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, Paris, France

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

Abstract

We propose ωMSO⋈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 ωMSO⋈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 ωMSO⋈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.

2012 ACM Subject Classification Theory of computation → Logic; Theory of computation → Tree languages; Theory of computation → Automata over infinite objects; Theory of computation → Automated reasoning

Keywords and phrases MSO, BAPA, Parikh-Muller tree automata, decidability, MSO-interpretations

Digital Object Identifier 10.4230/LIPIcs.CSL.2024.33

Related Version *An extended version of the paper including more details and full proofs is available at*: https://arxiv.org/abs/2305.01962

Funding BMBF (SCADS22B) and SMWK by funding ScaDS.AI Dresden/Leipzig.
Sebastian Rudolph: European Research Council, Consolidator Grant DeciGUT (771779).

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. In particular, the MSO theory of infinite binary trees is decidable by Rabin's famous result [50], and the same holds for structures of bounded treewidth, cliquewidth, and partitionwidth.

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* [40, 41] (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 [42]) can be recovered when confining set variables occurring in cardinality constraints to those existentially quantified in front ($\mathrm{MSO}^{\exists\mathrm{Card}}$). 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* [44, 45]. 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 [46]; 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⋈BAPA [ˈoːmzoǁˌbapa],[1] a logic for countable structures, which extends CMSO by BAPA's set operations and Presburger statements, strictly contains $\mathrm{MSO}^{\exists\mathrm{Card}}$, 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⋈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⋈BAPA over the class of labeled infinite binary trees thereby obtained is then lifted to all *tree-interpretable classes*, including vast and practically

[1] Note that the "⋈" inside the name is meant to be pronounced as lateral click, commonly used by riders and coachmen to urge on their horses, and present in several African languages as a consonant.

relevant classes of finite or countable structures that are bounded in terms of certain width measures (Section 7). Such width-bounded ωMSO⋈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 $\mathtt{a}, \mathtt{b}, \mathtt{c}, \ldots$, and, for every $n \in \mathbb{N}$, a set $\mathbf{P}_n$ of *n-ary predicates*, denoted by $\mathtt{P}, \mathtt{R}, \mathtt{Q}, \ldots$. The set of all predicates will be denoted by $\mathbf{P} \coloneqq \bigcup_{i\in\mathbb{N}} \mathbf{P}_n$, 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^ω_Σ. 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 $\xi \in T^\omega_\Sigma$ 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 $\pi = \pi_1\pi_2\ldots$ 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 $\xi \in T^\omega_\Sigma$ and a path π, we denote by $\xi(\pi)$ the infinite word $\xi(\pi_1)\xi(\pi_2)\ldots$ 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 $\Sigma = \{a_1,\ldots,a_n\}$ as structures over the signature $\mathbb{S} = \mathbb{S}_\mathbf{P} = \{\succ_0, \succ_1, \mathtt{P}_{a_1}, \ldots, \mathtt{P}_{a_n}\}$: Thereby, a tree $\xi \in T^\omega_\Sigma$ 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 $x, y, z, \ldots$), $\mathbf{V}_{\mathrm{set}}$ of* set variables *(denoted $X, Y, Z, \ldots$), and $\mathbf{V}_{\mathrm{num}}$ of* number variables *(denoted $\Bbbk, \ell, m, n\ldots$), we define the following sets of expressions by mutual induction:*

- *the set* $\mathbf{I}$ *of* individual terms*:* $\iota ::= \mathtt{a} \mid x$
- *the set* $\mathbf{S}$ *of* set terms *(*$\mathtt{P}$ *being a unary predicate):* $S ::= \{\mathtt{a}\} \mid \mathtt{P} \mid X \mid S^c \mid S_1 \cap S_2 \mid S_1 \cup S_2$
- *the set* $\mathbf{N}$ *of* number terms*:*[2] $t ::= \mathtt{n} \mid \mathtt{\infty} \mid \hbar \mid \#S \mid \mathtt{m}\, t \mid t_1 + t_2$
  (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*:*

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

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.*

▶ **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* $\hbar \in \mathbf{V}_{\mathrm{num}}$ *to a number* $\nu(\hbar) \in \mathbb{N} \cup \{\infty\}$*.*

We write $\nu_{x\mapsto a}$*,* $\nu_{X\mapsto A'}$*, and* $\nu_{\hbar\mapsto n}$ *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*

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

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

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

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

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

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.*

[2] 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.

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$\cdot$BAPA does not provide an explicit equality predicate, both individual and set equality can be expressed (see further below).

▶ **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$\cdot$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 ω*MSO⋈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.

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

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. †).

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

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

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

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

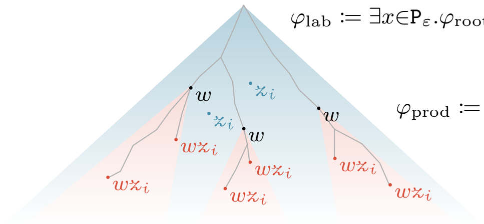
Just slightly relaxing the syntax of ωMSO⋈BAPA allows us to express Hilbert's 10th Problem.

▶ **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 $\{\varkappa_1, \ldots, \varkappa_k\}$ 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*

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



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

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

$$\varphi_{\text{sol}} \coloneqq \sum\nolimits_{w\in M} \mathtt{n}_w\, \#\mathtt{P}_w =_{\mathtt{fin}} \sum\nolimits_{w\in M} \mathtt{m}_w\, \#\mathtt{P}_w$$

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

Solvability of positive Diophantine equations is undecidable, which is a straightforward consequence of the undecidability of arbitrary Diophantine equations over integers [48].

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 [39, Thm. 8.13].

As detailed in Figure 1, we let $\varphi_{\mathcal{D}} \coloneqq \varphi_{\text{lab}} \wedge \varphi_{\text{prod}} \wedge \varphi_{\text{sol}}$ 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., $\#\mathtt{P}_\varkappa$) equals the number that ν assigns to $\varkappa$. Likewise, for each variable product $w\varkappa_i \in M$, we ensure that $\#\mathtt{P}_{w\varkappa_i} = \#\mathtt{P}_w \cdot \#\mathtt{P}_{\varkappa_i}$. 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 is a variable-free Presburger atom, the second is not in ωMSO⋈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 ωMSO·BAPA sentences of the shape $\varphi_{\mathcal{D}}$ is undecidable.*

5 Transformation into Normal Form

Toward establishing our decidability result, we show that ωMSO⋈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 ωMSO⋈BAPA formula, substitute complex set expressions in modulo and finiteness atoms by new set variables (e.g. $\mathrm{Fin}(\mathtt{P}\cap X)$ becomes $\exists Y.(Y{=}\mathtt{P}\cap X)\wedge \mathrm{Fin}(Y)$), remove set operations from set atoms (e.g. turning $(\mathtt{P}^c\cap X)(y)$ into $\neg\mathtt{P}(y)\wedge X(y)$), and rewrite all simple Presburger atoms into plain MSO (e.g. $\mathtt{2\,\#P \le 3}$ becomes $\forall xy.\mathtt{P}(x){\wedge}\mathtt{P}(y) \Rightarrow x{=}y$). Then, *skolemize* all assertive variables (e.g. $\exists x.\exists X.\forall y.\mathtt{R}(x,y){\Rightarrow}X(y)$ becomes $\forall y.\mathtt{R}(\mathtt{c}_x,y){\Rightarrow}\mathtt{P}_X(y)$). Next *"presburgerize"* all non-Presburger atoms containing (only) delicate variables (e.g. replacing $\#X \equiv_3 \mathtt{1}$ with $\exists\hbar.\#X =_{\mathtt{fin}} \mathtt{3}\,\hbar + \mathtt{1}$), which may require to turn delicate individual into set variables (e.g. $\forall y.\mathtt{P}(y) \Rightarrow X(y)$ becomes $\forall Y.(\#Y = 1) \wedge 1 \leq \#(\mathtt{P} \cap Y) \Rightarrow 1 \leq \#(X \cap Y)$). 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.[3]

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, $\#(\mathtt{P} \cup X) \leq \#\mathtt{P}^c$ becomes $k_{\mathtt{P}\cap X} + k_{\mathtt{P}^c\cap X} + k_{\mathtt{P}\cap X^c} \leq k_{\mathtt{P}^c\cap X} + k_{\mathtt{P}^c\cap X^c}$). 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.

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,[4] 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 $\#X \equiv_{\mathtt{n}} \mathtt{m}$ into the formula

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

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.*

[3] While this transformation is not very complicated technically, it may incur nonelementary blowup.
[4] Recall that CMSO is MSO with modulo and finiteness atoms over set variables.

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$\bowtie$BAPA over labeled infinite binary trees.

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

Given a finite set $\mathbf{V} \subseteq (\mathbf{V}_{\text{ind}} \cup \mathbf{V}_{\text{set}})$, 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 $\mathcal{L}_{\mathbf{V}}(\varphi) \coloneqq \{(\xi,\nu) \mid \xi \in T^{\omega}_{\Sigma}, \nu \in \Phi_{\mathbf{V}}, \xi,\nu \models \varphi\}$ and by $\mathcal{L}(\varphi)$ we mean $\mathcal{L}_{\text{free}(\varphi)}(\varphi)$. To represent $\mathbf{V}$-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 $\xi \in T^{\omega}_{\Sigma_{\mathbf{V}}}$ is *valid* (i.e., it encodes a variable assignment) if for each individual variable x in $\mathbf{V}$ there is exactly one position in ξ where x occurs. As there is a bijection between $T^{\omega}_{\Sigma} \times \Phi_{\mathbf{V}}$ and the set of all valid trees in $T^{\omega}_{\Sigma_{\mathbf{V}}}$, 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* $\Psi\colon T_{\Sigma\times D} \to \mathbb{N}^s$ is defined for each finite, non-empty tree $\xi \in T_{\Sigma\times D}$ by $\Psi(\xi) = \sum_{i\in\text{pos}(\xi)}(\xi(i))_D$.

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 [49]. 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*

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

$\mathcal{F} \subseteq 2^{Q_\omega}$ *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 $\zeta \in T^\omega_\Xi$ is a tree $\kappa_\zeta \in T^\omega_Q$ 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*

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

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 $\tau\colon T^\omega_\Sigma \to \mathcal{P}(T^\omega_\Gamma)$ 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 [41, 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 $Z^0_1, \dots, Z^K_1, \dots, Z^0_s, \dots, Z^K_s$ (with s being the number of counters and K the greatest counter increment occurring in $\mathcal{A}$'s transitions) – the presence of a variable $Z^{d_i}_i$ 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 $p_g(m_1,\ldots,m_l) = \vec{v}_0 + m_1\vec{v}_1 + \ldots + m_l\vec{v}_l$ with $\vec{v}_j = (v_{j,1},\ldots,v_{j,s})$. Then, using number variables $\boldsymbol{m}_1,\ldots,\boldsymbol{m}_l$, we encode p_g by

$$\varphi_{p_g} := \exists \boldsymbol{m}_1 \ldots \exists \boldsymbol{m}_l. \bigwedge_{i=1}^{s} \Big(\sum_{d=0}^{K} \mathtt{d}\#Z_i^d \mathrel{=_{\mathtt{fin}}} \mathtt{v}_{0,i} + \mathtt{v}_{1,i}\boldsymbol{m}_1 + \ldots + \mathtt{v}_{l,i}\boldsymbol{m}_l\Big),$$

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 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 $\mathtt{c} + \sum_{i\in[r]} \mathtt{c_i}\,\#X_i \mathrel{\leq_{\mathtt{fin}}} \mathtt{d} + \sum_{j\in[k]} \mathtt{d_j}\,\#Y_j$ 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

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

and, by the condition $<\infty$, both sums 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 $D = \{(i,j) \mid 0 \leq i \leq \sum_{l\in[r]} c_l, 0 \leq j \leq \sum_{l\in[k]} d_l\}$. 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

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

and $C = \{(z_1, z_2) \mid \mathtt{c} + z_1 \mathrel{\leq_{\mathtt{fin}}} \mathtt{d} + z_2\}$.[5] Then, one can easily show that $\mathcal{L}(\chi) = \mathcal{L}(\mathcal{A})$. ◀

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

Proof. Let φ be an ωMSO⋈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. The proof of the statement is an induction on the (now restricted) structure of φ using the well-known recognizability of MSO sentences [50], 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.

---

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

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 [40, 39], slightly adjusted to our setting.

▶ **Definition 18** (Parikh tree automaton [41]). *Let Σ be an alphabet, let $s \geq 1$, 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.*[6] *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*

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

It was shown in [39] 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 [39, 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 [50, 38], $\mathcal{A}_P$ 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]).

[6] We note that the PTAs defined in [41] 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.

▶ **Definition 22** (MSO-Interpretation). *Given two signatures $\mathbb{S}$ and $\mathbb{S}'$, an* MSO-interpretation *is a sequence $\mathcal{I} = (\varphi_{\mathrm{Dom}}(x), (\varphi_{\mathtt{c}}(x))_{\mathtt{c}\in\mathbb{S}_{\mathbf{C}}}, (\varphi_{\mathtt{Q}}(x_1,\ldots,x_{ar(\mathtt{Q})}))_{\mathtt{Q}\in\mathbb{S}_{\mathbf{P}}})$ 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 $\{a \in B \mid \mathfrak{A}, \{x \mapsto a\} \models \varphi_{\mathtt{c}}(x)\} = \{\mathtt{c}^{\mathfrak{B}}\}$ for every $\mathtt{c} \in \mathbb{S}_{\mathbf{C}}$, and, for every $\mathtt{Q} \in \mathbb{S}_{\mathbf{P}}$, we have $\mathtt{Q}^{\mathfrak{B}} = \{(a_1,\ldots,a_{ar(\mathtt{Q})}) \in B^{ar(\mathtt{Q})} \mid \mathfrak{A}, \{x_i \mapsto a_i\}_{1\leq i\leq ar(\mathtt{Q})} \models \varphi_{\mathtt{Q}}(x_1,\ldots,x_{ar(\mathtt{Q})})\}$. 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 $\mathcal{I}(T^{\omega}_{\Sigma})$ 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 $\mathfrak{A} \models \varphi^{\mathcal{I}} \iff \mathfrak{B} \models \varphi$ for every $\mathbb{S}'$-structure $\mathfrak{A}$ and $\mathbb{S}$-structure $\mathfrak{B}$ with $\mathcal{I}(\mathfrak{A}) \cong \mathfrak{B}$.*

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 [51], 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.[7] 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 $\mathbb{L} = \mathrm{FO}^2_{\mathrm{Pres}}$ [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 $|\{a \in A \mid \mathfrak{A}, \nu_{x\mapsto a} \models \varphi\}| \in S$. $\mathrm{FO}^2_{\mathrm{Pres}}$ 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].

[7] We refer to Kotek et al. [42] for a result that is similar in spirit, establishing decidability of finite satisfiability of treewidth-bounded MSO_2 coupled with C^2.

▶ **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* $\mathrm{FO}^2_{\mathrm{Pres}}$ *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 $\mathfrak{C} \models \varphi \wedge \psi$.*

In a nutshell, this result is obtained by exploiting the fact that, for every $\mathrm{FO}^2_{\mathrm{Pres}}$ 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 variations and extensions of propositional modal logics [8, 9] and description logics [4, 53]. This class contains most ontology languages as well as PDL [29], CTL* [24], the propositional modal μ-calculus [43] and their extensions. Modulo some representational variations, all these logics' model-theoretic semantics rest on structures over unary and binary predicates (often interpreted as a transition system's state space). While the simpler variants of this family can be seen as fragments of first-order logic, the more expressive ones cannot, as they feature fixed-point capabilities (through regular path expressions or explicit fixed-point operators). Typically, decidability of the satisfiability problem in these logics follows from some sort of tree-model property. Many of these logics exhibit some limited local counting capabilities [54], but recently, there has been an increased interest in accommodating more advanced arithmetic constraints [23, 47, 2, 5], including *global constraints* [3, 52] expressing statistical information such as "more than 50% of the state space's final states are successful".

We will demonstrate the usefulness of ωMSO⋈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.[8]

▶ **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.

[8] 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.

Given a structure $\mathfrak{A}$ *and a set variable assignment* $\nu : \mathbf{V}_{\text{set}} \to 2^A$*, the semantics* $[\![\varphi]\!]^{\mathfrak{A}}_{\nu} \subseteq A$ *of formulae* φ *is defined by the following function (stipulating* $(\mathtt{R}^-)^{\mathfrak{A}} = \{(a, a') \mid (a', a) \in \mathtt{R}^{\mathfrak{A}}\}$*):*

$$\begin{array}{llll}
\mathbf{true} \mapsto A \quad X \mapsto \nu(X) & \mathtt{c} \mapsto \{\mathtt{c}^{\mathfrak{A}}\} & \mathtt{P} \mapsto \mathtt{P}^{\mathfrak{A}} & \varphi \wedge \varphi' \mapsto [\![\varphi]\!]^{\mathfrak{A}}_{\nu} \cap [\![\varphi']\!]^{\mathfrak{A}}_{\nu} \\
\mathbf{false} \mapsto \emptyset & \neg\mathtt{c} \mapsto A \setminus \{\mathtt{c}^{\mathfrak{A}}\} & \neg\mathtt{P} \mapsto A \setminus \mathtt{P}^{\mathfrak{A}} & \varphi \vee \varphi' \mapsto [\![\varphi]\!]^{\mathfrak{A}}_{\nu} \cup [\![\varphi']\!]^{\mathfrak{A}}_{\nu} \\
\langle n, \alpha \rangle\varphi \mapsto \{a \mid |\{\alpha^{\mathfrak{A}} \cap (\{a\} \times [\![\varphi]\!]^{\mathfrak{A}}_{\nu})\}| \geq n\} & & \multicolumn{2}{l}{\mu X.\varphi \mapsto \bigcap\{A' \subseteq A \mid [\![\varphi]\!]^{\mathfrak{A}}_{\nu_{X \mapsto A'}} \subseteq A'\}} \\
[n, \alpha]\varphi \mapsto \{a \mid |\{\alpha^{\mathfrak{A}} \cap (\{a\} \times (A \setminus [\![\varphi]\!]^{\mathfrak{A}}_{\nu}))\}| \leq n\} & & \multicolumn{2}{l}{\nu X.\varphi \mapsto \bigcup\{A' \subseteq A \mid A' \subseteq [\![\varphi]\!]^{\mathfrak{A}}_{\nu_{X \mapsto A'}}\}}
\end{array}$$

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* $[\![\psi]\!]^{\mathfrak{A}}_{\emptyset}$ *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* $[\![\varphi]\!]^{\mathfrak{A}}_{\emptyset} \neq \emptyset$*, 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* $\{rw \mid r \in \mathit{Roots}, w \in \mathbb{N}^*\}$*,*
- *the roots coincide with the named elements, i.e.,* $\mathit{Roots} = \{\mathtt{a}^{\mathfrak{A}} \mid \mathtt{a} \in \mathbb{S}_{\mathbf{C}}\}$*, 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)* $\{a, a'\} \cap \mathit{Roots} \neq \emptyset$*, 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 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 $a \in [\![\varphi]\!]^{\mathfrak{A}}_{\emptyset}$. 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 $x \in \mathbf{V}_{\mathrm{ind}}$ 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 $\chi' \coloneqq \exists Z.(\forall x.Z(x) \Leftrightarrow \psi) \wedge \chi[\{x \mid \psi\} \mapsto Z]$. ωMSO⋈BAPA membership of such extended formulae can then be decided based on their "purified" variant,[9] 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.

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.

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

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*, pages 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 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.

39 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.

40 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).

41 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.

42 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 für Informatik, 2016. doi:10.4230/LIPIcs.CSL.2016.13.

43 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`.

44 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`.

45 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`.

46 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`.

47 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`.

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

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

50 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`.

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

52 "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`.

53 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`.

54 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`.