 constitutes a strong decidability result, also in view of the fact that lifting the width restriction immediately leads to undecidability even for much weaker logics like FO. A feasible way to nevertheless relax this restriction without putting decidability at risk and yet maintaining all the expressive power of ωMSO⋈BAPA is to "couple" it with another logic $\mathbb{L}$ whose satisfiability problem is decidable over arbitrary structures. Then, one considers sentences $\varphi \wedge \psi$, where φ is an ωMSO⋈BAPA sentence while ψ is an $\mathbb{L}$-sentence, and asks for models whose reduct to the signature of φ adheres to the width restriction. That way, signature elements of ψ not occurring in φ can "behave freely" and are not subject to the imposed width constraint.[6] Such a "coupling" of ωMSO⋈BAPA and $\mathbb{L}$ can be made more or less "tight" depending on the arity of the predicates allowed to be shared between φ and ψ.

We can show that a decidable coupling with shared unary predicates can be done for $\mathbb{L} = \text{FO}^2_{\text{Pres}}$ [7], an expressive extension of 2-variable first-order logic by Presburger-like counting quantifiers of the form $\exists^S$, where $S \subseteq \mathbb{N} \cup \{\infty\}$ is an ultimately periodic set from $\mathbb{N} \cup \{\infty\}$ with the semantics defined by $\mathfrak{A}, \nu \models \exists^S x.\varphi$ iff $|\{a \in A \mid \mathfrak{A}, \nu_{x \mapsto a} \models \varphi\}| \in S$. $\text{FO}^2_{\text{Pres}}$ subsumes the prominent counting 2-variable first-order fragment C^2 [32], but goes beyond first-order logic. Its satisfiability problem was shown to be decidable only recently [7].

▶ **Theorem 26.** *Let* w *be any of treewidth, cliquewidth, or partitionwidth, and let* $n \in \mathbb{N}$. *Let* $\mathbb{S}_a$ *and* $\mathbb{S}_b$ *be signatures whose only joint elements are unary predicates. Then the following problem is decidable: Given a* ωMSO⋈BAPA *sentence* φ *over* $\mathbb{S}_a$ *and a* $\text{FO}^2_{\text{Pres}}$ *sentence* ψ *over* $\mathbb{S}_b$, *does there exist a countable* $\mathbb{S}_a \cup \mathbb{S}_b$-*structure* $\mathfrak{C}$ *satisfying* $w(\mathfrak{C}|_{\mathbb{S}_a}) \leq n$ *and* $\mathfrak{C} \models \varphi \wedge \psi$.

In a nutshell, this result is obtained by exploiting the fact that, for every $\text{FO}^2_{\text{Pres}}$ formula ψ over $\mathbb{S}_b$, one can construct a ωMSO⋈BAPA formula ψ' over the purely unary signature $\mathbb{S}_a \cap \mathbb{S}_b$ that is satisfied by precisely those $\mathbb{S}_a$-structures that are "$\mathbb{S}_a \cap \mathbb{S}_b$-compatible" with some model of ψ. Consequently, the ωMSO⋈BAPA formula $\varphi \wedge \psi'$ over $\mathbb{S}_a$ is such that for any of its models $\mathfrak{A}$ one finds a "$\mathbb{S}_a \cap \mathbb{S}_b$-compatible" model $\mathfrak{B}$ of ψ. Then, superimposing $\mathfrak{A}$ and $\mathfrak{B}$ would yield a model $\mathfrak{C}$ of $\varphi \wedge \psi$, which by construction satisfies $w(\mathfrak{C}|_{\mathbb{S}_a}) = w(\mathfrak{A})$. Consequently, to solve the decision problem of Theorem 26, it suffices to check if the ωMSO⋈BAPA formula $\varphi \wedge \psi'$ has a model $\mathfrak{A}$ satisfying $w(\mathfrak{A}|_{\mathbb{S}_a}) \leq n$ which is decidable by Corollary 25. We note that the extended arithmetic capabilities of ωMSO⋈BAPA are essential for this result, as ψ' needs to encode linear inequalities over counts of realized atomic 1-types.

9 Showcase: Decidability of Tame Satisfiability of the Fully Enriched μ-Calculus with Global Presburger Counting

An important and practically relevant class of expressive logical formalisms, which play a pivotal role in logic-based knowledge representation and verification, is obtained from

---

[6] We refer to Kotek et al. [43] for a result that is similar in spirit, establishing decidability of finite satisfiability of treewidth-bounded MSO_2 coupled with C^2.

variations and extensions of propositional modal logics [8, 9] and description logics [4, 55]. This class contains most ontology languages as well as PDL [29], CTL* [24], the propositional modal μ-calculus [44] and their extensions. Many of these logics exhibit some limited local counting capabilities [58], but recently, there has been an increased interest in accommodating more advanced arithmetic constraints [23, 48, 2, 5], including *global constraints* [3, 54] expressing statistical information such as "more than 50% of the state space's final states are successful".

We will demonstrate the usefulness of ωMSO$\bowtie$BAPA for establishing decidability results at the example of adding global Presburger constraints to the *fully enriched μ-calculus*, a very powerful formalism used in verification. We first introduce syntax and semantics.[7]

▶ **Definition 27.** *Given a signature* $\mathbb{S} = \mathbb{S}_{\mathbf{C}} \cup \mathbb{S}_{\mathbf{P},1} \cup \mathbb{S}_{\mathbf{P},2}$ *of constants, unary predicates and binary predicates, the formulas of the fully enriched μ-calculus (FEμ) are defined by*

$$\varphi ::= \ \mathbf{true} \mid \mathbf{false} \mid X \mid \mathtt{c} \mid \neg\mathtt{c} \mid \mathtt{P} \mid \neg\mathtt{P} \mid \varphi \wedge \varphi' \mid \varphi \vee \varphi' \mid \langle n, \alpha \rangle \varphi \mid [n, \alpha]\varphi \mid \mu X.\varphi \mid \nu X.\varphi$$

where X is a set variable from some countable set $\mathbf{V}_{\mathrm{set}}$, $\mathtt{P} \in \mathbb{S}_{\mathbf{P},1}$, $n \in \mathbb{N}$ and α has the form $\mathtt{R}$ or $\mathtt{R}^-$ for some $\mathtt{R} \in \mathbb{S}_{\mathbf{P},2}$. For ease of presentation, we assume positive normal form. Given a structure $\mathfrak{A}$ and a set variable assignment $\nu : \mathbf{V}_{\mathrm{set}} \to 2^A$, the semantics $[\![\varphi]\!]_\nu^{\mathfrak{A}} \subseteq A$ of formulae φ is defined by the following function (stipulating $(\mathtt{R}^-)^{\mathfrak{A}} = \{(a, a') \mid (a', a) \in \mathtt{R}^{\mathfrak{A}}\}$):

$$
\begin{array}{llll}
\mathbf{true} \mapsto A & X \mapsto \nu(X) & \mathtt{c} \mapsto \{\mathtt{c}^{\mathfrak{A}}\} & \mathtt{P} \mapsto \mathtt{P}^{\mathfrak{A}} & \varphi \wedge \varphi' \mapsto [\![\varphi]\!]_\nu^{\mathfrak{A}} \cap [\![\varphi']\!]_\nu^{\mathfrak{A}} \\
\mathbf{false} \mapsto \emptyset & & \neg\mathtt{c} \mapsto A \setminus \{\mathtt{c}^{\mathfrak{A}}\} & \neg\mathtt{P} \mapsto A \setminus \mathtt{P}^{\mathfrak{A}} & \varphi \vee \varphi' \mapsto [\![\varphi]\!]_\nu^{\mathfrak{A}} \cup [\![\varphi']\!]_\nu^{\mathfrak{A}}
\end{array}
$$

$$\langle n, \alpha \rangle \varphi \mapsto \{a \mid |\{\alpha^{\mathfrak{A}} \cap (\{a\} \times [\![\varphi]\!]_\nu^{\mathfrak{A}})\}| \geq n\} \qquad \mu X.\varphi \mapsto \bigcap \{A' \subseteq A \mid [\![\varphi]\!]_{\nu_{X \mapsto A'}}^{\mathfrak{A}} \subseteq A'\}$$

$$[n, \alpha]\varphi \mapsto \{a \mid |\{\alpha^{\mathfrak{A}} \cap (\{a\} \times (A \setminus [\![\varphi]\!]_\nu^{\mathfrak{A}}))\}| \leq n\} \qquad \nu X.\varphi \mapsto \bigcup \{A' \subseteq A \mid A' \subseteq [\![\varphi]\!]_{\nu_{X \mapsto A'}}^{\mathfrak{A}}\}$$

A FEμ formula is closed *if all occurrences of set variables are in the scope of some μ or ν. A* global FEμ Presburger constraint *is a Parikh constraint (cf. Definition 7), where all set variables have been replaced by closed FEμ formulae. Given a set Π of global FEμ Presburger constraints, we let $\mathfrak{A} \models \Pi$ if for every element of Π, replacing each of its closed FEμ formulae ψ by $[\![\psi]\!]_\emptyset^{\mathfrak{A}}$ produces a statement valid in $\mathfrak{A}$. A closed FEμ formula φ is satisfiable wrt. Π if there is some structure $\mathfrak{A} \models \Pi$ with $[\![\varphi]\!]_\emptyset^{\mathfrak{A}} \neq \emptyset$, in which case we call $\mathfrak{A}$ a model of (φ, Π).*

In fact, unrestricted satisfiability in FEμ (even without Presburger constraints) is undecidable [13]. Decidability can be regained, however, when restricting to *tame structures*, also commonly known as "quasi-forests" [15, 12, 16, 6].

▶ **Definition 28** (tame structures). *Let $\mathbb{S} = \mathbb{S}_{\mathbf{C}} \cup \mathbb{S}_{\mathbf{P},1} \cup \mathbb{S}_{\mathbf{P},2}$ be a signature as above. A* tame *structure $\mathfrak{A}$ over $\mathbb{S}$ is a countable structure such that, for some finite set Roots,*
- *the domain A of $\mathfrak{A}$ is a forest, i.e., a prefix-closed subset of $\{rw \mid r \in Roots, w \in \mathbb{N}^*\}$,*
- *the roots coincide with the named elements, i.e., $Roots = \{\mathtt{a}^{\mathfrak{A}} \mid \mathtt{a} \in \mathbb{S}_{\mathbf{C}}\}$, and*
- *for every $a, a' \in A$ with $(a, a') \in \mathtt{R}^{\mathfrak{A}}$ for some $\mathtt{R} \in \mathbb{S}_{\mathbf{P},2}$, either (i) $\{a, a'\} \cap Roots \neq \emptyset$, or (ii) $a = a'$, or (iii) a is a child of a', or (iv) a' is a child of a.*

A logic has the tame model property *if every satisfiable formula φ has a model that is tame over the signature used by φ. The* tame satisfiability *problem consists in deciding if a given formula has a tame model.*

While the restriction to tame structures may seem somewhat arbitrary at first, it is well justified: three maximal decidable sublogics of FEμ have the tame-model-property [12], in

---

[7] For brevity and coherence, we slightly adjust the syntax and use classical model-theoretic semantics (structures with unary and binary predicates) instead of the original one of modal logic (Kripke structures with propositional variables and programs), as the two are well known to be equivalent.

which case satisfiability over arbitrary structures and tame structures coincide. Also, the structural restriction has some plausibility from a transition system perspective in that one distinguishes between a finite set of "named" states with arbitrary transitions between them and potentially infinitely many "anonymous" states with more restricted access. It is easy to see that all tame structures over $\mathbb{S} = \mathbb{S}_\mathbf{C} \cup \mathbb{S}_\mathbf{P}$ have a treewidth not larger than $|\mathbb{S}_\mathbf{C}| + 1$.

▶ **Theorem 29.** *The tame satisfiability problem of the fully enriched μ-calculus with global Presburger constraints is decidable.*

Proof (sketch). Let $\mathbb{S}$ be a finite signature, φ a closed FEμ formula over $\mathbb{S}$, and Π a finite set of global FEμ Presburger constraints. Being a tame structure over $\mathbb{S}$ can be expressed by an MSO sentence ψ_{tame}. We define a translation trans_x mapping closed FEμ formulae to ωMSO⋈BAPA formulae with free variable x such that $\mathfrak{A}, \{x \mapsto a\} \models \text{trans}_x(\varphi)$ iff $a \in [\![\varphi]\!]_\emptyset^\mathfrak{A}$. Based on this, we exhibit another translation trans, which maps global FEμ Presburger constraints to equivalent ωMSO⋈BAPA sentences. Then, tame satisfiability of (φ, Π) corresponds to satisfiability of the ωMSO⋈BAPA sentence $\psi_{\text{tame}} \wedge \exists x.\text{trans}_x(\varphi) \wedge \bigwedge \text{trans}(\Pi)$ over all countable structures of treewidth $\leq |\mathbb{S}_\mathbf{C}| + 1$, which is decidable by Corollary 25. ◀

Thanks to the expressive power of FEμ, the above result transfers to numerous other prominent logics (and their fragments), including PDL and CTL* as well as the description logics $\mu\mathcal{ALCOIQ}$ and $\mathcal{ALCOIQ}^{\text{reg}}$ [14], for all of which tame satisfiability is thus decidable even in the presence of global Presburger constraints. The argument easily extends to the description logic $\mathcal{ZOIQ}$ [16], adding Boolean combinations of binary predicates (programs).

10   Conclusion

We have proposed ωMSO⋈BAPA, a logic with a high combined structural and arithmetic expressivity, subsuming and properly extending existing popular formalisms for either purpose. We have established decidability of the satisfiability of ωMSO⋈BAPA formulae over arbitrary tree-interpretable classes of structures. A key role is played by Parikh-Muller Tree Automata, a novel type of automaton over labeled infinite binary trees with decidable emptiness.

For improving readability and succinctness, the syntax of our formalism could be extended by "comprehension expressions": set terms of the form $\{x \mid \psi\}$ with $x \in \mathbf{V}_{\text{ind}}$ and $\psi \in \mathbf{F}$, whose semantics is straightforwardly defined by $\{x \mid \psi\}^{\mathfrak{A}, \nu} = \{a \in A \mid \mathfrak{A}, \nu_{x \mapsto a} \models \psi\}$. E.g., this allows us to write $2\,\#\{x \mid \exists y.\mathtt{R}(x, y)\} = 3\,\#\{y \mid \exists x.\mathtt{R}(x, y)\}$ rather than the more unwieldy

$$\exists V_1.(\forall x.V_1(x) \Leftrightarrow \exists y.\mathtt{R}(x, y)) \wedge \exists V_2.(\forall y.V_2(y) \Leftrightarrow \exists x.\mathtt{R}(x, y)) \wedge 2\,\#V_1 = 3\,\#V_2.$$

Note that comprehension expressions do not increase expressivity; they can be removed from a formula φ yielding an equivalent formula φ' as follows: Let χ be the largest subformula of φ that contains the expression $\{x \mid \psi\}$ but no quantifiers binding any of the free variables of ψ. Then, obtain φ' from φ by replacing χ by χ', where $\chi' := \exists Z.(\forall x.Z(x) \Leftrightarrow \psi) \wedge \chi[\{x \mid \psi\} \mapsto Z]$. ωMSO⋈BAPA membership of such extended formulae can then be decided based on their "purified" variant,[8] or by means of an elaborately refined analysis of variable interactions.

Concluding, we are quite confident that this paper's findings and techniques will prove useful as a generic tool for establishing decidability results for formalisms from various areas of computer science such as knowledge representation or verification. That said, in view of the non-elementary blow-ups abounding in our methods, we concede that they are unlikely to be helpful in more fine-grained complexity analyses, once decidability is established.

---

[8] The described removal technique is optimized toward producing formulae in ωMSO⋈BAPA.

References

1 Stefan Arnborg, Jens Lagergren, and Detlef Seese. Problems easy for tree-decomposable graphs (extended abstract). In Timo Lepistö and Arto Salomaa, editors, *15th International Colloquium on Automata, Languages and Programming (ICALP 1988)*, volume 317 of *LNCS*, pages 38–51. Springer, 1988. `doi:10.1007/3-540-19488-6_105`.

2 Franz Baader, Bartosz Bednarczyk, and Sebastian Rudolph. Satisfiability and query answering in description logics with global and local cardinality constraints. In Giuseppe De Giacomo, Alejandro Catalá, Bistra Dilkina, Michela Milano, Senén Barro, Alberto Bugarín, and Jérôme Lang, editors, *24th European Conference on Artificial Intelligence, (ECAI 2020) - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020)*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 616–623. IOS Press, 2020. `doi:10.3233/FAIA200146`.

3 Franz Baader, Martin Buchheit, and Bernhard Hollunder. Cardinality restrictions on concepts. *Artif. Intell.*, 88(1-2):195–213, 1996. `doi:10.1016/S0004-3702(96)00010-0`.

4 Franz Baader, Ian Horrocks, Carsten Lutz, and Uli Sattler. *An Introduction to Description Logic*. Cambridge University Press, 2017. `doi:10.1017/9781139025355`.

5 Bartosz Bednarczyk, Maja Orlowska, Anna Pacanowska, and Tony Tan. On classical decidable logics extended with percentage quantifiers and arithmetics. In Mikolaj Bojanczyk and Chandra Chekuri, editors, *41st IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2021)*, volume 213 of *LIPIcs*, pages 36:1–36:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. `doi:10.4230/LIPIcs.FSTTCS.2021.36`.

6 Bartosz Bednarczyk and Sebastian Rudolph. Worst-case optimal querying of very expressive description logics with path expressions and succinct counting. In Sarit Kraus, editor, *28st International Joint Conference on Artificial Intelligence (IJCAI 2019)*, pages 1530–1536. ijcai.org, 2019. `doi:10.24963/ijcai.2019/212`.

7 Michael Benedikt, Egor V. Kostylev, and Tony Tan. Two variable logic with ultimately periodic counting. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming (ICALP 2020)*, volume 168 of *LIPIcs*, pages 112:1–112:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. `doi:10.4230/LIPIcs.ICALP.2020.112`.

8 Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*, volume 53 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2001. `doi:10.1017/CBO9781107050884`.

9 Patrick Blackburn, Johan F.A.K. van Benthem, and Frank Wolter, editors. *Handbook of Modal Logic*, volume 3 of *Studies in logic and practical reasoning*. North-Holland, 2007.

10 Achim Blumensath. *Structures of bounded partition width*. PhD thesis, RWTH Aachen University, Germany, 2003. URL: `http://sylvester.bth.rwth-aachen.de/dissertationen/2003/256/index.htm`.

11 Achim Blumensath. A model-theoretic characterisation of clique width. *Annals of Pure and Applied Logic*, 142(1-3):321–350, 2006. `doi:10.1016/j.apal.2006.02.004`.

12 Piero A. Bonatti, Carsten Lutz, Aniello Murano, and Moshe Y. Vardi. The complexity of enriched mu-calculi. *Log. Methods Comput. Sci.*, 4(3), 2008. `doi:10.2168/LMCS-4(3:11)2008`.

13 Piero A. Bonatti and Adriano Peron. On the undecidability of logics with converse, nominals, recursion and counting. *Artif. Intell.*, 158(1):75–96, 2004. `doi:10.1016/j.artint.2004.04.012`.

14 Diego Calvanese and Giuseppe De Giacomo. *Expressive Description Logics*, page 193–236. Cambridge University Press, 2 edition, 2007. `doi:10.1017/CBO9780511711787.007`.

15 Diego Calvanese, Thomas Eiter, and Magdalena Ortiz. Answering regular path queries in expressive description logics: An automata-theoretic approach. In *22nd Conference on Artificial Intelligence (AAAI 2007)*, pages 391–396. AAAI Press, 2007. URL: `http://www.aaai.org/Library/AAAI/2007/aaai07-061.php`.

16 Diego Calvanese, Thomas Eiter, and Magdalena Ortiz. Regular path queries in expressive description logics with nominals. In Craig Boutilier, editor, *21st International*

Joint Conference on Artificial Intelligence (IJCAI 2009), pages 714–720, 2009. URL: `http://ijcai.org/Proceedings/09/Papers/124.pdf`.

17 Bruno Courcelle. The monadic second-order logic of graphs, II: Infinite graphs of bounded width. *Mathematical Systems Theory*, 21(1):187–221, 1988. `doi:10.1007/BF02088013`.

18 Bruno Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990. `doi:10.1016/0890-5401(90)90043-H`.

19 Bruno Courcelle. The monadic second-order logic of graphs V. On closing the gap between definability and recognizability. *Theoretical Computer Science*, 80(2):153–202, 1991. `doi:10.1016/0304-3975(91)90387-H`.

20 Bruno Courcelle. Monadic second-order definable graph transductions: A survey. *Theoretical Computer Science*, 126(1):53–75, 1994. `doi:10.1016/0304-3975(94)90268-2`.

21 Bruno Courcelle. Clique-width of countable graphs: A compactness property. *Discrete Mathematics*, 276(1-3):127–148, 2004. `doi:10.1016/S0012-365X(03)00303-0`.

22 Bruno Courcelle and Joost Engelfriet. *Graph Structure and Monadic Second-Order Logic – A Language-Theoretic Approach*, volume 138 of *Encyclopedia of mathematics and its applications*. Cambridge University Press, 2012. `doi:10.1017/CBO9780511977619`.

23 Stéphane Demri and Denis Lugiez. Complexity of modal logics with presburger constraints. *J. Appl. Log.*, 8(3):233–252, 2010. `doi:10.1016/j.jal.2010.03.001`.

24 E. Allen Emerson and Joseph Y. Halpern. "sometimes" and "not never" revisited: On branching versus linear time. In John R. Wright, Larry Landweber, Alan J. Demers, and Tim Teitelbaum, editors, *10th Annual ACM Symposium on Principles of Programming Languages (POPL 1983)*, pages 127–140. ACM Press, 1983. `doi:10.1145/567067.567081`.

25 Joost Engelfriet. A characterization of context-free NCE graph languages by monadic second-order logic on trees. In Hartmut Ehrig, Hans-Jörg Kreowski, and Grzegorz Rozenberg, editors, *4th International Workshop on Graph-Grammars and Their Application to Computer Science (Graph Grammars 1990)*, volume 532 of *LNCS*, pages 311–327. Springer, 1990. `doi:10.1007/BFb0017397`.

26 Thomas Feller, Tim S. Lyon, Piotr Ostropolski-Nalewaja, and Sebastian Rudolph. Decidability of querying first-order theories via countermodels of finite width. 2023. `arXiv:2304.06348`.

27 Thomas Feller, Tim S. Lyon, Piotr Ostropolski-Nalewaja, and Sebastian Rudolph. Finite-cliquewidth sets of existential rules: Toward a general criterion for decidable yet highly expressive querying. In *26th International Conference on Database Theory (ICDT 2023)*, LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023.

28 Diego Figueira and Leonid Libkin. Path Logics for Querying Graphs: Combining Expressiveness and Efficiency. In *30th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2015)*, pages 329–340, 2015. `doi:10.1109/LICS.2015.39`.

29 Michael J. Fischer and Richard E. Ladner. Propositional dynamic logic of regular programs. *J. Comput. Syst. Sci.*, 18(2):194–211, 1979. `doi:10.1016/0022-0000(79)90046-1`.

30 Seymour Ginsburg and Edwin H. Spanier. Bounded Algol-Like Languages. *Transactions of the American Mathematical Society*, 113(2):333, 1964. `doi:10.2307/1994067`.

31 Seymour Ginsburg and Edwin H. Spanier. Semigroups, Presburger formulas, and languages. *Pacific Journal of Mathematics*, 16(2):285–296, 1966. `doi:10.2140/pjm.1966.16.285`.

32 Erich Grädel, Martin Otto, and Eric Rosen. Two-variable logic with counting is decidable. In *12th Annual IEEE Symposium on Logic in Computer Science (LICS'97)*, pages 306–317. IEEE Computer Society, 1997. `doi:10.1109/LICS.1997.614957`.

33 Mario Grobler, Leif Sabellek, and Sebastian Siebertz. Parikh Automata on Infinite Words, 2023. `arXiv:2301.08969`.

34 Mario Grobler, Leif Sabellek, and Sebastian Siebertz. Remarks on Parikh-recognizable omega-languages, 2023. `arXiv:2307.07238`.

35 Mario Grobler and Sebastian Siebertz. Büchi-like characterizations for Parikh-recognizable omega-languages, 2023. `arXiv:2302.04087`.

36   Martin Grohe and György Turán. Learnability and definability in trees and similar structures. *Theory of Computing Systems*, 37(1):193–220, 2004. `doi:10.1007/s00224-003-1112-8`.

37   Shibashis Guha, Ismaël Jecker, Karoliina Lehtinen, and Martin Zimmermann. Parikh automata over infinite words. In Anuj Dawar and Venkatesan Guruswami, editors, *42nd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2022)*, volume 250 of *LIPIcs*, pages 40:1–40:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. `doi:10.4230/LIPIcs.FSTTCS.2022.40`.

38   Matthew Hague and Anthony Widjaja Lin. Synchronisation- and Reversal-Bounded Analysis of Multithreaded Programs with Counters. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, P. Madhusudan, and Sanjit A. Seshia, editors, *Computer Aided Verification (CAV 2012)*, volume 7358 of *LNCS*, pages 260–276. Springer, 2012. `doi:10.1007/978-3-642-31424-7_22`.

39   Paul Hunter and Anuj Dawar. Complexity Bounds for Regular Games. In Joanna Jędrzejowicz and Andrzej Szepietowski, editors, *Mathematical Foundations of Computer Science (MFCS 2005)*, LNCS, pages 495–506. Springer, 2005. `doi:10.1007/11549345_43`.

40   Felix Klaedtke. *Automata-based decision procedures for weak arithmetics*. PhD thesis, University of Freiburg, Freiburg im Breisgau, Germany, 2004. URL: `http://freidok.ub.uni-freiburg.de/volltexte/1439/index.html`.

41   Felix Klaedtke and Harald Rueß. Parikh automata and monadic second-order logics with linear cardinality constraints. Technical Report 177, Albert-Ludwigs-Universität Freiburg, 2002. (revised version).

42   Felix Klaedtke and Harald Rueß. Monadic second-order logics with cardinalities. In Jos C. M. Baeten, Jan Karel Lenstra, Joachim Parrow, and Gerhard J. Woeginger, editors, *Automata, Languages and Programming, 30th International Colloquium (ICALP 2003)*, volume 2719 of *LNCS*, pages 681–696. Springer, 2003. `doi:10.1007/3-540-45061-0_54`.

43   Tomer Kotek, Helmut Veith, and Florian Zuleger. Monadic Second Order Finite Satisfiability and Unbounded Tree-Width. In Jean-Marc Talbot and Laurent Regnier, editors, *25th EACSL Annual Conference on Computer Science Logic (CSL 2016)*, volume 62 of *LIPIcs*, pages 13:1–13:20. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016. `doi:10.4230/LIPIcs.CSL.2016.13`.

44   Dexter Kozen. Results on the propositional μ-calculus. In Mogens Nielsen and Erik Meineche Schmidt, editors, *Automata, Languages and Programming, 9th Colloquium (ICALP 1982)*, volume 140 of *LNCS*, pages 348–359. Springer, 1982. `doi:10.1007/BFb0012782`.

45   Viktor Kuncak, Huu Hai Nguyen, and Martin Rinard. An Algorithm for Deciding BAPA: Boolean Algebra with Presburger Arithmetic. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Dough Tygar, Moshe Y. Vardi, Gerhard Weikum, and Robert Nieuwenhuis, editors, *Automated Deduction (CADE 2005)*, volume 3632 of *LNCS*, pages 260–277. Springer, 2005. `doi:10.1007/11532231_20`.

46   Viktor Kuncak, Huu Hai Nguyen, and Martin Rinard. Deciding Boolean Algebra with Presburger Arithmetic. *Journal of Automated Reasoning*, 36(3):213–239, 2006. `doi:10.1007/s10817-006-9042-1`.

47   Aless Lasaruk and Thomas Sturm. Effective Quantifier Elimination for Presburger Arithmetic with Infinity. In Vladimir P. Gerdt, Ernst W. Mayr, and Evgenii V. Vorozhtsov, editors, *Computer Algebra in Scientific Computing (CASC 2009)*, volume 5743 of *LNCS*, pages 195–212. Springer, 2009. `doi:10.1007/978-3-642-04103-7_18`.

48   Yensen Limón, Edgard Benítez-Guerrero, Everardo Bárcenas, Guillermo Molero-Castillo, and Alejandro Velázquez-Mena. A satisfiability algorithm for the mu-calculus for trees with presburger constraints. In *7th International Conference in Software Engineering Research and Innovation (CONISOFT 2019)*, pages 72–79, 2019. `doi:10.1109/CONISOFT.2019.00020`.

49   Yuri V. Matiyasevich. *Hilbert's Tenth Problem*. Foundations of Computing. MIT Press, 1993.

50   Rohit J. Parikh. On Context-Free Languages. *Journal of the ACM*, 13(4):570–581, 1966. `doi:10.1145/321356.321364`.

51   Mojżesz Presburger. Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt. In *Comptes Rendus du I congrès de Mathématiciens des Pays Slaves*, pages 92–101, 1929.

52   Michael O. Rabin. Decidability of Second-Order Theories and Automata on Infinite Trees. *Transactions of the American Mathematical Society*, 141:1–35, 1969. `doi:10.2307/1995086`.

53   Neil Robertson and P.D Seymour. Graph minors. III. Planar tree-width. *Journal of Combinatorial Theory, Series B*, 36(1):49–64, 1984. `doi:https://doi.org/10.1016/0095-8956(84)90013-3`.

54   "Johann" Sebastian Rudolph. Presburger concept cardinality constraints in very expressive description logics - allegro sexagenarioso ma non ritardando. In Carsten Lutz, Uli Sattler, Cesare Tinelli, Anni-Yasmin Turhan, and Frank Wolter, editors, *Description Logic, Theory Combination, and All That - Essays Dedicated to Franz Baader on the Occasion of His 60th Birthday*, volume 11560 of *LNCS*, pages 542–561. Springer, 2019. `doi:10.1007/978-3-030-22102-7_25`.

55   Sebastian Rudolph. Foundations of description logics. In Axel Polleres, Claudia d'Amato, Marcelo Arenas, Siegfried Handschuh, Paula Kroner, Sascha Ossowski, and Peter F. Patel-Schneider, editors, *Lecture Notes of the 7th International Reasoning Web Summer School (RW'11)*, volume 6848 of *LNCS*, pages 76–136. Springer, 2011. `doi:10.1007/978-3-642-23032-5_2`.

56   Bruno Scarpellini. Complexity of Subcases of Presburger Arithmetic. *Transactions of the American Mathematical Society*, 284(1):203–218, 1984. `doi:10.2307/1999283`.

57   Wolfgang Thomas. Languages, Automata, and Logic. In Grzegorz Rozenberg and Arto Salomaa, editors, *Handbook of Formal Languages*, pages 389–455. Springer, 1997. `doi:10.1007/978-3-642-59126-6_7`.

58   Stephan Tobies. The complexity of reasoning with cardinality restrictions and nominals in expressive description logics. *J. Artif. Intell. Res.*, 12:199–217, 2000. `doi:10.1613/jair.705`.

59   Kumar Neeraj Verma, Helmut Seidl, and Thomas Schwentick. On the Complexity of Equational Horn Clauses. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Dough Tygar, Moshe Y. Vardi, Gerhard Weikum, and Robert Nieuwenhuis, editors, *Automated Deduction (CADE 2005)*, volume 3632 of *LNCS*, pages 337–352. Springer, 2005. `doi:10.1007/11532231_25`.

A   Mildly Extending ωMSO⋈BAPA Leads to Undecidability

Given infinite trees $\xi, \xi_1 \in T_\Sigma^\omega$ and a position $\varrho \in \mathrm{pos}(\xi)$, we denote by $\xi[\varrho \to \xi_1]$ the tree ξ' resulting from substituting ξ_1 at position ϱ into ξ, i.e., for each $\varrho' \in \mathrm{pos}(\xi')$ we obtain $\xi'(\varrho') = \xi_1(\varrho_1)$ if $\varrho' = \varrho\varrho_1$ for some $\varrho_1 \in \{0,1\}^*$ and $\xi'(\varrho') = \xi(\varrho')$ otherwise. We abbreviate $(\ldots(\xi[\varrho_1 \to \xi_1])\ldots)[\varrho_n \to \xi_n]$ by $\xi[\varrho_1 \to \xi_1, \ldots, \varrho_n \to \xi_n]$. Furthermore, we denote by $[\xi]_\varrho$ the subtree of ξ at position ϱ, i.e., $[\xi]_\varrho(\varrho') = \xi(\varrho\varrho')$ for each $\varrho' \in \{0,1\}^*$. We let $|\xi|_\sigma = |\{\varrho \in \{0,1\}^* \mid \xi(\varrho) = \sigma\}|$ for each $\sigma \in \Sigma$.

Given two numbers $m, n \in \mathbb{N}$ such that $m < 2^n$ we denote by $\langle m \rangle_{\mathrm{bin}}^n$ the binary representation of m using n digits (with the least significant digit on the right).

▶ **Proposition 6.** *For any positive Diophantine equation $\mathcal{D}$, satisfaction of $\varphi_\mathcal{D}$ over (finite or infinite) labeled trees coincides with solvability of $\mathcal{D}$. Consequently, satisfiability of the class of ωMSO·BAPA sentences of the shape $\varphi_\mathcal{D}$ is undecidable.*

Proof. Let $\mathcal{D} = (NV, M, (n_w)_{w \in M}, (m_w)_{w \in M})$ be a positive Diophantine equation with number variables $NV = \{\varkappa_1, \ldots, \varkappa_{\Bbbk}\}$

$\Rightarrow$: Assume that $\nu : NV \to \mathbb{N}$ is a solution of $\mathcal{D}$.

Let $\Sigma = M \cup NV$ and $\hat{\Sigma} = \{\hat{\alpha} \mid \alpha \in \Sigma\}$. We construct a model $\xi \in T_{\Sigma \cup \hat{\Sigma}}^\omega$ of $\varphi_\mathcal{D}$ with the following intuition: for each position ϱ in ξ labeled by w and all variables $\varkappa_{i_1}, \ldots, \varkappa_{i_l}$ with $w\varkappa_{i_j} \in M$ we pick up a level n in $[\xi]_\varrho$ with enough space to label in parallel nodes from $\varrho \cdot \{0,1\}^n$ by $w\varkappa_{i_j}$ – such that, for each $j \in [l]$, $w\varkappa_{i_j}$ occurs $\nu(\varkappa_{i_j})$ many times in $[\xi]_\varrho$. The nodes in between are labeled by the padding symbol $\hat{w}$.

Let $w \in M \cup NV$ and $\zeta_w \in T_{\{w,\hat{w}\}}$ such that $\zeta_w(\varepsilon) = w$ and $\zeta_w(\varrho) = \hat{w}$ for each $\varrho \in \{0,1\}^+$. Then $\xi = \xi_\varepsilon$ with ξ_ε recursively defined as follows. For the base case, let $w \in M \cup NV$ such that there is no $\varkappa \in NV$ with $w\varkappa \in M$. Then $\xi_w = \zeta_w$.

For the inductive case let $w \in M \cup NV$ such that there is at least one $\varkappa_i \in NV$ with $w\varkappa_i \in M \cup NV$. For every such $w \in M \cup NV$, let

- $m_w = \displaystyle\sum_{w\varkappa_i \in M \cup NV} \nu(\varkappa_i)$ and let $n \geq 1$ such that $2^{n-1} \leq m_w \leq 2^n$, and

- for every $j \leq k$, let $u_j^w = \displaystyle\sum_{w\varkappa_s \in M \cup NV,\, 1 \leq s < j} \nu(\varkappa_s)$.

Then $\xi_w = \zeta_w[\mathrm{SUB}_w]$, where SUB_w is the substitution sequence composed of all subsequences $\mathrm{sub}_{w\varkappa_j}$ for all $w\varkappa_j \in M \cup NV$ where

$$\mathrm{sub}_{w\varkappa_j} \;:=\; \langle u_j^w \rangle_{\mathrm{bin}}^n \to \xi_{w\varkappa_j}, \;\ldots\;, \langle u_j^w + \nu(\varkappa_j) - 1 \rangle_{\mathrm{bin}}^n \to \xi_{w\varkappa_j}$$

if $\nu(\varkappa_j) \neq 0$ and $\mathrm{sub}_{w\varkappa_j} := \varepsilon$ otherwise.

The next two observations follow easily from the fact that ξ_w is constructed by substituting trees $\xi_{w\varkappa_i}$ in the nth level ($n > 1$) of ζ_w with leaving the root at level 0 unchanged.

▶ **Observation 30.** *For each $w \in M$ we have $\xi_w(\varepsilon) = w$.*

▶ **Observation 31.** *For each $u \in \mathrm{pos}(\xi_\varepsilon)$ we have $\xi_\varepsilon(u) = w\varkappa$ for some $w \in M$ and $\varkappa \in NV$ if and only if*
(a) $[\xi_\varepsilon]_u = \xi_{w\varkappa}$ and
(b) there is a $v \in \{0,1\}^*$ with $v <_{\mathrm{prefix}} u$ and $\xi_\varepsilon(v) = w$, and for each $\varrho \in \{0,1\}^*$ with $v <_{\mathrm{prefix}} \varrho <_{\mathrm{prefix}} u$ we have $\xi_\varepsilon(\varrho) = \hat{w}$.

The mapping $\nu\colon NV \to \mathbb{N}$ can be extended to $\hat{\nu}\colon NV^* \to \mathbb{N}$ by letting $\hat{\nu}(\varkappa_{i_1}\ldots\varkappa_{i_l}) = \nu(\varkappa_{i_1}) \cdot \ldots \cdot \nu(\varkappa_{i_l})$; in the following we identify $\hat{\nu}$ and ν.

$\triangleright$ **Claim 32.** For each $w \in M$ and $\varkappa_i \in NV$ such that $w\varkappa_i \in M$ we obtain $|\xi_\varepsilon|_{w\varkappa_i} = \nu(w\varkappa_i)$.

Proof. By induction on w: For the induction base assume that $w = \varepsilon$. By construction and Observation 31, each node labeled $\varkappa_i$ is the root of a subtree $\xi_{\varkappa_i}$ and originates from the substitution of $\mathrm{sub}_{\varkappa_i}$ into $\zeta_\varepsilon[\mathrm{sub}_{\varkappa_1}, \ldots, \mathrm{sub}_{\varkappa_{i-1}}]$. By definition of $\mathrm{sub}_{\varkappa_i}$, this substitution takes place $\nu(\varkappa_i)$ many times at parallel positions. Thus, $|\xi_\varepsilon|_{\varkappa_i} = \nu(\varkappa_i)$.

Now assume that $w = u\varkappa \in M$ with $\varkappa \in NV$ and $|\xi_\varepsilon|_w = \nu(w)$. Moreover, let $\varkappa_i \in NV$ such that $w\varkappa_i \in M$. By Observation 31 (and with the same argumentation as above), each node labeled $w\varkappa_i$ occurs $\nu(\varkappa_i)$ many times in ξ_w. As, by assumption, w and, thus, also ξ_w occurs $\nu(w)$ many times in ξ_ε, we obtain

$$|\xi_\varepsilon|_{w\varkappa_i} = |\xi_w|_{\varkappa_i} \cdot |\xi_\varepsilon|_w = \nu(\varkappa_i) \cdot \nu(w) = \nu(w\varkappa_i)\,.$$

◄

Now we want to show that ξ_ε is indeed a model of $\varphi_\mathcal{D}$, i.e., $\xi_\varepsilon \models \varphi_{\mathrm{lab}}$, $\xi_\varepsilon \models \varphi_{\mathrm{prod}}$, and $\xi_\varepsilon \models \varphi_{\mathrm{sol}}$.

$\xi_\varepsilon \models \varphi_{\mathrm{lab}}$: It follows from Observation 30 that $\xi_\varepsilon \models \exists x \in \mathrm{P}_\varepsilon.\varphi_{\mathrm{root}}(x)$. Now let $w \in M$, $u \in \mathrm{pos}(\xi_\varepsilon)$ such that $\xi_\varepsilon(u) \in \{w, \hat{w}\}$, and $v \in \{u0, u1\}$. If $\xi_\varepsilon(u) = w$, by Observation 31, $[\xi_\varepsilon]_u = \xi_w$ and, thus, either $\xi_\varepsilon(v) = \hat{w}$ or $[\xi_\varepsilon]_v = \xi_{w\varkappa_j}$ and $\xi_\varepsilon(v) = \xi_{w\varkappa_j}(\varepsilon)$ for some $j \in [k]$. By Observation 30, $\xi_{w\varkappa_j}(\varepsilon) = w\varkappa_j$. If $\xi_\varepsilon(u) = \hat{w}$, we can argue similarly. Hence, $\xi_\varepsilon \models \bigwedge_{w \in M} \big(\forall x \in \mathrm{P}_w \cup \mathrm{P}_{\hat{w}}.\forall y.x \succ y \Rightarrow \mathrm{P}_{\hat{w}}(y) \vee \bigvee_{w\varkappa_i \in M} \mathrm{P}_{w\varkappa_i}(y) \big)$.

$\xi_\varepsilon \models \varphi_{\mathrm{prod}}$: By analyzing φ_{prod} we observe that $\xi_\varepsilon \models \varphi_{\mathrm{prod}}$ iff for each $w, w\varkappa_i \in M$ and $u \in \mathrm{pos}(\xi_\varepsilon)$ with $\xi_\varepsilon(u) = w$ it holds that

$$|[\xi_\varepsilon]_u|_{w\varkappa_i} = |\xi_\varepsilon|_{\varkappa_i} < \infty\,.$$

By Claim 32, $|\xi_\varepsilon|_{\varkappa_i} = \nu(\varkappa_i)$. On the other hand, by Observation 31, $|[\xi_\varepsilon]_u|_{w\varkappa_i}$ corresponds to the number of occurrences of $\xi_{w\varkappa_i}$ in ξ_w which is ensured by $\mathrm{sub}_{w\varkappa_{i_j}}$ to be $\nu(\varkappa_i)$, too.

$\xi_\varepsilon \models \varphi_{\mathrm{sol}}$: By assumption we know that the equation

$$\sum_{w=\varkappa_1^{i_1}\ldots\varkappa_k^{i_k} \in M} n_w \cdot \nu(\varkappa_1)^{i_1} \cdot \ldots \cdot \nu(\varkappa_k)^{i_k} = \sum_{w=\varkappa_1^{i_1}\ldots\varkappa_k^{i_k} \in M} m_w \cdot \nu(\varkappa_1)^{i_1} \cdot \ldots \cdot \nu(\varkappa_k)^{i_k}\,.$$

is satisfied. Moreover, by Claim 32, $|\xi_\varepsilon|_w = \nu(w)$ for each $w \in M$. It follows that $\xi_\varepsilon \models \sum_{w \in M} \mathtt{n}_w \, \#\mathrm{P}_w =_{\mathtt{fin}} \sum_{w \in M} \mathtt{m}_w \, \#\mathrm{P}_w$.

$\Leftarrow$: The proof for the other direction works with a similar argumentation. Essentially, as Claim 32 can be shown for an arbitrary model ξ of φ_D and a mapping $\nu\colon NV \to \mathbb{N}$ given by $\nu(\varkappa) = |\xi|_\varkappa$, we obtain that ν is a solution of $\mathcal{D}$. ◄

B   Stepwise Simplification of ωMSO⋈BAPA Formulae

Throughout the transformation, we will make use of formulae of the shape $\varphi_S(\iota)$ where S is a set term and ι is an individual term. This is defined inductively as follows:

$$
\begin{aligned}
\varphi_{\{\mathtt{a}\}}(\iota) &:= \{\mathtt{a}\}(\iota) \\
\varphi_{\mathtt{P}}(\iota) &:= \mathtt{P}(\iota) \\
\varphi_X(\iota) &:= X(\iota) \\
\varphi_{S^c}(\iota) &:= \mathrm{NNF}(\neg\varphi_S(\iota)) \\
\varphi_{S_1 \cup S_2}(\iota) &:= \varphi_{S_1}(\iota) \wedge \varphi_{S_2}(\iota) \\
\varphi_{S_1 \cap S_2}(\iota) &:= \varphi_{S_1}(\iota) \vee \varphi_{S_2}(\iota)
\end{aligned}
$$

It should be clear that $\varphi_S(\iota)$ and $S(\iota)$ are equivalent.

Now we describe the normalization process. In the course of our treatise, we will often speak of (and introduce) *fresh* variables or signature elements. By this, we mean symbols that are entirely new and have not been seen before; in particular, they are neither already present in the current formula, nor in any of the formula's previous "versions" throughout the normalization process.

B.1   Simplification and Skolemization

- Remove complex set terms from finiteness and modulo atoms:
  - Replace every $\mathrm{Fin}(S)$ where S is not a set variable by $\exists Z.Z = S \wedge \mathrm{Fin}(Z)$ where Z is a fresh set variable.
  - Replace every $\#S \equiv_{\mathtt{m}} \mathtt{n}$ where S is not a set variable by $\exists Z.Z = S \wedge \#Z \equiv_{\mathtt{m}} \mathtt{n}$ where Z is a fresh set variable.

  We note that these replacements do not change the status of any variable inside S and the result will remain in ωMSO⋈BAPA.
- Remove set operations from set atoms by replacing every $S(\iota)$ by $\varphi_S(\iota)$, whenever S contains any of $\cdot^c$, $\cap$, $\cup$. Again, the transformation clearly preserves membership in ωMSO⋈BAPA.
- Remove simple Presburger atoms by the following subsequent equivalent transformations:
  - Evaluate all (simple) Presburger atoms without occurrence of any $\#S$, and, depending on the result, replace them with **true** or **false**.
  - Replace all simple Presburger atoms of the shape $t \leq \infty$ by **true**.
  - Replace all simple Presburger atoms of the shape $t \leq_{\mathtt{fin}} \infty$ by **false**.
  - Replace all simple Presburger atoms of the shape $t + \mathtt{n} \leq_{(\mathtt{fin})} t' + \mathtt{m}$ by
    * $t + \mathtt{k} \leq_{(\mathtt{fin})} t'$ if $n > m$, and
    * $t \leq_{(\mathtt{fin})} t' + \mathtt{k}$ otherwise,
    where $k = |m - n|$.
  - Replace $\mathtt{m} \leq_{(\mathtt{fin})} \mathtt{n}\,\#S$ by $\mathtt{k} \leq_{(\mathtt{fin})} \#S$, where $k = \lceil \frac{m}{n} \rceil$.
  - Replace $\mathtt{n}\,\#S \leq_{(\mathtt{fin})} \mathtt{m}$ by $\#S \leq_{(\mathtt{fin})} \mathtt{k}$, where $k = \lfloor \frac{m}{n} \rfloor$.
  - Replace $0 \leq \#S$ and $0 \leq \#S + \mathtt{n}$ by **true**.
  - Replace $0 \leq_{\mathtt{fin}} \#S$ and $0 \leq_{\mathtt{fin}} \#S + \mathtt{n}$ by $\mathrm{Fin}(S)$.
  - Replace $\mathtt{m} \leq_{\mathtt{fin}} \#S$ by $\mathrm{Fin}(S) \wedge \mathtt{m} \leq \#S$.
  - Replace $\mathtt{m} \leq \#S$ by

  $$
  \exists x_1 \ldots x_m. \bigwedge_{1 \leq i \leq m} \varphi_S(x_i) \wedge \bigwedge_{1 \leq i < j \leq m} x_i \neq x_j.
  $$

- Replace $\#S \leq_{(\texttt{fin})} \texttt{n}$ by

$$\forall x_1 \ldots x_{n+1}. \bigvee_{0 \leq i \leq n+1} \varphi_{S^c}(x_i) \vee \bigvee_{0 \leq i < j \leq n+1} x_i = x_j.$$

  (recall that the empty conjunction equals **true** and the empty disjunction equals **false**).

- Replace $\infty \leq \#S$ by $\neg\mathrm{Fin}(S)$.

It should be clear that this transformation produces an equivalent formula and entirely removes all simple Presburger atoms from the considered formula.

- Skolemize all assertive set and individual variables:
  - remove trailing $\exists X$ and $\exists x$
  - replace all free occurrences of X by $\texttt{P}_X$ (fresh unary predicate)
  - replace all free occurrences of x by $\texttt{c}_x$ (fresh individual constant)

This is a satisfiability-preserving transformation and afterward all assertive variables are gone. In particular, all variables that now still occur in Presburger atoms are delicate.

B.2   Presburgerization: Separation of Variables

We *"presburgerize"* all non-Presburger atoms with delicate set variables. This may require to introduce further auxiliary unary predicates, extending the signature, and is done in two steps:

1. One by one, turn each delicate individual variable y into a fresh delicate set variable Y through the following procedure:

- in case y is existentially quantified, replace the subformula $\exists y.\varphi$ by

$$\exists Y.\big((\#Y = 1) \wedge \varphi_Y^y\big)$$

- in case y is universally quantified, replace $\forall y.\varphi$ by

$$\forall Y.\big((\#Y = 1) \Rightarrow \varphi_Y^y\big)$$

- thereby, given any formula φ with a free variable y, and any fresh set variable Y, we obtain φ_Y^y from φ as the end product of an exhaustive transformation sequence $\varphi = \varphi_0 \rightsquigarrow \varphi_1 \rightsquigarrow \ldots \rightsquigarrow \varphi_h = \varphi_Y^y$ where φ_{i+1} is obtained from φ_i by one of the following actions:
  - replacing any set atom $S(y)$ by $1 \leq \#(S \cap Y)$
  - picking an atom $\texttt{Q}(\iota_1, \ldots, \iota_n)$ in φ_i that contains y and letting

$$\varphi_{i+1} = \varphi_i' \wedge \forall y'.(\texttt{Q}(\iota_1, \ldots, \iota_n)[y \mapsto y'] \Leftrightarrow \texttt{P}_{\texttt{Q}(\iota_1, \ldots, \iota_n)}(y')),$$

    where y' is a fresh (and obviously non-delicate) individual variable and φ_i' is obtained from φ_i by replacing $\texttt{Q}(\iota_1, \ldots, \iota_n)$ with $1 \leq \#(\texttt{P}_{\texttt{Q}(\iota_1, \ldots, \iota_n)} \cap Y)$.[9]

Under the given circumstances, the transformation produces an equivalent formula in which all delicate individual variables have been removed.

2. Turn all remaining non-Presburger atoms containing some delicate set variable (and for which then *all* contained individual and set variables must be delicate by definition) into subformulae with only Presburger atoms $t \leq_{(\texttt{fin})} t'$:

---

[9] Note that, by the assumption of $\omega\mathrm{MSO}\bowtie\mathrm{BAPA}$ (i.e., each predicate atom contains at most one delicate variable), y is the only variable occurring in $\texttt{Q}(\iota_1, \ldots, \iota_n)$.

- Replace any $\#S \equiv_{\mathtt{n}} \mathtt{m}$ with delicate variables by

$$\exists \Bbbk. \big(\#S = \mathtt{n} \, \Bbbk + \mathtt{m} \big).$$

- replace any finiteness atom $\mathrm{Fin}(S)$ with delicate variables by $\neg(\infty \leq \#S)$.
- replace any set atom of the shape $S(\mathtt{a})$ with delicate variables and constant $\mathtt{a}$ by $1 \leq \#(S \cap \{\mathtt{a}\})$.

At the end of this sequence of transformations, we have obtained a separation of variable types: Next to number variables, Presburger atoms $t \leq_{(\mathtt{fin})} t'$ contain exclusively delicate set variables, while all other kinds of atoms contain neither delicate nor number variables.

B.3   Disentangling Quantifiers

An $\omega\text{MSO·BAPA}$ formula is said to be in *negation normal form* (short: NNF), if $\neg$ only occurs directly in front of predicate, set, or finiteness atoms. Every $\omega\text{MSO·BAPA}$ formula can be equivalently transformed into NNF, using the commonly known equivalences plus:

$$
\begin{aligned}
\neg(t \leq_{\mathtt{fin}} t') &\equiv (t' + 1 \leq_{\mathtt{fin}} t) \vee (\infty \leq t + t') \\
\neg(t \leq t') &\equiv (t' + 1 \leq_{\mathtt{fin}} t) \vee ((\infty \leq t) \wedge (t' \leq_{\mathtt{fin}} t')) \\
\neg(\#S \equiv_{\mathtt{n}} \mathtt{m}) &\equiv (\infty \leq \#S) \vee \bigvee_{\substack{k \in \{0,\dots,n-1\} \\ k \neq m \mod n}} \#S \equiv_{\mathtt{n}} \mathtt{k}
\end{aligned}
$$

Also it is easy to see that the normal form of a $\omega\text{MSO⋈BAPA}$ formula is again in $\omega\text{MSO⋈BAPA}$. For what follows, we assume that the considered formulas are in NNF.

The purpose of the *"disentangeling"* step of our normalization procedure is the following: While the previous transformations have ensured a variable separation between the different atom types (Presburger atoms use exclusively number and delicate set variables, whereas all other atoms use exclusively non-delicate individual and set variables), the scopes of the respective quantifiers are still containing atoms of either type. Our goal is to make sure that the quantifier scopes of number and delicate variables contain exclusively Presburger atoms, whereas the scopes of non-delicate variables contain none.

Consequently, we will call a formula *entangled*, if one or both of the following is the case:
- It contains a subformula of the form $\exists X.\psi$ or $\forall X.\psi$ with delicate X, or $\exists \Bbbk.\psi$ or $\forall \Bbbk.\psi$ such that ψ contains a predicate, set, modulo or finiteness atom.
- It contains a subformula of the form $\exists X.\psi$ or $\forall X.\psi$ with non-delicate X, or $\exists x.\psi$ or $\forall x.\psi$ such that ψ contains a Presburger atom.

A formula is *disentangled* if it is not entangled. Also, we call a disentangled $\omega\text{MSO⋈BAPA}$ formula *arithmetic* if it only contains Presburger atoms, while we call it *(plain) CMSO* if it only contains predicate, set, modulo and finiteness atoms. It is not hard to see that, by virtue of the negation normal form, every disentangled sentence can be written as a positive Boolean combination of arithmetic and plain CMSO formulae. Applying the distributive law and intelligent grouping, we can be even more restrictive: every disentangled formula ψ allows for a *disentangled disjunctive normal form* (DDNF) as well as a *disentangled conjunctive normal form* (DCNF) with the shapes

$$
\mathsf{DDNF}(\psi) = \bigvee_{i=1}^{k} \psi_{\vee i}^{\mathrm{arith}} \wedge \psi_{\vee i}^{\mathrm{plain}} \qquad
\mathsf{DCNF}(\psi) = \bigwedge_{j=1}^{l} \psi_{\wedge j}^{\mathrm{arith}} \vee \psi_{\wedge j}^{\mathrm{plain}}
$$

where the $\psi_{\vee i}^{\mathrm{arith}}$ and $\psi_{\wedge j}^{\mathrm{arith}}$ are arithmetic formulae while the $\psi_{\vee i}^{\mathrm{plain}}$ and $\psi_{\wedge j}^{\mathrm{plain}}$ are plain CMSO formulae.

Now, in order to disentangle an impure $\omega\text{MSO}\bowtie\text{BAPA}$ formula φ, we repeatedly do the following: We pick an occurrence of a minimal entangled subformula χ of φ (i.e., one all of whose proper subformulae are disentangled), and replace it with χ', the "disentangled equivalent variant" of ψ. Obviously, this procedure terminates – the number of replacement steps needed is bounded by the number of quantifiers of φ – and produces a disentangled formula.

It remains to provide a method to obtain χ' from χ. First observe that by assumption of minimality, χ must start with a quantifier followed by some disentangled formula ψ. We now make a case distinction depending on the quantifier: obtain χ' from χ through the following function (where X is delicate while Y isn't, and we assume $\mathsf{DDNF}(\psi)$ and $\mathsf{DCNF}(\psi)$ as above):

$$\exists \Bbbk.\psi \;\mapsto\; \bigvee_{i=1}^{k} \exists \Bbbk.\psi_{\vee i}^{\mathrm{arith}} \wedge \psi_{\vee i}^{\mathrm{plain}} \qquad\qquad \forall \Bbbk.\psi \;\mapsto\; \bigwedge_{j=1}^{l} \forall \Bbbk.\psi_{\wedge j}^{\mathrm{arith}} \vee \psi_{\wedge j}^{\mathrm{plain}}$$

$$\exists X.\psi \;\mapsto\; \bigvee_{i=1}^{k} \exists X.\psi_{\vee i}^{\mathrm{arith}} \wedge \psi_{\vee i}^{\mathrm{plain}} \qquad\qquad \forall X.\psi \;\mapsto\; \bigwedge_{j=1}^{l} \forall X.\psi_{\wedge j}^{\mathrm{arith}} \vee \psi_{\wedge j}^{\mathrm{plain}}$$

$$\exists x.\psi \;\mapsto\; \bigvee_{i=1}^{k} \psi_{\vee i}^{\mathrm{arith}} \wedge \exists x.\psi_{\vee i}^{\mathrm{plain}} \qquad\qquad \forall x.\psi \;\mapsto\; \bigwedge_{j=1}^{l} \psi_{\wedge j}^{\mathrm{arith}} \vee \forall x.\psi_{\wedge j}^{\mathrm{plain}}$$

$$\exists Y.\psi \;\mapsto\; \bigvee_{i=1}^{k} \psi_{\vee i}^{\mathrm{arith}} \wedge \exists Y.\psi_{\vee i}^{\mathrm{plain}} \qquad\qquad \forall Y.\psi \;\mapsto\; \bigwedge_{j=1}^{l} \psi_{\wedge j}^{\mathrm{arith}} \vee \forall Y.\psi_{\wedge j}^{\mathrm{plain}}$$

Obviously, given the structure of ψ with the ensured variable separation, χ and χ' are equivalent, and χ' is indeed disentangled as claimed.

While it may seem rather innocuous at the first glance, it should be noted that disentangling a formula may incur non-elementary blowup caused by the alternating transformations into DDNF and DCNF.

B.4 Vennification: Eliminating Delicate Variables

The strategy for removing delicate variables is inspired by a very similar technique used for treating BAPA [45]. The basic idea is to replace them by number variables, exploiting the fact that Presburger atoms only talk about cardinalities. However, in order to properly account for set operations (and even the mere fact that sets can overlap in different ways), some pre-processing is required, where the interplay of sets is broken down to disjoint indivisible smallest subsets, usually referred to as *Venn regions*.[10]

Given some finite set $\mathbf{U} \subseteq \mathbf{P}_1 \cup \{\{\mathsf{a}\} \mid \mathsf{a} \in \mathbf{C}\} \cup \mathbf{V}_{\mathrm{set}}$ of unary predicates, set variables, and constant-singleton sets, a *Venn region over* $\mathbf{U}$ is a set term R that is an intersection containing for each $U \in \mathbf{U}$ either U^c or U. For convenience, we will consider any two Venn regions whose set expressions contain the same U and U^c as syntactically equal. We denote the set of all Venn regions over $\mathbf{U}$ by $\mathrm{VR}(\mathbf{U})$. Given some $R \in \mathrm{VR}(\mathbf{U})$ and a set term S that is a Boolean combination of (a selection of) elements from $\mathbf{U}$ (short: S is a *set term over* $\mathbf{U}$), we write $R \models S$ to denote that $R^{\mathfrak{A},\nu} \subseteq S^{\mathfrak{A},\nu}$ for all structures $\mathfrak{A}$ and corresponding variable assignments ν. We note that $R \models S$ can be easily decided by solving the corresponding entailment problem for Boolean formulas. Furthermore $R \not\models S$ holds exactly if $R^{\mathfrak{A},\nu} \cap S^{\mathfrak{A},\nu} = \emptyset$ for all structures $\mathfrak{A}$ and corresponding variable assignments ν. Likewise, for any $R_1, R_2 \in \mathrm{VR}(\mathbf{U})$ with $R_1 \neq R_2$, we have that $R_1^{\mathfrak{A},\nu} \cap R_2^{\mathfrak{A},\nu} = \emptyset$ holds

---

[10] Another way to think of this is that the cardinality of each of these Venn regions is the number of elements realizing one complete unary type.

for all structures $\mathfrak{A}$ and corresponding variable assignments ν. These insights allow us to equivalently re-write, for any set term S over $\mathbf{U}$, the numerical term $\#S$ into

$$\sum_{\substack{R \in \mathrm{VR}(\mathbf{U}) \\ R \models S}} \#R.$$

Given a sentence φ, in order to have direct "access" to the Venn regions over $\mathbf{U}(\varphi) := \mathbf{P}_1(\varphi) \cup \{\{\mathtt{a}\} \mid \mathtt{a} \in \mathbf{C}(\varphi)\}$, we introduce one auxiliary unary predicate $\mathtt{V}_R$ for every $R \in \mathrm{VR}(\mathbf{U}(\varphi))$, and define the corresponding definitorial description as the sentence

$$\psi_\varphi^{\mathrm{VennDef}} := \bigwedge_{R \in \mathrm{VR}(\mathbf{U}(\varphi))} \forall x.(\mathtt{V}_R(x) \Leftrightarrow \varphi_R(x)).$$

Then, $\varphi' := \varphi \wedge \psi_\varphi^{\mathrm{VennDef}}$ is a model-conservative extension of φ and every model $\mathfrak{A}$ of φ' and every variable assignment ν satisfy the correspondency

$$\mathtt{V}_R^{\mathfrak{A}} = \mathtt{V}_R^{\mathfrak{A},\nu} = R^{\mathfrak{A},\nu}.$$

After these preparations, we finally remove the delicate variables from φ'. The intuitive idea behind this technique is to maintain a "histogram" storing all sizes of all Venn Regions constructible from the unary Predicates and "currently active variables". Thereby, the size of each Venn region R is "stored" in a number variable called $\Bbbk_R$. Whenever a new (necessarily delicate) set variable X is supposed to be introduced via an existential or universal quantifier, we instead "refine" the histogram of Venn regions: every prior Venn region is split in two (depending on membership or non-membership in X), giving rise to newly introduced number variables $\Bbbk_{R \cap X}$ and $\Bbbk_{R \cap X^c}$ which need to add up to $\Bbbk_R$. As discussed above, this numerical information is sufficient to provide enough information for evaluating the Presburger atoms.

Formally, this transformation is realized by replacing any of its arithmetic sub-sentences χ by

$$\left(\exists \Bbbk_R\right)_{R \in \mathrm{VR}(\mathbf{U}(\varphi))} \cdot \left(\bigwedge_{R \in \mathrm{VR}(\mathbf{U}(\varphi))} \Bbbk_R = \#\mathtt{V}_R\right) \wedge \mathsf{transf}(\chi, \mathbf{U}(\varphi)),$$

where transf is recursively defined by

$$
\begin{aligned}
\mathsf{transf}(\psi_1 \wedge \psi_2, \mathbf{U}) &:= \mathsf{transf}(\psi_1, \mathbf{U}) \wedge \mathsf{transf}(\psi_1, \mathbf{U}) \\
\mathsf{transf}(\psi_1 \vee \psi_2, \mathbf{U}) &:= \mathsf{transf}(\psi_1, \mathbf{U}) \vee \mathsf{transf}(\psi_1, \mathbf{U}) \\
\mathsf{transf}(\ \exists \Bbbk.\psi\ , \mathbf{U}) &:= \exists \Bbbk.\mathsf{transf}(\psi, \mathbf{U}) \\
\mathsf{transf}(\ \forall \Bbbk.\psi\ , \mathbf{U}) &:= \forall \Bbbk.\mathsf{transf}(\psi, \mathbf{U}) \\
\mathsf{transf}(\ \exists X.\psi\ , \mathbf{U}) &:= (\exists \Bbbk_{R \cap X}.\exists \Bbbk_{R \cap X^c})_{R \in \mathrm{VR}(\mathbf{U})} \cdot \left(\bigwedge_{R \in \mathrm{VR}(\mathbf{U})} \Bbbk_R = \Bbbk_{R \cap X} + \Bbbk_{R \cap X^c}\right) \wedge \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathsf{transf}(\psi, \mathbf{U} \cup \{X\}) \\
\mathsf{transf}(\ \forall X.\psi\ , \mathbf{U}) &:= (\forall \Bbbk_{R \cap X}.\forall \Bbbk_{R \cap X^c})_{R \in \mathrm{VR}(\mathbf{U})} \cdot \left(\bigvee_{R \in \mathrm{VR}(\mathbf{U})} \Bbbk_R \neq \Bbbk_{R \cap X} + \Bbbk_{R \cap X^c}\right) \vee \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathsf{transf}(\psi, \mathbf{U} \cup \{X\})
\end{aligned}
$$

$$
\begin{aligned}
\mathsf{transf}(\ t_1 \leq t_2\ , \mathbf{U}) &:= \mathsf{transf}(t_1, \mathbf{U}) \leq \mathsf{transf}(t_2, \mathbf{U}) \\
\mathsf{transf}(t_1 \leq_{\mathtt{fin}} t_2, \mathbf{U}) &:= \mathsf{transf}(t_1, \mathbf{U}) \leq_{\mathtt{fin}} \mathsf{transf}(t_2, \mathbf{U}) \\
\mathsf{transf}(\ t_1 + t_2\ , \mathbf{U}) &:= \mathsf{transf}(t_1, \mathbf{U}) + \mathsf{transf}(t_2, \mathbf{U}) \\
\mathsf{transf}(\quad \mathtt{m}\, t\quad , \mathbf{U}) &:= \mathtt{m}\, \mathsf{transf}(t_2, \mathbf{U}) \\
\mathsf{transf}(\quad \mathtt{n}\quad , \mathbf{U}) &:= \mathtt{n} \\
\mathsf{transf}(\quad \infty\quad , \mathbf{U}) &:= \infty \\
\mathsf{transf}(\quad \Bbbk\quad , \mathbf{U}) &:= \Bbbk \\
\mathsf{transf}(\quad \#S\quad , \mathbf{U}) &:= \sum_{\substack{R \in \mathrm{VR}(\mathbf{U}) \\ R \models S}} \Bbbk_R
\end{aligned}
$$

After this transformation, we arrive at a modified φ wherein all arithmetic subsentences are entirely free of individual and set variables.

B.5 Eliminating Number Variables

In the next step, we will eliminate all number variables from arithmetic subsentences. To this end, we will use the well known quantifier elimination procedure from Presburger arithmetic. As classical Presburger arithmetic is defined over the natural numbers (without ∞), this requires another preprocessing step. Intuitively, this preprocessing implements a case distinction for every variable introduced through quantification, as to if the variable is instatiated by a natural number proper, or ∞.

Clearly, via standard transformation (extended to ∞ in the straightforward way), every remaining number term can be written in summarized, maximally simplified form of the shape

$$\mathbf{n}_0 + \sum_{i=1}^{k} \mathbf{n}_i t_i^* \qquad \text{or} \qquad \infty$$

where $n_0 \in \mathbb{N}$, each $n_i \in \mathbb{N} \setminus \{0\}$, and every t_i^* is either a number variable or of the form #P. For any term t, let $\mathsf{NF}(t)$ denote the corresponding equivalent term of this form. For formulae φ, let $\mathsf{NF}(\varphi)$ denote φ with every maximal term t occurring in φ replaced by $\mathsf{NF}(t)$.

We now define the function smpl which simplifies arithmetic formulae as follows (where t^*, t_1^*, t_2^* stand for number terms whose NF is not ∞):

$$
\begin{aligned}
\mathbf{true} &\mapsto \mathbf{true} \\
\mathbf{false} &\mapsto \mathbf{false} \\
\varphi \wedge \varphi' &\mapsto \mathsf{smpl}(\varphi) \wedge \mathsf{smpl}(\varphi') \\
\varphi \vee \varphi' &\mapsto \mathsf{smpl}(\varphi) \vee \mathsf{smpl}(\varphi') \\
\exists \Bbbk.\varphi &\mapsto \mathsf{smpl}(\mathsf{NF}(\varphi[\Bbbk \mapsto \infty])) \vee \exists \Bbbk.\mathsf{smpl}(\varphi) \\
\forall \Bbbk.\varphi &\mapsto \mathsf{smpl}(\mathsf{NF}(\varphi[\Bbbk \mapsto \infty])) \wedge \forall \Bbbk.\mathsf{smpl}(\varphi)
\end{aligned}
\qquad
\begin{aligned}
t \lessgtr_{\mathtt{fin}} \infty &\mapsto \mathbf{false} \\
t \leq \infty &\mapsto \mathbf{true} \\
\infty \lessgtr_{\mathtt{fin}} t^* &\mapsto \mathbf{false} \\
\infty \leq t^* &\mapsto \mathbf{false} \\
t_1^* \lessgtr_{\mathtt{(fin)}} t_2^* &\mapsto t_1^* \lessgtr_{\mathtt{fin}} t_2^*
\end{aligned}
$$

This is an equivalent transformation when confining the attention to structures that map all occurring #P to finite numbers. Building on that, we define the function smpl^* that equivalently rewrites arbitrary arithmetic subsentences as follows. For an arithmetic subsentence χ in NNF of the overall formula φ, let

$$\mathsf{smpl}^*(\chi) := \bigvee_{\mathbf{U} \subseteq \mathbf{U}(\varphi)} \left(\left(\bigwedge_{S \in \mathbf{U}} \neg \mathrm{Fin}(S) \right) \wedge \left(\bigwedge_{S \in \mathbf{U}(\varphi) \setminus \mathbf{U}} \mathrm{Fin}(S) \right) \wedge \chi^{\mathbf{U}} \right)$$

where

$$\chi^{\mathbf{U}} := \mathsf{smpl}\left(\mathsf{NF}(\varphi\,[\#S \mapsto \infty]_{S \in \mathbf{U}}) \right).$$

We note that every $\chi^{\mathbf{U}}$ can be seen as a formula of standard Presburger arithmetic, as all still occurring #P are constrained to be finite by the environment of $\chi^{\mathbf{U}}$ in $\mathsf{smpl}^*(\chi)$. This allows us to apply classical quantifier elimination [51] to each $\chi^{\mathbf{U}}$, yielding a sentence $\mathrm{QE}(\chi^{\mathbf{U}})$ free of any number variables, but possibly containing modulo atoms next to classical Presburger atoms. Letting $\mathrm{QE}(\mathsf{smpl}^*(\chi))$ denote $\mathsf{smpl}^*(\chi)$ with every subsentence $\chi^{\mathbf{U}}$ replaced by $\mathrm{QE}(\chi^{\mathbf{U}})$, we finally obtain the new formula φ by replacing all arithmetic subsentences χ by $\mathrm{QE}(\mathsf{smpl}^*(\chi))$. This yields us a sentence entirely free of number variables, which is a positive Boolean combination of sentences of the following two kinds:

▪ sentences containing (possibly negated) predicate, set, finiteness and modulo atoms, and

⬛ unnegated classical Presburger atoms without variables.

Using distributivity, we can ensure that the obtained sentence has the form

$$\bigvee_{i=1}^{k} \left(\varphi_i \wedge \bigwedge_{j=1}^{l_i} \chi_{i,j} \right),$$

where the φ_i are CMSO sentences, i.e., MSO sentences with possible additional occurrences of modulo and finiteness atoms, while the $\chi_{i,j}$ are unnegated classical Presburger atoms without variables.

B.6   De-Skolemization

At this point, the formula (say, φ^{new}) we have arrived at uses a signature $\mathbb{S}(\varphi^{\mathrm{new}})$ that is a superset of the signature $\mathbb{S}(\varphi^{\mathrm{orig}})$ used by the original formula φ^{orig}. The initial Skolemization has introduced auxiliary constant and unary predicate names, and more unary predicates were introduced in the course of "Presburgerization" and "Vennization". The formula φ^{new} is then a model-conservative extension of the original formula φ^{orig}: every model of the obtained formula is a model of the original one and every model of the original one can be turned into a model of the current one by picking appropriate interpretations for the added signature elements.[11] In order to re-gain equivalence with φ^{orig} we therefore need to "project away" these additional signature elements by conceiving them as individual and set variables and existentially quantifying over them. An exception is made for the representatives of the formerly free variables, which will be free again. Formally, we obtain

$$\varphi^{\mathrm{free\text{-}again}} := \varphi^{\mathrm{new}} \big[\mathsf{P}_X \mapsto X \big]_{X \in \mathrm{free}(\varphi^{\mathrm{orig}}) \cap \mathbf{V}_{\mathrm{set}}} \big[\mathsf{c}_x \mapsto x \big]_{x \in \mathrm{free}(\varphi^{\mathrm{orig}}) \cap \mathbf{V}_{\mathrm{ind}}}$$

and, with $\mathbb{S}^{\mathrm{diff}} = \mathbb{S}(\varphi^{\mathrm{new}}) \setminus \mathbb{S}(\varphi^{\mathrm{orig}})$ we let

$$\varphi^{\mathrm{de\text{-}skol}} := \big(\exists X_{\mathsf{P}} \big)_{\mathsf{P} \in \mathbb{S}^{\mathrm{diff}}_{\mathbf{P}}} . \big(\exists x_{\mathsf{c}} \big)_{\mathsf{c} \in \mathbb{S}^{\mathrm{diff}}_{\mathbf{C}}} . \varphi^{\mathrm{free\text{-}again}} \big[\mathsf{P} \mapsto X_{\mathsf{P}} \big]_{\mathsf{P} \in \mathbb{S}^{\mathrm{diff}}_{\mathbf{P}}} \big[\mathsf{c} \mapsto x_{\mathsf{c}} \big]_{\mathsf{c} \in \mathbb{S}^{\mathrm{diff}}_{\mathbf{C}}},$$

and conclude that $\varphi^{\mathrm{de\text{-}skol}}$ is equivalent to φ^{orig}. Now having in mind that by construction, $\varphi^{\mathrm{de\text{-}skol}}$ has the shape

$$\exists X_1. \cdots \exists X_n. \exists x_1. \cdots \exists x_m. \bigvee_{i=1}^{k} \left(\varphi_i \wedge \bigwedge_{j=1}^{l_i} \chi_{i,j} \right),$$

where all $\chi_{i,j}$ are free of individual variables, we see that we can pull the quantifier block $\exists x_1. \cdots \exists x_m$ past the disjunction and then inside the φ_i. Last not least, we can, for every $\#\mathsf{P}$ still occurring in some $\chi_{i,j}$, replace it by $\#X_{\mathsf{P}}$ (where X_{P} is a fresh set variable), add the conjunct $X_{\mathsf{P}} = \mathsf{P}$ to every φ_i and put the quantifier $\exists X_{\mathsf{P}}$ in front of the whole formula.

With this we have – at last – arrived at a formula in GNF as specified in Definition 7.

---

[11] In case the original formula had free variables, a similar statement holds where the models are accompanied by variable assignments.

C Properties of PMTA

Given some $n \in \mathbb{N}$, a set A, and a tuple $a = (a_1, \ldots, a_n) \in A^n$, we let $\mathrm{pr}_i(a) = a_i$ for every $i \in [n]$, that is, pr_i denotes the ith *projection*. Projections are lifted to sets of tuples as usual.

By writing $\xi[\xi_1, \ldots, \xi_k]$ we denote the composition of a finite tree ξ with trees $\xi_1, \ldots, \xi_k$ in the usual way, i.e., we equip ξ with variables (two variables under each leaf and one variable under each position with only one successor) and we replace the ith variable of ξ by the tree ξ_i – under the assumption that k corresponds to the number of variables ξ is equipped with. If $\mathrm{pos}(\xi) = \emptyset$, we let $\xi[\xi_1] = \xi_1$.

C.1 Closure Properties

▶ **Proposition 14.** *The set of tree languages recognized by Parikh-Muller tree automata is closed under union, intersection, and relabeling.*

Proof. The result follows from the subsequent Lemmas 33, 34, and 35. ◀

▶ **Lemma 33.** *PMTA are closed under union.*

Proof. Let $\mathcal{A}_i = (Q_i, \Xi_i, q_{I,i}, \Delta_i, \mathcal{F}_i, C_i)$ for $i \in [2]$ be two PMTA with $Q_i = Q_{P,i} \cup Q_{\omega,i} \cup \{q_{I,i}\}$, $\Xi_i = (\Sigma \times D_i) \cup \Sigma$, and $\Delta_i = \Delta_{P,i} \cup \Delta_{\omega,i}$. W.l.o.g. we can assume that $\mathcal{A}_1$ and $\mathcal{A}_2$ are both of dimension s for some $s \geq 1$ (semi-linear sets are closed under concatenation by Lemma 11), that Q_1 and Q_2 are disjoint, and that q_I is a fresh state not in $Q_1 \cup Q_2$.

We construct the PMTA $\mathcal{A} = (Q, \Xi, q_I, \Delta, \mathcal{F}, C)$ where

- $Q = Q_1 \cup Q_2 \cup \{q_I\}$,
- $\Xi = \Big(\Sigma \times (D_1 \cdot \{(0), (1)\} \cup D_2 \cdot \{(0), (2)\}) \Big) \cup \Sigma$,
- $\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2$,
- $C = C_1 \cdot \{(1)\} \cup C_2 \cdot \{(2)\}$, and
- $\Delta = \Delta_P \cup \Delta_\omega$ consists of the transitions
  - $\Delta_P = \{(q, (\sigma, \vec{d} \cdot (0)), q_1, q_2) \mid (q, (\sigma, \vec{d}), q_1, q_2) \in \Delta_{P,1} \cup \Delta_{P,2}\} \cup$
    $\{(q_I, (\sigma, \vec{d} \cdot (i)), q_1, q_2) \mid (q_{I,i}, (\sigma, \vec{d}), q_1, q_2) \in \Delta_{P,i} \text{ for some } i \in [2]\}$ and
  - $\Delta_\omega = \Delta_{\omega,1} \cup \Delta_{\omega,2} \cup \{(q_I, \gamma, q_1, q_2) \mid (q_{I,i}, \gamma, q_1, q_2) \in \Delta_{\omega,i} \text{ for some } i \in [2]\}$.

$\mathcal{L}(\mathcal{A}_1) \cup \mathcal{L}(\mathcal{A}_2) \subseteq \mathcal{L}(\mathcal{A})$: Let $\xi \in \mathcal{L}(\mathcal{A}_i)$ for some $i \in [2]$. Then there exists some $\zeta \in T_{\Xi_i}^\omega$ s.t. $(\zeta)_\Sigma = \xi$ and there is an accepting run κ_ζ of $\mathcal{A}_i$ on ζ. By def., $\kappa_\zeta(\varepsilon) = q_{I,i}$.

(1) Assume that $\zeta_{\mathrm{cnt}} \neq \emptyset$. Consider the tree $\zeta' \in T_\Xi^\omega$ with

$$\zeta'(\varrho) = \begin{cases} (\sigma, \vec{d} \cdot (i)) & \text{if } \varrho = \varepsilon, \zeta(\varepsilon) = (\sigma, \vec{d}) \\ (\sigma, \vec{d} \cdot (0)) & \text{if } \varrho \in \mathrm{pos}(\zeta_{\mathrm{cnt}}) \setminus \{\varepsilon\}, \zeta(\varrho) = (\sigma, \vec{d}) \\ \zeta(\varrho) & \text{otherwise} \end{cases}.$$

Clearly, $(\zeta')_\Sigma = (\zeta)_\Sigma = \xi$. Now consider the tree κ' with $\kappa'(\varepsilon) = q_I$ and $\kappa'(\varrho) = \kappa_\zeta(\varrho)$ for each $\varrho \in \{0,1\}^+$. By construction and as κ_ζ respects the transition relation of $\mathcal{A}_i$, $(q_I, \zeta'(\varepsilon), \kappa'(0), \kappa'(1)) \in \Delta$ and $(\kappa'(\varrho), \zeta'(\varrho), \kappa'(\varrho 0), \kappa'(\varrho 1)) \in \Delta$ for each $\varrho \in \{0,1\}^+$. Thus, κ' is a run of $\mathcal{A}$ on ζ'. As $\Psi(\zeta'_{\mathrm{cnt}}) = \Psi(\zeta_{\mathrm{cnt}}) \cdot (i)$ and $\Psi(\zeta_{\mathrm{cnt}}) \in C_i$, $\Psi(\zeta'_{\mathrm{cnt}}) \in (C_i \cdot \{(i)\}) \subseteq C$. Finally, as $\inf(\kappa'(\pi)) = \inf(\kappa_\zeta(\pi))$ for each path π, we obtain that κ' is accepting. Hence, $\xi \in \mathcal{L}(\mathcal{A})$.

(2) Assume that $\zeta_{\mathrm{cnt}} = \emptyset$. Similar argumentation as above.

$\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{A}_1) \cup \mathcal{L}(\mathcal{A}_2)$: Let $\xi \in \mathcal{L}(\mathcal{A})$. Then there exists some $\zeta \in T_\Xi^\omega$ s.t. $(\zeta)_\Sigma = \xi$ and there is some accepting run κ_ζ of $\mathcal{A}$ on ζ.

(1) Assume $\zeta_{\mathrm{cnt}} \neq \emptyset$. Then $\zeta(\varepsilon) = (\sigma, \vec{d} \cdot (i))$ for some $i \in [2]$, $\sigma \in \Sigma$, $\vec{d} \in D_i$, and, for each $\varrho \in \mathrm{pos}(\zeta_{\mathrm{cnt}}) \setminus \{\varepsilon\}$, $\zeta(\varrho) = (\sigma_\varrho, \vec{d}_\varrho \cdot (0))$ for some $\sigma_\varrho \in \Sigma$, $\vec{d}_\varrho \in D_i$. By construction of Δ,

- $(q_{I,i}, (\sigma, \vec{d}), \kappa_\zeta(0), \kappa_\zeta(1)) \in \Delta_{P,i}$,
- $(\kappa_\zeta(\varrho), (\sigma_\varrho, \vec{d}_\varrho), \kappa_\zeta(\varrho 0), \kappa_\zeta(\varrho 1)) \in \Delta_{P,i}$ for each $\varrho \in \mathrm{pos}(\zeta_{\mathrm{cnt}}) \setminus \{\varepsilon\}$, and
- $(\kappa_\zeta(\varrho'), \zeta(\varrho'), \kappa_\zeta(\varrho' 0), \kappa_\zeta(\varrho' 1)) \in \Delta_{\omega,i}$ for each $\varrho' \in \{0,1\}^* \setminus \mathrm{pos}(\zeta_{\mathrm{cnt}})$.

It follows that the tree κ' with $\kappa'(\varepsilon) = q_{I,i}$ and $\kappa'(\varrho) = \kappa_\zeta$ for all $\varrho \in \{0,1\}^+$ is a run of $\mathcal{A}_i$ on the tree ζ' obtained from ζ by cropping the last element of each $\vec{d} \in D_i \cdot \{(0), (i)\}$. Clearly, $(\zeta')_\Sigma = (\zeta)_\Sigma = \xi$. As $\Psi(\zeta_{\mathrm{cnt}}) = \vec{w} \cdot (i)$ and $\Psi(\zeta_{\mathrm{cnt}}) \in C_i \cdot \{(i)\}$, we obtain that $\Psi(\zeta'_{\mathrm{cnt}}) = \vec{w} \in C_i$. Moreover, $\inf(\kappa'(\pi)) = \inf(\kappa_\zeta(\pi))$ for each path π. Thus, κ' is accepting and $\xi \in \mathcal{L}(\mathcal{A})$.

(2) Assume $\zeta_{\mathrm{cnt}} = \emptyset$. Similar argumentation as above. ◀

▶ **Lemma 34.** *PMTA are closed under intersection.*

Proof. Let, for each $i \in [2]$, $\mathcal{A}_i = (Q_i, \Xi_i, q_{I,i}, \Delta_i, \mathcal{F}_i, C_i)$ be a PMTA of dimension s_i with $Q_i = Q_{P,i} \cup Q_{\omega,i} \cup \{q_{I,i}\}$, $\Xi_i = (\Sigma \times D_i) \cup \Sigma$, and $\Delta_i = \Delta_{P,i} \cup \Delta_{\omega,i}$. W.l.o.g. we can assume that $(0)^{s_i} \in D_i$. Note that on a tree $\xi \in \mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2)$, $\mathcal{A}_1$ and $\mathcal{A}_2$ might not test the same initial part of ξ arithmetically. In order to simulate this "superposition", we need to test a bigger initial part and let non-active counters "idle" using zero-increments $(0)^{s_i}$.

Thus, we define the Cartesian product $\mathcal{A} = (Q, \Xi, (q_{I,1}, q_{I,2}), \Delta, \mathcal{F}, C)$ where

- $Q_\omega = Q_{\omega,1} \times Q_{\omega,2}$, and $Q_P = (Q_1 \times Q_2) \setminus Q_\omega$,
- $\Xi = (\Sigma \times D_1 \cdot D_2) \cup \Sigma$,
- $\mathcal{F} = \{F \subseteq Q_\omega \mid \mathrm{pr}_1(F) \in \mathcal{F}_1 \text{ and } \mathrm{pr}_2(F) \in \mathcal{F}_2\}$,
- $C = C_1 \cdot C_2$,

and $\Delta = \Delta_P \cup \Delta_\omega$ where

- Δ_P is the smallest set such that
  - for each $(q, (\sigma, \vec{d}), q_1, q_2) \in \Delta_{P,1}$ and $(q', (\sigma, \vec{d'}), q_1', q_2') \in \Delta_{P,2}$ we have $((q, q'), (\sigma, \vec{d} \cdot \vec{d'}), (q_1, q_1'), (q_2, q_2')) \in \Delta_P$,
  - for each $(q, (\sigma, \vec{d}), q_1, q_2) \in \Delta_{P,1}$ and $(q', \sigma, q_1', q_2') \in \Delta_{\omega,2}$ we have $((q, q'), (\sigma, \vec{d} \cdot (0)^{s_2}), (q_1, q_1'), (q_2, q_2')) \in \Delta_P$,
  - for each $(q, \sigma, q_1, q_2) \in \Delta_{\omega,1}$ and $(q', (\sigma, \vec{d'}), q_1', q_2') \in \Delta_{P,2}$ we have $((q, q'), (\sigma, (0)^{s_1} \cdot \vec{d'}), (q_1, q_1'), (q_2, q_2')) \in \Delta_P$, and
- $\Delta_\omega = \{((q, q'), \gamma, (q_1, q_1'), (q_2, q_2')) \mid (q, \gamma, q_1, q_2) \in \Delta_{\omega,1}, (q', \gamma, q_1', q_2') \in \Delta_{\omega,2}\}$.

$\mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2) \subseteq \mathcal{L}(\mathcal{A})$: Let $\xi \in \mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2)$. Then there are $\zeta_1 \in T^\omega_{\Xi_1}$, $\zeta_2 \in T^\omega_{\Xi_2}$ such that $(\zeta_1)_\Sigma = (\zeta_2)_\Sigma = \xi$ and for each $i \in [2]$ there exists an accepting run κ_i of $\mathcal{A}_i$ on ζ_i. Note that ζ_1 and ζ_2 might not have the same prefix enriched by Parikh vectors. Now consider the tree $\zeta \in T^\omega_\Xi$ with

$$
\zeta(\varrho) = \begin{cases}
(\sigma, \vec{d} \cdot \vec{d'}) & \text{if } \varrho \in \mathrm{pos}(\zeta_{1,\mathrm{cnt}}) \cap \mathrm{pos}(\zeta_{2,\mathrm{cnt}}), \zeta_1(\varrho) = (\sigma, \vec{d}), \zeta_2(\varrho) = (\sigma, \vec{d'}) \\
(\sigma, \vec{d} \cdot (0)^{s_2}) & \text{if } \varrho \in \mathrm{pos}(\zeta_{1,\mathrm{cnt}}) \setminus \mathrm{pos}(\zeta_{2,\mathrm{cnt}}), \zeta_1(\varrho) = (\sigma, \vec{d}), \zeta_2(\varrho) = \sigma \\
(\sigma, (0)^{s_1} \cdot \vec{d'}) & \text{if } \varrho \in \mathrm{pos}(\zeta_{2,\mathrm{cnt}}) \setminus \mathrm{pos}(\zeta_{1,\mathrm{cnt}}), \zeta_1(\varrho) = \sigma, \zeta_2(\varrho) = (\sigma, \vec{d'}) \\
\sigma & \text{if } \varrho \notin \mathrm{pos}(\zeta_{1,\mathrm{cnt}}) \cup \mathrm{pos}(\zeta_{2,\mathrm{cnt}}), \zeta_1(\varrho) = \zeta_2(\varrho) = \sigma
\end{cases}
$$

as well as the tree $\kappa \in T^\omega Q$ given by $\kappa(\varrho) = (\kappa_1(\varrho), \kappa_2(\varrho))$ for each $\varrho \in \{0,1\}^*$. As $\kappa_1(\varepsilon) = q_{I,1}$ and $\kappa_2(\varepsilon) = q_{I,2}$, $\kappa(\varepsilon) = (q_{I,1}, q_{I,2})$. Moreover, we obtain for each position ϱ

- if $\varrho \in \mathrm{pos}(\zeta_{1,\mathrm{cnt}}) \cap \mathrm{pos}(\zeta_{2,\mathrm{cnt}})$, then $(\kappa_i(\varrho), \zeta_i(\varrho), \kappa_i(\varrho 0), \kappa_i(\varrho 1)) \in \Delta_{P,i}$ for each $i \in [2]$. By construction of Δ_P, $((\kappa_1(\varrho), \kappa_2(\varrho)), \zeta(\varrho), (\kappa_1(\varrho 0)\kappa_2(\varrho 0)), (\kappa_1(\varrho 1)\kappa_2(\varrho 1))) \in \Delta_{P,i}$,

and for all other positions we can argue for the existence of the respective transition in a similar way. Hence, κ is a run of $\mathcal{A}$ on ζ.

Furthermore, if $\mathrm{pos}(\zeta_{1,\mathrm{cnt}}) \cup \mathrm{pos}(\zeta_{2,\mathrm{cnt}}) \neq \emptyset$, we obtain by assuming $\zeta_{\mathrm{cnt}}(j) = (\sigma_j, \vec{d_j} \cdot \vec{d_j}')$

$$\Psi(\zeta_{\mathrm{cnt}}) = \sum_{j \in \mathrm{pos}(\zeta_{\mathrm{cnt}})} \vec{d_j} \cdot \vec{d_j}' = (\sum_{j \in \mathrm{pos}(\zeta_{\mathrm{cnt}})} \vec{d_j}) \cdot (\sum_{j \in \mathrm{pos}(\zeta_{\mathrm{cnt}})} \vec{d_j}') = \Psi(\zeta_{1,\mathrm{cnt}}) \cdot \Psi(\zeta_{2,\mathrm{cnt}})$$

where the last equality holds since $\mathrm{pos}(\zeta_{i,\mathrm{cnt}}) \subseteq \mathrm{pos}(\zeta_{\mathrm{cnt}})$ and $\vec{d} + (0)^{s_i} = \vec{d}$ for each $i \in [2]$ and $\vec{d} \in \mathbb{N}^{s_i}$. As $\Psi(\zeta_{1,\mathrm{cnt}}) \in C_1$ and $\Psi(\zeta_{2,\mathrm{cnt}}) \in C_2$, we obtain $\Psi(\zeta_{1,\mathrm{cnt}}) \cdot \Psi(\zeta_{2,\mathrm{cnt}}) \in C_1 \cdot C_2$.

Finally, let π be an arbitrary path. Then $\inf(\kappa_1(\pi)) = F_1$ and $\inf(\kappa_2(\pi)) = F_2$ for some $F_1 \in \mathcal{F}_1$ and $F_2 \in \mathcal{F}_2$. As $\mathrm{pr}_i(\kappa(\pi)) = \kappa_i(\pi)$ and $\mathrm{pr}_i(\inf(\kappa(\pi))) = \inf(\mathrm{pr}_i(\kappa(\pi)))$, we obtain that $\mathrm{pr}_i(\inf(\kappa(\pi))) = \inf(\kappa_i(\pi))$ for each $i \in [2]$. Thus, $\inf(\kappa(\pi)) \in \mathcal{F}$ and κ is accepting on ζ. As $(\zeta)_\Sigma = \xi$, $\xi \in \mathcal{L}(\mathcal{A})$.

$\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2)$: Similar. By construction of Δ, each ζ and accepting run $\kappa \in \mathrm{Run}_{\mathcal{A}}(\zeta)$ are of a particular form: ζ can be decomposed into

$$\hat{\zeta}[\zeta_1[\zeta_1^1, \ldots, \zeta_{m_1}^1], \ldots, \zeta_k[\zeta_1^k, \ldots, \zeta_{m_k}^k]]$$

where $\hat{\zeta}$ comprises all positions that are assigned by κ states from $Q_{P,1} \times Q_{P,2}$, each ζ_j consists completely of labels $(\sigma, \vec{d} \cdot (0)^{s_2})$ and the corresponding positions of κ assign states from $Q_{P,1} \times Q_{\omega,2}$ (or similar the other way around), and each ζ_v^u has labels from Σ and gets assigned states from Q_ω. From this, we easily can reconstruct two trees ζ_1' and ζ_2' with $(\zeta_1')_\Sigma = (\zeta_2')_\Sigma = (\zeta)_\Sigma$ and accepting runs $\kappa_1' \in \mathrm{Run}_{\mathcal{A}_1}(\zeta_1')$ and $\kappa_2' \in \mathrm{Run}_{\mathcal{A}_2}(\zeta_2')$. ◄

▶ **Lemma 35.** *PMTA are closed under relabeling.*

Proof (sketch). Let Σ, Γ be alphabets, let $\mathcal{A} = (Q, \Xi, q_I, \Delta, \mathcal{F}, C)$ be a PMTA with $Q = Q_P \cup Q_\omega$, $\Xi = (\Sigma \times D) \cup \Sigma$, and $\Delta = \Delta_P \cup \Delta_\omega$, and let $\tau \colon \Sigma \to \mathcal{P}(\Gamma)$ be a relabeling. We construct the WPT3 $\mathcal{A}' = (Q, \Xi', q_I, \Delta', \mathcal{F}, C)$ with $\Xi' = (\Gamma \times D) \cup \Gamma$ and $\Delta' = \Delta_P' \cup \Delta_\omega'$ where

- $\Delta_P' = \{(q, (\gamma, \vec{d}), q_1, q_2) \mid (q, (\sigma, \vec{d}), q_1, q_2) \in \Delta_P, \gamma \in \tau(\sigma)\}$ and
- $\Delta_\omega' = \{(q, \gamma, q_1, q_2) \mid (q, \sigma, q_1, q_2) \in \Delta_\omega, \gamma \in \tau(\sigma)\}$.

◄

C.2 Correspondence with ωMSO⋈BAPA

▶ **Proposition 15.** *For any PMTA $\mathcal{A}$, there is an ωMSO⋈BAPA sentence φ with $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\varphi)$.*

Proof. This proof adopts (and slightly simplifies) the idea in [42, Thm. 10] how to logically encode counter values and the semilinear set C to our setting and uses the well-known construction to define the behavior of a Muller tree automaton by means of an MSO formula.

Let Σ be an alphabet, let $D \subseteq \mathbb{N}^s$ be finite for some $s \geq 1$, and let $\mathcal{A} = (Q, \Xi, q_I, \Delta, \mathcal{F}, C)$. W.l.o.g. assume $Q = \{1, \ldots, r\}$ for some $r \geq 1$. As usual, we use set variables $X_1, \ldots, X_r$ to encode a run of $\mathcal{A}$ on a tree ζ.

Let K be the maximal value occurring in vectors from D. Each position $i \in [s]$ of a vector $d = (d_1, \ldots, d_s) \in D$ carries a value $d_i \leq K$, in the following represented by the set variable $Z_i^{d_i}$. Thus, the presence of a variable $Z_i^{d_i}$ at a position x of ζ indicates that d_i has to be added to the ith counter to simulate the extended Parikh map $\Psi(\zeta)$.

Formally, let

$$\varphi := \exists Z_1^0 \ldots Z_1^K \ldots Z_s^K \exists X_1 \ldots X_r \forall P. \varphi_{\mathrm{part}_Q} \wedge \varphi_{\mathrm{part}_Z} \wedge \varphi_{\mathrm{run}} \wedge \varphi_{\mathrm{acc}} \wedge \varphi_C$$

where

- $\varphi_{\mathrm{part}_Q} := \forall x. \big(\bigwedge_{p,q \in Q, p \neq q} \neg (X_p(x) \wedge X_q(x)) \big)$

- $\varphi_{\mathrm{part}_Z} := \bigwedge_{1 \leq i \leq s} \big(\forall x. \bigwedge_{u,v \in \{0,\ldots,K\}, u \neq v} \neg (Z_i^u(x) \wedge Z_i^v(x)) \big)$

- $\varphi_{\mathrm{run}} := \forall x. \big((\varphi_{\mathrm{root}}(x) \Rightarrow X_{q_I}(x)) \wedge (\varphi_{\mathrm{trans}_P} \vee \varphi_{\mathrm{trans}_\omega}) \big)$

  - $\varphi_{\mathrm{trans}_P} := \bigvee_{(q,(\sigma,\vec{d}),q_0,q_1) \in \Delta_P} \big(X_q(x) \wedge \mathsf{P}_\sigma(x) \wedge X_{q_0}(x0) \wedge X_{q_1}(x1) \wedge \bigwedge_{1 \leq i \leq s} Z_i^{(\vec{d})_i}(x) \big)$

  - $\varphi_{\mathrm{trans}_\omega} := \bigvee_{(q,\sigma,q_0,q_1) \in \Delta_\omega} \big(X_q(x) \wedge \mathsf{P}_\sigma(x) \wedge X_{q_0}(x0) \wedge X_{q_1}(x1) \wedge \bigwedge_{\substack{1 \leq i \leq s \\ 0 \leq d \leq K}} \neg Z_i^d(x) \big)$

- $\varphi_{\mathrm{acc}} := \varphi_{\mathrm{path}}(P) \Rightarrow \bigvee_{F \in \mathcal{F}} \big(\bigwedge_{q \in F} \varphi_{\mathrm{inf}}^{\cap}(X_q, P) \wedge \bigwedge_{q \notin F} \neg \varphi_{\mathrm{inf}}^{\cap}(X_q, P) \big)$

and φ_C encodes the Parikh condition as follows. By definition of C, there exist some $k, \ell \geq 1$ and linear polynomials $p_1, \ldots, p_k \colon \mathbb{N}^\ell \to \mathbb{N}^s$ such that C is the union of the images of $p_1, \ldots, p_k$. Assume $p_g(m_1, \ldots, m_l) = \vec{v}_0 + m_1 \vec{v}_1 + \ldots + m_l \vec{v}_l$ with $\vec{v}_j = (v_{j,1}, \ldots, v_{j,s})$. Then, using number variables $m_1, \ldots, m_l$, we encode p_g by

$$\varphi_{p_g} := \exists m_1 \ldots \exists m_l. \bigwedge_{i=1}^s \Big(\textstyle\sum_{d=0}^K \mathsf{d}\# Z_i^d =_{\mathtt{fin}} \mathsf{v}_{0,i} + \mathsf{v}_{1,i} m_1 + \ldots + \mathsf{v}_{l,i} m_l \Big),$$

and let $\varphi_C := \big(\bigwedge_{i=1}^s \bigwedge_{d=0}^K \forall x. \neg Z_i^d(x) \big) \vee \varphi_{p_1} \vee \ldots \vee \varphi_{p_k}$. This finishes the construction of the overall sentence specifying $\mathcal{L}(\mathcal{A})$, which can be easily shown to be in $\omega\mathrm{MSO}\bowtie\mathrm{BAPA}$. ◄

▶ **Lemma 16.** *For each Parikh constraint χ there is a PMTA $\mathcal{A}$ with $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\chi)$.*

Proof. We assume w.l.o.g. that χ is of the form $\mathtt{c} + \sum_{i \in [r]} \mathtt{c_i} \# X_i \leq_{\mathtt{fin}} \mathtt{d} + \sum_{j \in [k]} \mathtt{d_j} \# Y_j$ where all X_i are pairwise distinct, and all Y_j likewise. Given a subset $\theta \subseteq \mathrm{free}(\chi)$, we denote by $|\theta|_X$ the number $\sum_{X_i \in \theta} c_i$ (and similar for $|\theta|_Y$). Then, assuming $\xi(\varrho) = (\sigma_\varrho^\xi, \theta_\varrho^\xi)$, we get

$$\mathcal{L}(\chi) = \{\xi \in T^\omega_{\Sigma_{\mathrm{free}(\chi)}} \mid c + \textstyle\sum_{\varrho \in \mathrm{pos}(\xi)} |\theta_\varrho^\xi|_X \leq d + \sum_{\varrho \in \mathrm{pos}(\xi)} |\theta_\varrho^\xi|_Y < \infty\}$$

and, by the condition $< \infty$, both sums have finite support. Thus, for each $\xi \in \mathcal{L}(\chi)$, there is a non-empty, finite, prefix-closed $Z \subset \{0,1\}^*$ that comprises all positions holding variable assignments and for which $\xi|_Z$ satisfies χ; if Z is a minimal such set, we denote $\xi|_Z$ in the following as ξ_χ. This can be tested by a PMTA as follows.

Let $D = \{(i,j) \mid 0 \leq i \leq \sum_{l \in [r]} c_l, 0 \leq j \leq \sum_{l \in [k]} d_l\}$. We construct the PMTA $\mathcal{A} = (\{q_I, q_f\}, \Xi, q_I, \Delta, \{\{q_f\}\}, C)$ with $\Xi = (\Sigma_{\mathrm{free}(\chi)} \times D) \cup \Sigma_{\mathrm{free}(\chi)}$, $\Delta = \Delta_P \cup \Delta_\omega$ where

- $\Delta_P = \{(q_I, ((\sigma,\theta), (|\theta|_X, |\theta|_Y)), q, q) \mid (\sigma, \theta) \in \Sigma_{\mathrm{free}(\chi)}, q \in \{q_I, q_f\}\}$ and
- $\Delta_\omega = \{(q_f, (\sigma, \emptyset), q_f, q_f) \mid \sigma \in \Sigma\}$

and $C = \{(z_1, z_2) \mid \mathtt{c} + z_1 \leq_{\mathtt{fin}} \mathtt{d} + z_2\}$.[12]

$\mathcal{L}(\chi) \subseteq \mathcal{L}(\mathcal{A})$: Let $\xi \in \mathcal{L}(\chi)$. We prove that there exists a tree $\zeta \in T^\omega_\Xi$ with $(\zeta)_{\Sigma_{\mathrm{free}(\chi)}} = \xi$ and an accepting run κ of $\mathcal{A}$ on ζ. Let ζ be such that, for each position ϱ,

$$\zeta(\varrho) = \begin{cases} ((\sigma, \theta), (|\theta|_X, |\theta|_Y)) & \text{if } \varrho \in \mathrm{pos}\{\xi_\chi\}, \xi(\varrho) = (\sigma, \theta) \\ \xi(\varrho) & \text{otherwise} \end{cases}.$$

---

[12] Note that by Lemma 11 we can use this description for a semilinear set.

Clearly, $(\zeta)_{\Sigma_{\text{free}(\chi)}} = \xi$ and $\text{pos}(\zeta_{\text{cnt}}) = \text{pos}(\xi_\chi)$. Moreover, define $\kappa \in T^\omega_{\{q_I, q_f\}}$ such that

$$\kappa(\varrho) = \begin{cases} q_I & \text{if } \varrho \in \text{pos}\{\xi_\chi\} \\ q_f & \text{otherwise} \end{cases}$$

for each position ϱ. Due to the construction of Δ, κ is compatible with the transitions of $\mathcal{A}$ in connection with ζ. As $\varepsilon \in \text{pos}\{\xi_\chi\}$ and, thus, $\kappa(\varepsilon) = q_I$, κ is a run of $\mathcal{A}$ on ζ.

It remains to argue that κ is accepting. We can calculate, by assuming $\zeta(\varrho) = (\sigma_\varrho, \theta_\varrho)$,

$$\Psi(\zeta_{\text{cnt}}) = \sum_{\varrho \in \text{pos}(\zeta_{\text{cnt}})} (\zeta_{\text{cnt}}(\varrho))_D = \sum_{\varrho \in \text{pos}(\zeta_{\text{cnt}})} (|\theta_\varrho|_X, |\theta_\varrho|_Y) = \Big(\sum_{\varrho \in \text{pos}(\zeta_{\text{cnt}})} |\theta_\varrho|_X, \sum_{\varrho \in \text{pos}(\zeta_{\text{cnt}})} |\theta_\varrho|_Y \Big) .$$

As $\text{pos}(\zeta_{\text{cnt}}) = \text{pos}(\xi_\chi)$ and we know from ξ_χ that $c + \sum_{\varrho \in \text{pos}(\xi)} |\theta_\varrho|_X \le d + \sum_{\varrho \in \text{pos}(\xi)} |\theta_\varrho|_Y$, we obtain $\Psi(\zeta_{\text{cnt}}) \in C$. Finally, as ξ_χ is finite, we obtain for each path π that $\inf(\kappa(\pi)) = \{q_f\}$. Thus, κ is accepting and $\xi \in \mathcal{L}(\mathcal{A})$.

$\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\psi)$: Let $\xi \in \mathcal{L}(\mathcal{A})$. As $\mathcal{A}$ recognizes all assignments of variables with transitions from Δ_P, it is clear that each variable is counted finitely often in ξ. Moreover, Δ_P only contains transitions where the counter values fit to the variable occurrences of each label in ξ. Thus, as $\Psi(\zeta_{\text{cnt}}) \in C$ for each ζ with $(\zeta)_{\Sigma_{\text{free}(\chi)}} = \xi$ and $\text{Run}_{\mathcal{A}}(\zeta) \ne \emptyset$, we obtain that $c + \sum_{\varrho \in \text{pos}(\xi)} |\theta_\varrho|_X \le d + \sum_{\varrho \in \text{pos}(\xi)} |\theta_\varrho|_Y$ also holds for ξ_χ. Thus, $\xi \in \mathcal{L}(\psi)$. ◄

▶ **Proposition 17.** *For every $\omega\text{MSO}\bowtie\text{BAPA}$ formula φ there is a PMTA $\mathcal{A}$ with $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\varphi)$.*

Proof. Let φ be an $\omega\text{MSO}\bowtie\text{BAPA}$ formula. By Theorem 10, we can assume that φ is in tree normal form, i.e., of the form $\exists X_1. \cdots \exists X_n. \bigvee_{i=1}^{k} \big(\varphi_i \wedge \bigwedge_{j=1}^{l_i} \chi_{i,j} \big)$, where φ_i are plain MSO sentences and the $\chi_{i,j}$ are (unnegated) Parikh constraints.. Then we proceed by induction on the (now restricted) structure of φ.

For the induction base, we consider the following cases: If φ is an MSO-formula, then, by [52], $\mathcal{L}(\varphi)$ can be recognized by an MTA. As MTA are an instance of PMTA, we obtain that $\mathcal{L}(\varphi)$ is PMTA-recognizable. Now let φ be a Parikh constraint. By Lemma 16 we obtain that $\mathcal{L}(\varphi)$ is PMTA-recognizable.

Now assume that φ is of the form $\varphi_1 \wedge \varphi_2$. Then $\mathcal{L}(\varphi) = \mathcal{L}_{\text{free}(\varphi)}(\varphi_1) \cap \mathcal{L}_{\text{free}(\varphi)}(\varphi_2)$ and, by induction hypothesis, $\mathcal{L}(\varphi_1)$ and $\mathcal{L}(\varphi_1)$ both are PMTA-recognizable. But then also $\mathcal{L}_{\text{free}(\varphi)}(\varphi_1)$ and $\mathcal{L}_{\text{free}(\varphi)}(\varphi_1)$ are PMTA-recognizable[13], respectively. Finally, by Proposition 14, $\mathcal{L}_{\text{free}(\varphi)}(\varphi_1) \cap \mathcal{L}_{\text{free}(\varphi)}(\varphi_2)$ is PMTA-recognizable. With a similar argumentation and with help of Proposition 14 we obtain that $\varphi_1 \vee \varphi_2$ is PMTA-recognizable.

Now assume that φ is of the form $\exists X.\varphi_1$. In the usual way, we define the relabeling $\tau \colon \Sigma_{\text{free}(\varphi_1) \cup \{X\}} \to \mathcal{P}(\Sigma_{\text{free}(\varphi)})$ given by $\tau(\sigma, \theta) = \{(\sigma, \theta \cap \text{free}(\varphi))\}$ and observe that $\mathcal{L}(\varphi) = \tau(\mathcal{L}_{\text{free}(\varphi_1) \cup \{X\}}(\varphi_1))$. By induction hypothesis, using the same argumentation as above, and by Proposition 14, we obtain that $\exists X.\varphi_1$ is PMTA-recognizable. ◄

C.3 Parikh-Muller Automata on Words correspond to Reachability-Regular Parikh Automata

The set of all infinite words over alphabet Σ will be denoted by Σ^ω; we use Σ^* to denote the set of all finite words over Σ.

We will use the projections $\cdot_\Sigma$ and $\cdot_D$ as well as the extended Parikh map Ψ with their obvious restrictions to word domains.

---

[13] It can be shown by a standard construction involving a relabeling that, for each $\mathcal{V} \supseteq \text{free}(\varphi)$, $\mathcal{L}(\varphi)$ is recognizable iff $\mathcal{L}_{\mathcal{V}}(\varphi)$ is recognizable.

▶ **Definition 36** (Parikh ω-Word Automaton). *Let Σ be an alphabet, let $s \in \mathbb{N} \setminus \{0\}$, let $D \subseteq \mathbb{N}^s$ be finite, and denote $(\Sigma \times D) \cup \Sigma$ by Ξ. A Parikh Word Automaton (of* dimension *s) is a tuple $\mathcal{A} = (Q, \Xi, q_I, \Delta, \mathcal{F}, C)$ where $Q = Q_P \cup Q_\omega \cup \{q_I\}$ is a finite set of* states *with Q_P, Q_ω disjoint and q_I being the* initial state, *$\Delta = \Delta_P \cup \Delta_\omega$ is the* transition relation *with*

$$\Delta_P \subseteq (Q_P \cup \{q_I\}) \times (\Sigma \times D) \times Q \quad and \quad \Delta_\omega \subseteq (Q_\omega \cup \{q_I\}) \times \Sigma \times Q_\omega,$$

$\mathcal{F}$ is the acceptance condition *given by either*
- *a set of* final state sets *$\mathcal{F} \subseteq 2^{Q_\omega}$ (Muller acceptance) or*
- *a set of* final states *$\mathcal{F} \subseteq Q_\omega$ (Büchi acceptance),*
and $C \subseteq \mathbb{N}^s$ is a semilinear set named final constraint.

A Parikh ω-word automaton that uses a Muller acceptance condition will be called a *Parikh-Muller word automaton* (PMWA); if it instead uses a Büchi acceptance condition it will be referred to as *Parikh-Büchi word automaton* (PBWA). We note that, by choosing $\Delta_P = \emptyset$, we reobtain the concept of a Muller (respectively Büchi) word automaton.

▶ **Definition 37** (Semantics of PMWA and PBWA). *A run of $\mathcal{A}$ on a word $w \in \Xi^\omega$ is a word $\kappa_w \in Q^\omega$ whose initial position carries q_I and which respects Δ jointly with w. By definition of Δ, if a run exists, then w is of the form $w_{cnt}w'$ with $w_{cnt} \in (\Sigma \times D)^*$ and $w' \in \Sigma^\omega$. A run κ_w is* accepting *if*
1. *either $\inf(\kappa_w) \cap \mathcal{F} \neq \emptyset$ (Büchi acceptence) or $\inf(\kappa_w) \in \mathcal{F}$ (for Muller acceptance) and*
2. *if $w_{cnt} \neq \varepsilon$, then $\Psi(w_{cnt}) \in C$.*
Note that, by the first condition, κ_w being accepting implies finiteness of w_{cnt} and, thus, well-definedness of the sum in $\Psi(w_{cnt})$. The set of all accepting runs of $\mathcal{A}$ on w will be denoted by $\mathrm{Run}_\mathcal{A}(w)$. Then, the word language *of $\mathcal{A}$, denoted by $\mathcal{L}(\mathcal{A})$, is the set*

$$\mathcal{L}(\mathcal{A}) = \{u \in \Sigma^\omega \mid \exists w \in \Xi^\omega \text{ with } \mathrm{Run}_\mathcal{A}(w) \neq \emptyset \text{ and } w_\Sigma = u\}.$$

Now we recall the model of reachability-regular Parikh automata from Grobler et al. [34].

▶ **Definition 38** (Reachability-Regular Parikh Automata). *Let Σ be an alphabet and $s \in \mathbb{N} \setminus \{0\}$. A reachability-regular Parikh automaton (RRPA) of dimension s is a tuple $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F, C)$ where Q is a finite set of states, $q_0 \in Q$ is the initial state, $F \subseteq Q$ is the set of final states, $\Delta \subseteq Q \times \Sigma \times \mathbb{N}^s \times Q$ is a finite set of transitions and $C \subseteq \mathbb{N}^s$ is a semilinear set.*

▶ **Definition 39** (Semantics of RRPA). *A run of $\mathcal{A}$ on an infinite word $w = w_1 w_2 w_3 ...$ is an infinite sequence $\tau = r_1 r_2 r_3 ...$ of transitions $r_i = (p_{i-1}, w_i, d_i, p_i)$ such that $p_0 = q_0$. We say that τ is accepting if there is an $i \geq 1$ such that $p_i \in F$ and $d_1 + ... + d_i \in C$, and there are infinitely many $j \geq 1$ such that $p_j \in F$. The set of all accepting runs of $\mathcal{A}$ on w will be denoted by $\mathrm{Run}_\mathcal{A}(w)$. Then, the* word language *of $\mathcal{A}$, denoted by $\mathcal{L}(\mathcal{A})$, is the set*

$$\mathcal{L}(\mathcal{A}) = \{w \in \Sigma^\omega \mid \mathrm{Run}_\mathcal{A}(w) \neq \emptyset\}.$$

The goal is to show the following theorem:

▶ **Theorem 40.** *Let $\mathcal{L} \subseteq \Sigma^\omega$. Then the following are equivalent:*
1. *$\mathcal{L} = \mathcal{L}(\mathcal{A})$ for some PMWA $\mathcal{A}$,*
2. *$\mathcal{L} = \mathcal{L}(\mathcal{A})$ for some PBWA $\mathcal{A}$, and*
3. *$\mathcal{L} = \mathcal{L}(\mathcal{A})$ for some RRPA $\mathcal{A}$.*

Proof. The theorem immediately follows from Lemma 41, Lemma 42, and Theorem 48. ◀

From RRPA to PBWA

▶ **Lemma 41.** *Given an RRPA $\mathcal{A}$, we can construct a PBWA $\mathcal{A}'$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}')$.*

Proof. Let $\mathcal{A} = (Q, \Sigma, q_0, \Delta, F, C)$ be an RRPA. We construct a PBWA $\mathcal{A}'$ with the following intuition. We duplicate the state space of $\mathcal{A}$ to obtain Q_P and Q_ω. Everytime $\mathcal{A}'$ reaches a final state from $\mathcal{A}$ it can nondeterministically choose if it stays in Q_P and proceeds with counting or if it goes into Q_ω and reads the remaining word without counting.

Formally, we let $D = \{d \in \mathbb{N}^s \mid (q, w, d, q') \in \Delta\}$ and define $\mathcal{A}' = (Q', (\Sigma \times D) \cup \Sigma, q_I, \Delta_P \cup \Delta_\omega, F', C)$ where

- $Q' = Q_P \cup Q_\omega \cup \{q_I\}$ with
  - $Q_P = \{q^P \mid q \in Q\}$ and
  - $Q_\omega = \{q^\omega \mid q \in Q\}$,
- $\Delta_P = \Delta_I \cup \Delta'$ with
  - $\Delta_I = \{(q_I, (\sigma, d), q^P) \mid (q_0, \sigma, d, q) \in \Delta\} \cup \{(q_I, (\sigma, d), q^\omega) \mid (q_0, \sigma, d, q) \in \Delta, q \in F\}$, and
  - $\Delta' = \{(q^P, (\sigma, d), p^P) \mid (q, \sigma, d, p) \in \Delta\} \cup \{(q^P, (\sigma, d), p^\omega) \mid (q, \sigma, d, p) \in \Delta, p \in F\}$,
- $\Delta_\omega = \{(q^\omega, \sigma, p^\omega) \mid (q, \sigma, d, p) \in \Delta\}$,
- $F' = \{q^\omega \mid q \in F\}$.

$\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{A}')$: Let $\tau = r_1 r_2 r_3 \dots$ with $r_i = (p_{i-1}, w_i, d_i, p_i)$ be an accepting run of $\mathcal{A}$ on $w = w_1 w_2 \dots$. Then there is a position $i \geq 1$ such that $p_i \in F$ and $d_1 + \dots + d_n \in C$. Now let $\kappa = q_I \, p_1^P \dots p_{i-1}^P \, p_i^\omega \, p_{i+1}^\omega \dots$. By construction, κ is a run of $\mathcal{A}'$ on $w' = (w_1, d_1) \dots (w_i, d_i) w_{i+1} w_{i+2} \dots$. Moreover, as $\Psi((w_1, d_1) \dots (w_i, d_i)) = d_1 + \dots + d_i \in C$ and $p_j^\omega \in F$ iff $p_j \in F$, we obtain that κ is accepting. Finally, as $w'_\Sigma = w$, $w \in \mathcal{L}(\mathcal{A}')$.

$\mathcal{L}(\mathcal{A}') \subseteq \mathcal{L}(\mathcal{A})$: Similar argumentation as above, "in reverse". ◀

From PBWA to RRPA

▶ **Lemma 42.** *Given a PBWA $\mathcal{A}$, we can construct an RRPA $\mathcal{A}'$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}')$.*

Proof. Let $\mathcal{A} = (Q, \Xi, q_I, \Delta, \mathcal{F}, C)$ be a PBWA of dimension s. We construct a RRBA $\mathcal{A}'$ where we enrich the set of final states by copies of those states that are visited by $\mathcal{A}$ when entering Q_ω. In order to keep apart runs that use transitions from Δ_P and runs which directly start in Δ_ω, we use one additional counter. The transitions of Δ_ω are simulated by $\mathcal{A}'$ by using the zero-vector $\vec{0} = (0)^{s+1}$. Formally, let $\mathcal{A}' = (Q', \Sigma, q_I, \Delta', F', C')$ where

- $Q' = Q \cup \{\hat{q} \mid q \in Q_\omega\}$,
- $F' = \mathcal{F} \cup \{\hat{q} \mid q \in Q_\omega\}$,
- $\Delta' = \Delta'_P \cup \Delta'_\omega$ with
  - $\Delta'_P = \{(q, \sigma, d \cdot (0), q') \mid (q, (\sigma, d), q') \in \Delta_P, q' \notin Q_\omega\} \cup \{(q, \sigma, d \cdot (1), \hat{q}') \mid (q, (\sigma, d), q') \in \Delta_P, q' \in Q_\omega\}$ and
  - $\Delta'_\omega = \{(\hat{q}, \sigma, \vec{0}, q') \mid (q, \sigma, q') \in \Delta_\omega\} \cup \{(q, \sigma, \vec{0}, q') \mid (q, \sigma, q') \in \Delta_\omega\}$, and
- $C' = C \cdot (1) \cup C_0$ with $C_0 = \{\vec{0}\}$ if $(q_I, \sigma, q) \in \Delta_\omega$ for some σ, q and $C_0 = \emptyset$ otherwise.

$\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{A}')$: Let $w \in \mathcal{L}(\mathcal{A})$. Then there is some $u \in \Xi^\omega$ with $(u)_\Sigma = w$ and there is some accepting run κ of $\mathcal{A}$ on u. Assume that $\kappa = q_I q_1 q_2 \dots$.

If $u_{\text{cnt}} \neq \emptyset$, there is some $i \geq 1$ such that $q_j \in Q_\omega$ for each $j \geq i$ and $q_l \in Q_P$ for each $1 \leq l < i$. Moreover, u is of the form $(\sigma_1, d_1) \dots (\sigma_i, d_i) \sigma_{i+1} \sigma_{i+2} \dots$. Let $\tau = r_1 r_2 \dots$ such

that for each $k \geq 1$

$$
r_k = \begin{cases}
(q_I, \sigma_1, d_1 \cdot (0), q_1) & \text{if } k = 1, i \neq 1 \\
(q_I, \sigma_1, d_1 \cdot (1), \hat{q}_1) & \text{if } k = i = 1 \\
(q_{k-1}, \sigma_k, d_k \cdot (0), q_k) & \text{if } 1 < k < i \\
(q_{i-1}, \sigma_i, d_i \cdot (1), \hat{q}_i) & \text{if } k = i, i > 1 \\
(\hat{q}_i, \sigma_{i+1}, \vec{0}, q_{i+1}) & \text{if } k = i + 1 \\
(q_{k-1}, \sigma_k, \vec{0}, q_k) & \text{if } k > i + 1 .
\end{cases}
$$

By construction, each r_k is in Δ'. It is not hard to see that τ is an accepting run of $\mathcal{A}'$ on w: as $d_1 \cdot (0) + \ldots + d_i \cdot (1) = \Psi(u_{\mathrm{cnt}}) \cdot (1)$, $\Psi(u_{\mathrm{cnt}}) \cdot (1) \in C \cdot (1)$, and $\hat{q}_i \in F'$, the first condition is satisfied. Moreover, as in $r_{i+2} r_{i+3} \ldots$ the same states occur infinitely often as in the sequence $q_{i+2} q_{i+3} \ldots$, by construction of F' the Büchi condition is satisfied as well.

If, on the other hand, $u_{\mathrm{cnt}} = \emptyset$, then $q_i \in Q_\omega$ for each $i \geq 1$ and $u = w = \sigma_1 \sigma_2 \ldots$ for some $\sigma_i \in \Sigma$. Then construct $\tau = r_1 r_2 \ldots$ with $r_1 = (q_I, \sigma_1, \vec{0}, q_1)$ and $r_j = (q_{j-1}, \sigma_j, \vec{0}, q_j)$ for each $j > 1$. By construction, $r_i \in \Delta'$ for each $i \geq 1$. Moreover, as τ uses the same states infinitely often as κ, by construction of F' the Büchi acceptance condition is satisfied. Finally, as the extended Parikh image is $\vec{0}$ at each position of τ and, by construction, $\vec{0} \in C_0$, τ is an accepting run of $\mathcal{A}'$ on w.

$\mathcal{L}(\mathcal{A}') \subseteq \mathcal{L}(\mathcal{A})$: As the extended Parikh image of $\mathcal{A}'$ does not change anymore after entering the states of Q_ω, we can, as above, construct from each accepting run τ of $\mathcal{A}'$ a corresponding accepting run κ of $\mathcal{A}$. ◀

From PBWA to PMWA and back

By using standard constructions to transform a Büchi automaton into a Muller automaton and vice versa, we obtain the expressive equivalence of PBWA and PMWA. As any Büchi acceptance condition can be immediately expressed by a Muller acceptance condition, the first direction is straight forward.

▶ **Lemma 43.** *For each PBWA $\mathcal{A}$ there is a PMWA $\mathcal{A}'$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}')$.*

Proof. Let $\mathcal{A} = (Q, \Xi, q_I, \Delta, \mathcal{F}, C)$ be a PBWA. In the usual way (cf. [57, Prop. 5.3]), we construct the PMWA $\mathcal{A}' = (Q, \Xi, q_I, \Delta, \mathcal{F}', C)$ where $\mathcal{F}' = \{F' \subseteq Q_\omega \mid F' \cap \mathcal{F} \neq \emptyset\}$. Then $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}')$. ◀

For the other direction, we decompose a given PMWA into a number of Parikh (word) automata and Muller (word) automata (compare with the proof of Theorem 20), transform the Muller automata into equivalent Büchi automata (which well-known to be possible in the word case), and finally reverse our decomposition, obtaining an equivalent PBWA.

▶ **Lemma 44.** *For each PMWA $\mathcal{A}$ there is a PBWA $\mathcal{A}'$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}')$.*

Proof. Let $\mathcal{A} = (Q, \Xi, q_I, \Delta, \mathcal{F}, C)$ with $Q = Q_P \cup Q_\omega \cup \{q_I\}$ and $\Delta = \Delta_P \cup \Delta_\omega$ be a PMWA. Let $Q_{\omega,r} = \{q \in Q_\omega \mid (p, (\sigma, d), q) \in \Delta_P \text{ for some } p \in Q_P, (\sigma, d) \in \Sigma \times D\}$. For each $q \in Q_{\omega,r}$ we construct a pair $(\mathcal{A}_q^P, \mathcal{A}_q^M)$ consisting of a Parikh word automaton[14] $\mathcal{A}_q^P$ and a Muller word automaton $\mathcal{A}_q^M$ as follows:

---

[14] For a definition of Parikh word automata (over finite words), we refer the reader to [37, page 4].

- We let $\mathcal{A}_q^P = (Q_P \cup \{q_I, q\}, \Sigma \times D, q_I, \Delta_q, \{q\}, C)$ with $\Delta_q = \{(p, (\sigma, d), p') \in \Delta_P \mid p' \notin Q_\omega$ or $p' = q\}$.
- We let $\mathcal{A}_q^M = (Q_\omega, \Sigma, q, \Delta_\omega, \mathcal{F})$.

Moreover, we construct the Muller word automaton $\mathcal{A}_{q_I}^M = (Q_\omega \cup \{q_I\}, \Sigma, q_I, \Delta_\omega, \mathcal{F})$.

▷ **Claim 45.** $\mathcal{L}(\mathcal{A}) = (\bigcup_{q \in Q_{\omega,r}} \mathcal{L}(\mathcal{A}_q^P) \cdot \mathcal{L}(\mathcal{A}_q^M)) \cup \mathcal{L}(\mathcal{A}_{q_I}^M)$

Proof. $\subseteq$: Let $w \in \mathcal{L}(\mathcal{A})$. Then there is some $u \in \Xi^\omega$ with $(u)_\Sigma = w$ and $\kappa \in \mathrm{Run}_\mathcal{A}(u)$. If $u_{\mathrm{cnt}} = \emptyset$, by construction, $\kappa \in \mathrm{Run}_{\mathcal{A}_{q_I}^M}(u)$. If $u_{\mathrm{cnt}} \neq \emptyset$, κ is of the form $q_I q_1 q_2 \dots$ and there is some $i \geq 1$ such that $q_j \in Q_\omega$ for each $j \geq i$ and $q_l \in Q_P$ for each $1 \leq l < i$. Moreover, u is of the form $(\sigma_1, d_1) \dots (\sigma_i, d_i) \sigma_{i+1} \sigma_{i+2} \dots$. By construction of $\mathcal{A}_{q_i}^P$, $\kappa' = q_I \dots q_i$ is a run of $\mathcal{A}_{q_i}^P$ on $(\sigma_1, d_1) \dots (\sigma_i, d_i)$ and, as $d_1 + \dots + d_i$ is in C, κ' is accepting. Thus, $\sigma_1 \dots \sigma_i \in \mathcal{L}(\mathcal{A}_{q_i}^P)$. Similarly, by construction of $\mathcal{A}_{q_i}^M$, $q_i q_{i+1} \dots$ is an accepting run of $\mathcal{A}_{q_i}^M$ on $\sigma_{i+1} \sigma_{i+2} \dots$. Thus, $w \in \mathcal{L}(\mathcal{A}_{q_i}^P) \cdot \mathcal{L}(\mathcal{A}_{q_i}^M)$.

$\supseteq$: If $w \in \mathcal{L}(\mathcal{A}_{q_I}^M)$ we directly obtain that $w \in \mathcal{L}(\mathcal{A})$ as the transitions of $\mathcal{A}_{q_I}^M$ are a subset of the transitions of $\mathcal{A}$. Now let $w \in \mathcal{L}(\mathcal{A}_q^P) \cdot \mathcal{L}(\mathcal{A}_q^M)$ for some $q \in Q_{\omega,r}$, i.e., $w = w_1 w_2$ for some $w_1 \in \mathcal{L}(\mathcal{A}_q^P)$ and $w_2 \in \mathcal{L}(\mathcal{A}_q^M)$. As each run κ_1 of $\mathcal{A}_q^P$ on some u_1 with $(u_1)_\Sigma = w_1$ ends in the state q and each run κ_2 of $\mathcal{A}_q^M$ on w_2 starts in q, we can easily combine κ_1 and κ_2 to a run κ of $\mathcal{A}$ on $u_1 w_2$. ◄

Now let $q \in Q_{\omega,r} \cup \{q_I\}$. Let $\mathcal{A}_q^B = (Q_{B,q}, \Sigma, p_{B,q}, \Delta_{B,q}, \mathcal{F}_{B,q})$ be the Büchi word automaton equivalent to $\mathcal{A}_q^M$, obtained by the standard construction from Muller to Büchi word automata [57, Prop. 5.3]. Clearly, $\mathcal{L}(\mathcal{A}_q^P) \cdot \mathcal{L}(\mathcal{A}_q^M) = \mathcal{L}(\mathcal{A}_q^P) \cdot \mathcal{L}(\mathcal{A}_q^B)$ for each $q \in Q_{\omega,r}$.

In the next step, for each $q \in Q_{\omega,r}$ we compose $(\mathcal{A}_q^P, \mathcal{A}_q^B)$ into one PBWA $\hat{\mathcal{A}}_q$. W.l.o.g. we assume that $(Q_P \cup \{q_I\}) \cap Q_{B,q} = \emptyset$ and we let $\hat{\mathcal{A}}_q = (Q_P \cup Q_{B,q} \cup \{q_I\}, \Xi, q_I, \Delta_{PB}, \mathcal{F}_{B,q}, C)$ where $\Delta_{PB} = \{(p, (\sigma, d), p') \in \Delta_P \mid p' \notin Q_\omega\} \cup \{(p, (\sigma, d), p_{B,q}) \mid (p, (\sigma, d), p') \in \Delta_P, p' = q\} \cup \Delta_{B,q}$.

▷ **Claim 46.** $\mathcal{L}(\mathcal{A}_q^P) \cdot \mathcal{L}(\mathcal{A}_q^B) = \mathcal{L}(\hat{\mathcal{A}}_q)$

Proof. $\subseteq$: Let $w \in \mathcal{L}(\mathcal{A}_q^P) \cdot \mathcal{L}(\mathcal{A}_q^B)$. Then $w = w_1 w_2$ with $w_1 \in \mathcal{L}(\mathcal{A}_q^P)$ and $w_2 \in \mathcal{L}(\mathcal{A}_q^B)$, there is some $u \in (\Sigma \times D)^*$ with $(u)_\Sigma = w_1$ and $\kappa_1 \in \mathrm{Run}_{\mathcal{A}_q^P}(u)$, and there is some $\kappa_2 \in \mathrm{Run}_{\mathcal{A}_q^B}(w_2)$. By inspecting the construction of $\mathcal{A}_q^P$ and $\mathcal{A}_q^B$, we obtain that κ_1 is of the form $q_I q_1 \dots q_i$ with $q_i = q$ and κ_2 is of the form $p_1 p_2 \dots$ with $p_1 = p_{B,q}$. Then we construct the run $\kappa = q_I q_1 \dots q_{i-1} p_1 p_2 \dots$ which, by definition of $\hat{\mathcal{A}}_q$, respects Δ_{PB} jointly with $u w_2$. Moreover, as κ inherits the extended Parikh image from κ_1 and the states occurring infinitely often from κ_2, κ is accepting.

$\supseteq$: Let $w \in \mathcal{L}(\hat{\mathcal{A}}_q)$. Then, similar to above, there must be a decomposition $w = w_1 w_2$ and a run $\kappa \in \mathrm{Run}_{\hat{\mathcal{A}}_q}(u w_2)$ for some $u \in (\Sigma \times D)^*$ with $(u)_\Sigma = w_1$. This run can be easily decomposed into a run $\kappa_1 \in \mathrm{Run}_{\mathcal{A}_q^P}(u)$ (with the same extended Parikh image), and a run $\kappa_2 \in \mathrm{Run}_{\mathcal{A}_q^B}(w_2)$ (with the same states occurring infinitely often), witnessing $w_1 \in \mathcal{L}(\mathcal{A}_q^P)$ as well as $w_2 \in \mathcal{L}(\mathcal{A}_q^B)$. ◄

Moreover, we construct the PBWA $\hat{\mathcal{A}}_{q_I} = (Q_{B,q_I} \cup \{q_0\}, \Sigma, q_0, \Delta', \mathcal{F}_{B,q}, \emptyset)$ where $\Delta' = \Delta_{B,q} \cup \{(q_0, \sigma, q) \mid (p_{B,q}, \sigma, q) \in \Delta_{B,q}\}$.

▷ **Claim 47.** $\mathcal{L}(\mathcal{A}_{q_I}^B) = \mathcal{L}(\hat{\mathcal{A}}_{q_I})$

Proof. As Büchi word automata are a special case of PBWA, it is not hard to see that $\mathcal{L}(\mathcal{A}_{q_I}^B) = \mathcal{L}(\hat{\mathcal{A}}_{q_I})$. As PBWA do not allow to use their initial state more than once, we introduce the new initial state q_0. ◄

By taking together Claim 45, Claim 46, and Claim 47, we finally observe that $\mathcal{L}(\mathcal{A}) = \bigcup_{q \in Q_{\omega,r} \cup \{q_I\}} \mathcal{L}(\hat{\mathcal{A}}_q)$. As PBWA are closed under union (which follows either from adapting the proof of Proposition 14, or from the closure of RRPA under union [34, Lm. 2] together with the expressive equivalence of RRPA and PBWA we already established via Lemma 41 and Lemma 42), we obtain that the language $\bigcup_{q \in Q_{\omega,r} \cup \{q_I\}} \mathcal{L}(\hat{\mathcal{A}}_q)$ is recognizable by some PBWA. ◄

▶ **Theorem 48.** *PBWA and PMWA are expressively equivalent.*

C.4   Emptiness

▶ **Proposition 19** (based on [40, 28]). *Given a PTA $\mathcal{A}$, deciding $\mathcal{L}(\mathcal{A}) \neq \emptyset$ is NP-complete.*

Proof. Let $\mathcal{A} = (Q, \Sigma \times D, \delta, q_I, F, C)$ be a PTA of dimension s where all numbers are encoded in binary. W.l.o.g. assume that $D = \{d_1, \ldots, d_n\}$ with $d_i \in \mathbb{N}^s$. Moreover, let $\mathcal{A}' = (Q, \Sigma \times D, \delta, q_I, F)$ be a tree automaton[15] over $\Sigma \times D$. It is easy to observe that $\mathcal{L}(\mathcal{A}) \neq \emptyset$ iff $\Psi(\mathcal{L}(\mathcal{A}')) \cap C \neq \emptyset$. Our aim is to show that $\Psi(\mathcal{L}(\mathcal{A}))$ can be encoded by a Presburger formula.

Following and slightly adjusting [40, proof of Lm. 8.15], in a first step we construct a context-free grammar that generates all words obtained by "flattening" the trees in $\mathcal{L}(\mathcal{A}')$ and where the symbols from $\Sigma \times D$ are projected to their D-component. We let $G' = (Q, D, q_i, R)$ where R consists of the following rules:

- if $(q, (\sigma, d), q_1, q_2) \in \delta$, then $q \to q_1 d q_2 \in R$, and
- if $q \in F$, then $q \to \varepsilon \in R$.

We observe that $\Psi(\mathcal{L}(\mathcal{A}')) = \Psi(\mathcal{L}(G'))$ (where we sum on the right-hand side over string positions and assume that $(d)_D = d$).

Now we proceed as in [28, Proposition III.2.]: as the result used, to construct a Presburger formula for the language of an automaton, was stated for context-free grammars, it works in our more general setting as well.

By [59, Theorem 4] (which had originally a small mistake fixed in [38]) we can construct in linear time an existential Presburger formula $\varphi_{G'}^P(z_1, \ldots, z_n)$ encoding the Parikh image (counts of occurrences of each symbol) of $\mathcal{L}(G')$. Also due to [59, proof of Theorem 4], one can obtain from $\varphi_{G'}^P(z_1, \ldots, z_n)$ in polynomial time an existential[16] Presburger formula $\varphi_{G'}(y_1, \ldots, y_s)$ representing $\Psi(\mathcal{L}(G'))$. Moreover, by [59, Lemma II.1], an existential Presburger formula $\varphi_C(y_1, \ldots, y_s)$ describing C can be built in linear time. Finally, satisfiability of the Presburger sentence $\exists y_1, \ldots, y_s. \varphi_{G'}(y_1, \ldots, y_s) \wedge \varphi_C(y_1, \ldots, y_s)$ over $\mathbb{N}$ can be tested in NP [56]. NP-hardness follows directly from the word case. ◄

▶ **Theorem 20.** *Given a PMTA $\mathcal{A}$, deciding $\mathcal{L}(\mathcal{A}) \neq \emptyset$ is PSPACE-complete.*

Proof. Let $\mathcal{A} = (Q, \Xi, q_I, \Delta, \mathcal{F}, C)$ be a PMTA with $Q = Q_P \cup Q_\omega \cup \{q_I\}$, $\Xi = (\Sigma \times D) \cup \Sigma$, and $\Delta = \Delta_P \cup \Delta_\omega$. As each tree in the language of $\mathcal{A}$ can be decomposed into some finite tree over $\Sigma \times D$, on which the Parikh constraint is tested, and a number of infinite trees from T_Σ, we can reduce PMTA non-emptiness testing to deciding non-emptiness of Muller tree automata and Parikh tree automata. To this end, consider

---

[15] In our setting, a tree automaton $\mathcal{A}$ over $\Sigma \times D$ is defined to be a PTA with $C = \mathbb{N}^s$ which simplifies $\mathcal{L}(\mathcal{A})$ to $\{\xi \in T_{\Sigma \times D} \mid \mathrm{Run}_{\mathcal{A}}(\xi) \neq \emptyset\}$.

[16] This was not explicitly stated in [59, proof of Theorem 4], however, their proof is based on this assumption.

- the Muller tree automaton $\mathcal{A}_{q_I} = (Q_\omega \cup \{q_I\}, \Sigma, q_I, \Delta_\omega, \mathcal{F})$,
- the Muller tree automata $\mathcal{A}_q = (Q_\omega, \Sigma, q, \Delta_\omega, \mathcal{F})$ for all $q \in Q_\omega$, and
- the Parikh tree automaton $\mathcal{A}_P = (Q, \Sigma \times D, q_I, \Delta_P, F_P, C)$ with $F_P = \{q \in Q_\omega \mid \mathcal{L}(\mathcal{A}_q) \neq \emptyset\}$.

As deciding $\mathcal{L}(\mathcal{A}_q) \neq \emptyset$ is PSPACE-complete [52, 39], $\mathcal{A}_P$ can be constructed in PSPACE and, by Proposition 19, its non-emptiness can be decided in NP. Thus, the overall PSPACE complexity follows from the observation that $\mathcal{L}(\mathcal{A}) \neq \emptyset$ iff $\mathcal{L}(\mathcal{A}_{q_I}) \neq \emptyset$ or $\mathcal{L}(\mathcal{A}_P) \neq \emptyset$, shown next.

$\Rightarrow$: Assume that $\mathcal{L}(\mathcal{A}) \neq \emptyset$. Then there exist some $\xi \in T_\Sigma^\omega$, $\zeta \in T_\Xi^\omega$ with $(\zeta)_\Sigma = \xi$, and $\kappa \in \mathrm{Run}_\mathcal{A}(\zeta)$. We distinguish the following two cases:

Assume $(\zeta)_\Sigma = \zeta$. Then $\kappa(u) \in Q_\omega$ for each $u \in \mathrm{pos}(\zeta) \setminus \{\varepsilon\}$ and, hence, ζ is recognized by $\mathcal{A}$ only with transitions from Δ_ω. But then, $\kappa \in \mathrm{Run}_{\mathcal{A}_{q_I}}(\zeta)$ and, thus, $\mathcal{L}(\mathcal{A}_{q_I}) \neq \emptyset$.

Now assume $(\zeta)_\Sigma \neq \zeta$. Then ζ is of the form $\zeta_{\mathrm{cnt}}[\zeta_1, \ldots, \zeta_m]$ for some $m \in \mathbb{N}$ and, similarly, κ can be decomposed into $\bar{\kappa}[\kappa_1, \ldots, \kappa_m]$ with $\mathrm{pos}(\bar{\kappa}) = \mathrm{pos}(\zeta_{\mathrm{cnt}})$, $\bar{\kappa}$ is labeled with states from $Q_P \cup \{q_I\}$, and κ_i is labeled with states from Q_ω for each $i \in [m]$. As, for each $i \in [m]$ and each path π reaching κ_i, we have that $\inf(\kappa(\pi)) \in \mathcal{F}$, also $\inf(\kappa_i(\pi \setminus \bar{\kappa})) \in \mathcal{F}$ with $\pi \setminus \bar{\kappa}$ being the path obtained by cutting the initial part going through $\bar{\kappa}$ from π. Thus, $\kappa_i \in \mathrm{Run}_{\mathcal{A}_{\kappa_i(\varepsilon)}}(\zeta_i)$ and $\kappa_i(\varepsilon) \in F_P$. Moreover, we obtain by construction of $\mathcal{A}_P$ that $\kappa' = \bar{\kappa}[\kappa_1(\varepsilon), \ldots, \kappa_m(\varepsilon)]$ is a run of $\mathcal{A}_P$ on ζ_{cnt}. Finally, as $\Psi(\zeta_{\mathrm{cnt}}) \in C$, κ' is accepting and, thus, $(\zeta_{\mathrm{cnt}})_\Sigma \in \mathcal{L}(\mathcal{A}_P)$. Hence, $\mathcal{L}(\mathcal{A}_P) \neq \emptyset$.

$\Leftarrow$: First assume that $\mathcal{L}(\mathcal{A}_{q_I}) \neq \emptyset$. Then there exists some $\xi \in T_\Sigma^\omega$ and $\kappa \in \mathrm{Run}_{\mathcal{A}_{q_I}}(\xi)$. But then, by construction of $\mathcal{A}_{q_I}$, we also have $\kappa \in \mathrm{Run}_\mathcal{A}(\xi)$. Thus, $\mathcal{L}(\mathcal{A}) \neq \emptyset$.

Now assume $\mathcal{L}(\mathcal{A}_P) \neq \emptyset$. Then there is some $\xi \in T_\Sigma$, $\zeta \in T_{\Sigma \times D}$ with $(\zeta)_\Sigma = \xi$, and run $\kappa \in \mathrm{Run}_{\mathcal{A}_P}(\zeta)$. By construction of $\mathcal{A}_P$ we can decompose κ into $\bar{\kappa}[q_1, \ldots, q_m]$ for some $m \in \mathbb{N}$ such that $\mathrm{pos}(\bar{\kappa}) = \mathrm{pos}(\zeta)$, $\bar{\kappa}$ is labeled with states from $Q_P \cup \{q_I\}$, and $q_1, \ldots, q_m \in Q_\omega$. As κ is accepting, $q_1, \ldots, q_m \in F_P$ and, hence, $\mathcal{L}(\mathcal{A}_{q_i}) \neq \emptyset$ for each $i \in [m]$. Thus, there are trees $\xi_1, \ldots, \xi_m \in T_\Sigma^\omega$ and runs $\kappa_i \in \mathrm{Run}_{\mathcal{A}_{q_i}}(\xi_i)$ starting in q_i. But then, $\bar{\kappa}[\kappa_1, \ldots, \kappa_m]$ is also a run of $\mathcal{A}$ on $\zeta' = \zeta[\xi_1, \ldots, \xi_m]$. Obviously, $\zeta'_{\mathrm{cnt}} = \zeta$ and, thus, $\Psi(\zeta'_{\mathrm{cnt}}) \in C$. Finally, as $\inf(\kappa_i(\pi)) \in \mathcal{F}$ for each path π and this transfers to κ, we obtain that $\kappa \in \mathrm{Run}_\mathcal{A}(\zeta')$. Hence, $\xi[\xi_1, \ldots, \xi_m] \in \mathcal{L}(\mathcal{A})$ and $\mathcal{L}(\mathcal{A}) \neq \emptyset$. ◀

D Details on MSO-Interpretations

▶ **Lemma 23.** *Let $\mathcal{I}$ be an MSO-interpretation. Then, for every ωMSO⋈BAPA sentence φ over $\mathbb{S}$ one can compute an ωMSO⋈BAPA sentence $\varphi^{\mathcal{I}}$ over $\mathbb{S}'$ satisfying $\mathfrak{A} \models \varphi^{\mathcal{I}} \iff \mathfrak{B} \models \varphi$ for every $\mathbb{S}'$-structure $\mathfrak{A}$ and $\mathbb{S}$-structure $\mathfrak{B}$ with $\mathcal{I}(\mathfrak{A}) \cong \mathfrak{B}$.*

Proof. The proof uses a variant of the well-known construction for plain MSO. W.l.o.g. we assume that the ωMSO⋈BAPA sentence φ is in GNF (cf. Definition 7), i.e. of the shape

$$\exists X_1. \cdots \exists X_n. \bigvee_{i=1}^{k} \left(\varphi_i \wedge \bigwedge_{j=1}^{l_i} \chi_{i,j} \right),$$

where the φ_i are CMSO formulae, whereas the $\chi_{i,j}$ are (unnegated) Parikh constraints. We then define $\varphi^{\mathcal{I}}$ to be the ωMSO⋈BAPA formula

$$\bigwedge_{\mathtt{a} \in \mathbb{S}_\mathbf{C}} (\forall z. \varphi_\mathtt{a}(z) \Leftrightarrow z = \mathtt{a}) \wedge \exists Z_{\mathrm{Dom}}. \Big(\big(\forall z. Z_{\mathrm{Dom}}(z) \Leftrightarrow \varphi_{\mathrm{Dom}}(z) \big) \wedge \varphi' \Big),$$

with $$\varphi' := \exists X_1 \subseteq Z_{\mathrm{Dom}}. \cdots \exists X_n \subseteq Z_{\mathrm{Dom}}. \bigvee_{i=1}^{k} \big(\varphi'_i \wedge \bigwedge_{j=1}^{l_i} \chi_{i,j} \big)$$

where φ'_i is obtained from φ_i by (inside out) replacing
- all $\mathtt{Q}(\iota_1, \ldots, \iota_m)$ by $\varphi_\mathtt{Q}(\iota_1, \ldots, \iota_m)$
- all $\forall x. \psi$ by $\forall x \in Z_{\mathrm{Dom}}. \psi$ as well as all $\exists x. \psi$ by $\exists x \in Z_{\mathrm{Dom}}. \psi$ and
- all $\forall X. \psi$ by $\forall X \subseteq Z_{\mathrm{Dom}}. \psi$ as well as all $\exists X. \psi$ by $\exists X \subseteq Z_{\mathrm{Dom}}. \psi$

While $\varphi^{\mathcal{I}}$ is not in GNF, it is easy to check that it is indeed in ωMSO⋈BAPA (as all set variables occurring in all $\chi_{i,j}$ are assertive). Moreover, an easy induction shows $\mathfrak{A} \models \varphi^{\mathcal{I}} \iff \mathfrak{B} \models \varphi$ (assuming such an appropriate $\mathfrak{B}$ exists). ◀

We note that this proof gives no garantees for $\varphi^{\mathcal{I}}$ in case $\mathcal{I}(\mathfrak{A})$ is undefined. We also want to point out that in the literature, there are subtle differences as to how MSO-interpretations are defined. Sometimes, an MSO-interpretation is also stipulated to feature an extra MSO formula $\varphi_=(x, y)$ with two free variables, which imposes an equality predicate (and thus a factorization on the elements of $\mathfrak{A}$, whose equivalence classes become the elements of $\mathfrak{B}$). Under such circumstances, however, finiteness and modulo atoms cannot be faithfully translated any more. This justifies our more conservative choice of MSO-interpretation.

▶ **Theorem 24.** *Let $\mathbb{S}$ be a class of structures over which satisfiability of ωMSO⋈BAPA is decidable, let $\mathcal{I}$ be an MSO-interpretation. Then satisfiability of ωMSO⋈BAPA over $\mathcal{I}(\mathbb{S})$ is decidable as well. In particular, ωMSO⋈BAPA is decidable over any tree-interpretable class.*

Proof. Given an MSO-interpretation $\mathcal{I}$, we make use of the auxiliary formula

$$\varphi_{\mathcal{I}}^{\mathrm{def}} := \exists z. \varphi_{\mathrm{Dom}}(z) \wedge \bigwedge_{\mathtt{a} \in \mathbb{S}_\mathbf{C}} \big(\exists z. \varphi_\mathtt{a}(z) \wedge \forall z' \forall z''. \varphi_\mathtt{a}(z') \wedge \varphi_\mathtt{a}(z'') \Rightarrow z' = z'' \big).$$

It is easy to see that $\varphi_{\mathcal{I}}^{\mathrm{def}}$ is satisfied by some $\mathbb{S}'$-structure $\mathfrak{A}$ exactly if $\mathcal{I}(\mathfrak{A})$ is defined.[17] Also, $\varphi_{\mathcal{I}}^{\mathrm{def}}$ is in plain MSO and hence in ωMSO⋈BAPA.

Let now φ be an ωMSO⋈BAPA formula whose satisfiability over $\mathcal{I}(\mathbb{S})$ we want to check. We then obtain, with the help of Lemma 23, that $\mathcal{I}(\mathbb{S})$ contains a model $\mathfrak{B}$ of φ if and

---

[17] We recall that we assume a model theory that requires nonempty domains, but note that this choice does not substantially affect our results.

only if there is an $\mathfrak{A} \in \mathcal{S}$ with $\mathfrak{B} = \mathcal{I}(\mathfrak{A})$ and $\mathfrak{A} \models \varphi^{\mathcal{I}}$. This, in turn, is equivalent to the existence of some $\mathfrak{A} \in \mathcal{S}$ with $\mathfrak{A} \models \varphi_{\mathcal{I}}^{\mathrm{def}} \wedge \varphi^{\mathcal{I}}$. Therefore, satisfiability of φ over $\mathcal{I}(\mathcal{S})$ coincides with satisfiability of $\varphi_{\mathcal{I}}^{\mathrm{def}} \wedge \varphi^{\mathcal{I}}$ over $\mathcal{S}$. As $\varphi_{\mathcal{I}}^{\mathrm{def}} \wedge \varphi^{\mathcal{I}}$ is in ωMSO$\bowtie$BAPA, this is a decidable problem by assumption, which concludes the proof of our claim.

The particular case for tree-interpretable classes of structures is now a straightforward consequence of Corollary 21. ◀

E   Coupling ωMSO$\bowtie$BAPA with $\mathrm{FO}^2_{\mathrm{Pres}}$

▶ **Definition 49** (Ultimately periodic sets). *A (one-dimensional) linear set is a set of the form* $\{a + ip \mid i \in \mathbb{N}\}$ *for fixed natural numbers a and p. We also consider $\emptyset$ and $\{\infty\}$ to be linear. An* ultimately periodic *set is a finite union of linear sets.*

▶ **Definition 50** ($\mathrm{FO}^2_{\mathrm{Pres}}$). *The logic $\mathrm{FO}^2_{\mathrm{Pres}}$ consists of formulae written using only 2 variables (x and y), atoms (over a signature $\mathbb{S}$ containing only unary and binary predicates), equality, conjunction, disjunction, negation and ultimately periodic existential quantification. For S an ultimately periodic set, the ultimately periodic quantifier $\exists^S x$ has the following semantics: for φ a formula in $\mathrm{FO}^2_{\mathrm{Pres}}$, the sentence $\exists^S x.\varphi$ is true for a model $\mathfrak{A}$ and variable assignment ν if and only if the cardinal of the set $\{a \in A \mid \mathfrak{A}, \nu_{x \mapsto a} \models \varphi\}$ is in S.*

This very expressive logic has recently been shown to be decidable [7]. In fact, the satisfiability problem for $\mathrm{FO}^2_{\mathrm{Pres}}$ is reduced to the satisfiability of a formula in existential Presburger Arithmetic. Given that ωMSO$\bowtie$BAPA can express existential Presburger Arithmetics, and that the translation from $\mathrm{FO}^2_{\mathrm{Pres}}$ to Presburger Arithmetic is transparent enough to partially account for the $\mathrm{FO}^2_{\mathrm{Pres}}$ models, we can pair them both to obtain a handier result. For the following, recall that, for a $\mathbb{S}$-structure $\mathfrak{C}$ and a signature $\mathbb{S}' \subseteq \mathbb{S}$, $\mathfrak{C}|_{\mathbb{S}'}$ denotes the $\mathbb{S}'$-reduct of $\mathfrak{C}$, i.e., the structure obtained by forgetting from $\mathfrak{C}$ the predicates that are not in $\mathbb{S}'$.

▶ **Lemma 51.** *Let ψ be a sentence in $\mathrm{FO}^2_{\mathrm{Pres}}$, over a signature $\mathbb{S}$ of unary and binary predicates and let $\mathbb{S}_1 \subseteq \mathbb{S}$ be a subset of the unary predicates of $\mathbb{S}$. Then there is a sentence $\psi_{\mathbb{S}_1} \in \omega$MSO$\bowtie$BAPA over the signature $\mathbb{S}_1$ such that*
- *every structure $\mathfrak{B} \models \psi$ satisfies $\mathfrak{B} \models \psi_{\mathbb{S}_1}$ and*
- *for every structure $\mathfrak{A} \models \psi_{\mathbb{S}_1}$, there is a structure $\mathfrak{B} \models \psi$ such that $\mathfrak{A}|_{\mathbb{S}_1} = \mathfrak{B}|_{\mathbb{S}_1}$.*

See further below for the corresponding proof. From this lemma, we can deduce the wanted theorem:

▶ **Theorem 26.** *Let w be any of treewidth, cliquewidth, or partitionwidth, and let $n \in \mathbb{N}$. Let $\mathbb{S}_a$ and $\mathbb{S}_b$ be signatures whose only joint elements are unary predicates. Then the following problem is decidable: Given a ωMSO$\bowtie$BAPA sentence φ over $\mathbb{S}_a$ and a $\mathrm{FO}^2_{\mathrm{Pres}}$ sentence ψ over $\mathbb{S}_b$, does there exist a countable $\mathbb{S}_a \cup \mathbb{S}_b$-structure $\mathfrak{C}$ satisfying $w(\mathfrak{C}|_{\mathbb{S}_a}) \leq n$ and $\mathfrak{C} \models \varphi \wedge \psi$.*

Proof. Let $\mathbb{S}_1 = \mathbb{S}_a \cap \mathbb{S}_b$. By applying Lemma 51 for ψ, obtain $\psi_{\mathbb{S}_1} \in \omega$MSO$\bowtie$BAPA. Then, $\varphi \wedge \psi_{\mathbb{S}_1}$ and $\varphi \wedge \psi$ are equisatisfiable over the class of structures $\mathfrak{C}$ with $w(\mathfrak{C}|_{\mathbb{S}_a}) \leq n$:
- Let $\mathfrak{A}$ be a model of $\varphi \wedge \psi_{\mathbb{S}_1}$ with $w(\mathfrak{A}|_{\mathbb{S}_a}) \leq n$. By the second item of Lemma 51, we find a $\mathfrak{B} \models \psi$ such that $\mathfrak{A}$ and $\mathfrak{B}$ have the same domain and agree on all "shared" predicates of $\mathbb{S}_a \cap \mathbb{S}_b$ (all of them unary by assumption). Thus, there exists a unique structure $\mathfrak{C}$ over $\mathbb{S}_a \cup \mathbb{S}_b$ which is an expansion of both $\mathfrak{A}$ and $\mathfrak{B}$ (the "superposition" of the two structures). By virtue of this property, $\mathfrak{C}$ satisfies all valid $\mathbb{S}_b$-sentences of $\mathfrak{A}$ (in particular ψ) and all valid $\mathbb{S}_a$-sentences of $\mathfrak{B}$ (in particular φ), therefore $\mathfrak{C}' \models \varphi \wedge \psi$. Now, from $\mathfrak{C}|_{\mathbb{S}_a} = \mathfrak{A}|_{\mathbb{S}_a}$ also follows $w(\mathfrak{C}|_{\mathbb{S}_a}) \leq n$
- Let $\mathfrak{C}$ be a model of $\varphi \wedge \psi$ with $w(\mathfrak{C}|_{\mathbb{S}_a}) \leq n$. Then, by the first item of Lemma 51, $\mathfrak{C}$ is also a model of $\varphi \wedge \psi_{\mathbb{S}_1}$.

Concluding, we have shown that our decision problem can be reduced to the question if an ωMSO$\bowtie$BAPA sentence (namely $\varphi \wedge \psi_{\mathbb{S}_1}$) has a countable model $\mathfrak{C}$ with $w(\mathfrak{C}) \leq n$, which is decidable by Corollary 25.   ◀

What remains to be proven is Lemma 51. To this end, we need to introduce the notion of atomic 1-types.

▶ **Definition 52** (Atomic 1-type). *Let $\mathbb{S}$ be a signature. An* atomic 1-type *(short: 1-type) π over $\mathbb{S}$ is a maximal coherent set of atoms and negation of atoms with only one variable x. We often see π as the conjunction of its elements.*

Note that, since $\mathbb{S}$ is finite, the set of 1-types over $\mathbb{S}$ is finite. Moreover, every element b in a structure $\mathfrak{B}$ realizes exactly one certain 1-type, that is, the 1-type π satisfying $\mathfrak{B}, \{x \mapsto b\} \models \pi$. Thus, the 1-types form a finite partition of the universe of a structure.

Now, let us prove Lemma 51 by adapting the proof from [7]. Let ψ be a sentence in $\mathrm{FO}^2_{\mathrm{Pres}}$ over a signature $\mathbb{S}$. The first step is to translate ψ into a sentence of $\mathrm{FO}^2_{\mathrm{Pres}}$ in normal form, obtaining a conservative extension ψ^* using an extended signature $\mathbb{S}' \supseteq \mathbb{S}$. That is, every model of ψ^* is a model of ψ and every model of ψ can be expanded to a model of ψ^*. The details of this effective translation can be found in the appendix of the full version of [7].

The second step is to introduce so-called *behaviors* $g_1, \ldots, g_m$ depending on ψ^*. Every element $a \in A$ in a structure $\mathfrak{A}$ will be assigned exactly one behavior out of $g_1, \ldots, g_m$. The assigned behavior quantitatively represents how a is related to the other elements in the structure. For our purposes, it is sufficient to account for the behaviors by defining, for every $\mathbb{S}'$-structure $\mathfrak{A}$, a partition over its domain A, which we will represent by means of set variables $P_{g_1}, \ldots, P_{g_m}$.

We extract the following lemma from [7, Lemma 6].

▶ **Lemma 53.** *Let $\psi^* \in \mathrm{FO}^2_{\mathrm{Pres}}$ be a sentence in normal form over a signature $\mathbb{S}$, let $\pi_1, \ldots, \pi_n$ be an enumeration of the atomic 1-types over $\mathbb{S}$, and let $g_1, \ldots, g_m$ be the list of behaviors corresponding to ψ^*.*

Then one can compute a Presburger arithmetic formula consistent$(\varkappa_1, \ldots, \varkappa_{n \cdot m})$ *over $n \cdot m$ variables that is satisfied by $(|A_{\pi_1, g_1}|, |A_{\pi_1, g_2}|, \ldots, |A_{\pi_n, g_m}|)$ if and only if there is a structure $\mathfrak{A} \models \psi^*$ where A_{π_i, g_j} is the set of the elements a of A of 1-type π_i and behavior g_j.*

To conclude, we encode this satisfiability problem with fresh set variables $P_{\pi_1}, \ldots, P_{\pi_n}$ and $P_{g_1}, \ldots, P_{g_m}$, and write the following $\omega\mathrm{MSO}{\bowtie}\mathrm{BAPA}$-sentence:

$$\psi_{\mathbb{S}_1} = \exists P_{\pi_1}. \ldots .\exists P_{\pi_n}.\exists P_{g_1}. \ldots .\exists P_{g_m}.\mathrm{consistent}(|P_{\pi_1} \cap P_{g_1}|, \ldots, |P_{\pi_n} \cap P_{g_m}|)$$

$$\wedge \forall x. \bigvee_{i=1}^{m} P_{g_i}(x) \wedge \bigwedge_{1 \leq i < j \leq m} \neg(P_{g_i}(x) \wedge P_{g_j}(x))$$

$$\wedge \forall x. \bigvee_{i=1}^{n} P_{\pi_i}(x) \wedge \bigwedge_{1 \leq i < j \leq n} \neg(P_{\pi_i}(x) \wedge P_{\pi_j}(x))$$

$$\wedge \forall x. \bigwedge_{i=1}^{n} P_{\pi_i}(x) \rightarrow \bigwedge \{(\neg)\mathrm{P}(x) \in \pi_i \mid \mathrm{P} \in \mathbb{S}_1\}$$

It is now clear that the sentence $\psi_{\mathbb{S}_1}$ is as required by Lemma 51: in any structure $\mathfrak{A} \models \psi_{\mathbb{S}_1}$, there are assignments of the 1-types and behavior functions that are partitions, such that the 1-types are coherent with the unary predicates of $\mathbb{S}_1$, and these assignments satisfy consistent, which means that they can be completed with binary and further unary relations in a way that ψ^* is satisfied. On the other hand, any structure satisfying ψ^* also satisfies ψ.

▶ **Lemma 54.** *Let $\mathfrak{A}$ be a tame structure over $\mathbb{S} = \mathbb{S}_\mathbf{C} \cup \mathbb{S}_{\mathbf{P},1} \cup \mathbb{S}_{\mathbf{P},2}$. Then the treewidth of $\mathfrak{A}$ is at most $|\mathbb{S}_\mathbf{C}| + 1$.*

Proof. Recall that A is a prefix-closed subset of of $\{rw \mid r \in Roots, w \in \mathbb{N}^*\}$ where $Roots = \{\mathsf{a}^{\mathfrak{A}} \mid \mathsf{a} \in \mathbb{S}_\mathbf{C}\}$. Let $V = \{Roots\} \cup \{Roots \cup \{a, a'\} \mid a \text{ is child of } a'\}$ and let

$$E = \{(Roots, A') \mid A' \in V, \ |A'| = |Roots| + 1\} \cup \{(A', A'') \in V \times V \mid A' \cap A'' \setminus Roots \neq \emptyset\}.$$

Then it is easy to see that $T = (V, E)$ is a tree decomposition of $\mathfrak{A}$ and the maximal size of any bag $A' \in V$ is $\leq |\mathbb{S}_\mathbf{C}| + 2$, witnessing that $\mathfrak{A}$ that has a treewidth $\leq |\mathbb{S}_\mathbf{C}| + 1$. ◀

▶ **Definition 55.** *Given a fixed signature $\mathbb{S} = \mathbb{S}_\mathbf{C} \cup \mathbb{S}_{\mathbf{P},1} \cup \mathbb{S}_{\mathbf{P},2}$, we define*

$$\varphi_{\mathrm{anon}}(z) := \bigwedge_{\mathsf{c} \in \mathbb{S}_\mathbf{C}} z \neq \mathsf{c}$$

$$\varphi_{\bullet\bullet}(z, z') := z \neq z' \wedge \varphi_{\mathrm{anon}}(z) \wedge \varphi_{\mathrm{anon}}(z') \wedge \bigvee_{\mathsf{R} \in \mathbb{S}_{\mathbf{P},2}} \mathsf{R}(z, z') \vee \mathsf{R}(z', z)$$

$$\varphi_{\mathrm{closedres}}(Y, X) := \forall y, y' \in X.\big(y \in Y \wedge \varphi_{\bullet\bullet}(y, y') \Rightarrow y' \in Y\big)$$

$$\varphi_{\mathrm{reachvia}}(z, z', X) := z \in X \wedge z' \in X \wedge \forall Y \subseteq X.\Big(z \in Y \wedge \varphi_{\mathrm{closedres}}(Y, X)\Big) \Rightarrow z' \in Y$$

$$\varphi_{\mathrm{connected}}(X) := \forall y, y' \in X.\varphi_{\mathrm{reachvia}}(y, y', X)$$

$$\varphi_{\mathrm{pthbtw}}(z, z', X) := \varphi_{\mathrm{reachvia}}(z, z', X) \wedge \varphi_{\mathrm{connected}}(X) \wedge$$
$$\exists^{=1} y \in X.\varphi_{\bullet\bullet}(z, y) \wedge \exists^{=1} y \in X.\varphi_{\bullet\bullet}(z', y) \wedge \forall x \in X.\exists^{\leq 2} y \in X.\varphi_{\bullet\bullet}(x, y)$$

Then we define $\psi_{\mathrm{tame}}^{\mathbb{S}}$ to be the MSO sentence

$$\neg \exists z, z', X, X'. \varphi_{\mathrm{pthbtw}}(z, z', X) \wedge \varphi_{\mathrm{pthbtw}}(z, z', X') \wedge X \neq X'$$

For better readability, the above definition uses the counting quantifiers $\exists^{=1}$ and $\exists^{\leq 2}$ as abbreviations, which can be easily expressed:

$$\exists^{=1} y \in X.\varphi_{\bullet\bullet}(x, y) \quad \equiv \quad \exists y.\varphi_{\bullet\bullet}(x, y) \ \wedge \ \forall y_1, y_2 \in X.\big(\varphi_{\bullet\bullet}(x, y_1) \wedge \varphi_{\bullet\bullet}(x, y_2)\big) \Rightarrow y_1 = y_2$$

$$\exists^{\leq 2} y \in X.\varphi_{\bullet\bullet}(x, y) \quad \equiv \quad \forall y_1, y_2, y_3 \in X.\big(\varphi_{\bullet\bullet}(x, y_1) \wedge \varphi_{\bullet\bullet}(x, y_2) \wedge \varphi_{\bullet\bullet}(x, y_3)\big)$$
$$\Rightarrow y_1 = y_2 \vee y_1 = y_3 \vee y_2 = y_3$$

Now we observe that φ_{anon} holds for all those $\{z \mapsto a\}$ where a is an "anonymous" element, i.e., one not "named" by any of the constants from $\mathbb{S}_\mathbf{C}$. Further, $\varphi_{\bullet\bullet}$ is satisfied by those $\{z \mapsto a, z' \mapsto a'\}$ where both a and a' are anonymous, they are distinct, and co-occur in some relation pair of some $\mathsf{R}^{\mathfrak{A}}$ with $\mathsf{R} \in \mathbb{S}_{\mathbf{P},2}$; we will abbreviate this situation by $a \bullet\!\!-\!\!\bullet a'$. Then, $\varphi_{\mathrm{closedres}}$ characterizes those $\{Y \mapsto A', X \mapsto A''\}$, where A' is closed under the restricted co-occurence relation $\bullet\!\!-\!\!\bullet \cap (A'' \times A'')$. Moreover, $\varphi_{\mathrm{reachvia}}$ holds for $\{z \mapsto a, z' \mapsto a', X \mapsto A'\}$ whenever $a = a'$ or there is an entirely anonymous $\bullet\!\!-\!\!\bullet$-path from a to a' which only traverses elements of A'. Then, $\varphi_{\mathrm{connected}}$ will identify those $\{X \mapsto A'\}$ for which the set A' is $\bullet\!\!-\!\!\bullet$-connected. Consequently, $\varphi_{\mathrm{pthbtw}}$ will be satisfied by $\{z \mapsto a, z' \mapsto a', X \mapsto A'\}$ iff A' consists of the elements of a simple (i.e., repetition-free) $\bullet\!\!-\!\!\bullet$-path between a and a'. Finally, $\psi_{\mathrm{tame}}^{\mathbb{S}}$ disallows the case that two anonymous domain elements are connected by two distinct entirely anonymous simple $\bullet\!\!-\!\!\bullet$-paths. Yet, this property precisely characterizes structures isomorphic to tame structures: a structure is tame iff removing all relation instances with participating named elements and considering the Gaifman graph of the ensuing structure, we find it to be a disjoint union of undirected trees, which is characterized by the property that between any two elements, there is at most one simple path.

▶ **Lemma 56.** *Given a signature $\mathbb{S} = \mathbb{S}_{\mathbf{C}} \cup \mathbb{S}_{\mathbf{P},1} \cup \mathbb{S}_{\mathbf{P},2}$ a $\mathbb{S}$-structure $\mathfrak{A}$ satisfies $\psi_{\mathrm{tame}}^{\mathbb{S}}$ iff it is isomorphic to a tame structure over $\mathbb{S}$.*

Toward the proof of Theorem 29, we first provide a translation from FEµ formulae to ωMSO⋈BAPA.

▶ **Definition 57.** *Given an individual variable x, we define the translation function trans_x mapping FEµ formulae to MSO formulae:*

$$
\begin{array}{ll}
\mathbf{true} \mapsto \mathbf{true} & \varphi \wedge \varphi' \mapsto \mathrm{trans}_x(\varphi) \wedge \mathrm{trans}_x(\varphi') \\
\mathbf{false} \mapsto \mathbf{false} & \varphi \vee \varphi' \mapsto \mathrm{trans}_x(\varphi) \vee \mathrm{trans}_x(\varphi') \\
X \mapsto X(x) & \langle n, \alpha \rangle \varphi \mapsto \exists Y. \mathtt{n} \leq \#Y \wedge \forall x' \in Y. \alpha(x, x') \wedge \mathrm{trans}_{x'}(\varphi) \\
\mathtt{c} \mapsto x = \mathtt{c} & [n, \alpha] \varphi \mapsto \forall Y. \mathtt{n} + 1 \leq \#Y \Rightarrow \exists x' \in Y. \neg \alpha(x, x') \vee \mathrm{trans}_{x'}(\varphi) \\
\neg \mathtt{c} \mapsto x \neq \mathtt{c} & \mu X. \varphi \mapsto \forall X. (\forall x'. \mathrm{trans}_{x'}(\varphi) \Rightarrow X(x')) \Rightarrow X(x) \\
\mathtt{P} \mapsto \mathtt{P}(x) & \nu X. \varphi \mapsto \exists X. (\forall x'. X(x') \Rightarrow \mathrm{trans}_{x'}(\varphi)) \wedge X(x) \\
\neg \mathtt{P} \mapsto \neg \mathtt{P}(x) &
\end{array}
$$

where $\mathtt{R}^-(x, x')$ is understood to paraphrase $\mathtt{R}(x', x)$. Note that, in the cases for μ and ν, the set variable X is supposed to occur freely in φ (and hence also in $\mathrm{trans}_{x'}(\varphi)$). Contrarily, Y is always meant to be a fresh set variable (in particular one not occurring in $\mathrm{trans}_{x'}(\varphi)$).

The provided translation closely follows and slightly extends established canonical translations from the μ-calculus to MSO logic, and along the same lines, it is straightforward to obtain the following correspondence.

▶ **Lemma 58.** *Let φ be a closed FEµ formula over a signature $\mathbb{S}$ and let $\mathfrak{A}$ be a structure over $\mathbb{S}$. Then $\mathfrak{A}, \{x \mapsto a\} \models \mathrm{trans}_x(\varphi)$ iff $a \in [\![\varphi]\!]_{\emptyset}^{\mathfrak{A}}$.*

In words, given a structure $\mathfrak{A}$, the semantics of FEµ assigns to each closed FEµ formula φ a subset $A' = [\![\varphi]\!]_{\emptyset}^{\mathfrak{A}}$ of A (intuitively, A' represents the elements/states/worlds, "wherein φ holds"). The translation function trans_x maps each such φ to an ωMSO⋈BAPA formula $\varphi'(x)$ (with one free individual variable x) which holds precisely for those $a \in A$ that are contained in the set A' described by φ. Then, the expression $\#\varphi$ inside a global FEµ Presburger constraint PC (which is the only context such a φ can occur in) simply stands for $|A'|$. With this insights and in view of Lemma 58, it is then not hard to come up with an ωMSO⋈BAPA counterpart of PC where all $\#\varphi_i$ inside PC are replaced back by $\#X_i$ for fresh set variables X_i, which are then axiomatized to be instantiated with the corresponding $[\![\varphi_i]\!]_{\emptyset}^{\mathfrak{A}}$.

▶ **Definition 59.** *Given a global FEµ Presburger constraint $PC(\#\varphi_1, \ldots, \#\varphi_k)$ let*

$$
\mathrm{trans}(PC) = \exists X_1, \ldots, X_k. PC(\#X_1, \ldots, \#X_k) \wedge \bigwedge_{1 \leq i \leq k} \forall z. X_i(z) \Leftrightarrow \mathrm{trans}_z(\varphi_i).
$$

For a finite set Π of global FEµ Presburger constraints, let $\mathrm{trans}(\Pi) = \{\mathrm{trans}(PC) \mid PC \in \Pi\}$.

▶ **Lemma 60.** *Let Π be a finite set of global FEµ Presburger constraints. Then $\mathfrak{A} \models \Pi$ iff $\mathfrak{A} \models \mathrm{trans}(\Pi)$.*

We now have assembled all ingredients to establish the wanted result.

▶ **Theorem 29.** *The tame satisfiability problem of the fully enriched μ-calculus with global Presburger constraints is decidable.*

Proof. Let $\mathbb{S}$ be a finite signature, φ a corresponding sentence of the fully enriched μ-calculus, and Π a finite set of global FEμ Presburger constraints. Then, by virtue of the above lemmas, tame satisfiability of (φ, Π) corresponds to satisfiability of the $\omega\text{MSO}\bowtie\text{BAPA}$ sentence $\psi_{\text{tame}}^{\mathbb{S}} \wedge \exists x.\text{trans}_x(\varphi) \wedge \bigwedge \text{trans}(\Pi)$ over all countable structures of treewidth $\leq |\mathbb{S}_{\mathbf{C}}| + 1$, which is decidable by Corollary 25.   ◀